Advancing Semantic and Digital Communications through Machine Learning

Dissertation

zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt dem Fachbereich 1 (Physik/Elektrotechnik)

der Universität Bremen

von

Edgar Beck, M.Sc.

Eingereicht am:	23.05.2025
Tag des öffentlichen Kolloquiums:	10.10.2025
Gutachter der Dissertation:	Prof. Dr.-Ing. Armin Dekorsy
	Prof. Dr. Petar Popovski
Weitere Prüfer:	Prof. Dr.-Ing. Karl-Ludwig Krieger
	Prof. Dr.-Ing. Björn Lüssem

Bremen, November 18, 2025

Abstract

Artificial Intelligence (AI) is becoming increasingly prevalent in daily life, driven by rapid advancements in Machine Learning (ML) since 2010. These breakthroughs, enabled by innovations such as specialized hardware, Deep Neural Networks (DNNs), and advances in training techniques, have allowed AI systems to match or even exceed human performance in tasks like autonomous driving and medical diagnostics, with systems such as ChatGPT and AlphaGo standing out as key examples. These achievements have raised public awareness and acceptance of AI technologies.

In the realm of wireless communication, emerging applications such as virtual reality and autonomous systems are pushing traditional digital communication systems to their limits. Conventional content-agnostic approaches struggle to meet the growing demands for bandwidth, power efficiency, and low latency.

This dissertation explores mastering these challenges by integrating ML techniques into wireless communication systems. It introduces CMDNet, a novel framework for symbol detection, designed to improve communication efficiency by combining strengths of traditional model-based designs with those of advanced ML methods. Furthermore, integrating semantic content into communications is identified as crucial for further enhancing system efficiency. Semantic communication aims at transmitting the meaning conveyed by the data rather than the exact bits, which can introduce a model deficit that challenges traditional communication designs. This challenge in design of semantic communication is addressed using advanced ML techniques, as demonstrated in the SINFONY approach.

Together, these contributions demonstrate how ML advances, such as DNNs, can overcome existing limitations in terms of model and algorithmic deficits and significantly enhance the efficiency and capabilities of future communication systems.

Keywords

Algorithm deficit, artificial intelligence, CMDNet, deep neural networks, deep unfolding, information maximization principle, information theory, machine learning, massive MIMO, model deficit, semantic communication, SINFONY, soft detection, wireless communication systems

Kurzfassung

Künstliche Intelligenz (KI) wird im Alltag zunehmend präsenter, angetrieben durch rasante Fortschritte im Bereich des Maschinellen Lernens (ML) seit 2010. Diese Durchbrüche, ermöglicht durch Innovationen wie spezialisierte Hardware, tiefe Neuronale Netze (Deep Neural Networks, DNNs) sowie verbesserte Trainingsmethoden, haben es KI-Systemen erlaubt, in Aufgaben wie autonomem Fahren und medizinischer Diagnostik menschliche Leistungen zu erreichen oder gar zu übertreffen. Systeme wie ChatGPT und AlphaGo sind herausragende Beispiele hierfür. Diese Erfolge haben das öffentliche Bewusstsein für und die Akzeptanz von KI-Technologien deutlich erhöht.

Im Bereich der drahtlosen Kommunikation bringen neue Anwendungen wie virtuelle Realität und autonome Systeme die traditionellen digitalen Kommunikationssysteme an ihre Grenzen. Herkömmliche inhaltsagnostische Ansätze haben Schwierigkeiten, die wachsenden Anforderungen an Bandbreite, Energieeffizienz und geringe Latenzzeiten zu erfüllen.

In dieser Dissertation geht es um die Bewältigung dieser Herausforderungen durch die Integration von ML-Techniken in drahtlose Kommunikationssysteme. Vorgestellt wird CMDNet, ein neuartiges Framework zur Symboldetektion, das darauf abzielt, die Kommunikationseffizienz durch die Kombination der Stärken modellbasierter Methoden mit denen der fortgeschrittenen ML-Verfahren zu verbessern. Darüber hinaus wird die Integration von semantischen Inhalten in die Kommunikation als entscheidend für die weitere Verbesserung der Systemeffizienz identifiziert. Semantische Kommunikation zielt darauf ab, die Bedeutung der Daten zu übertragen, anstatt deren exakte Bits, was zu einem Modelldefizit führt. Dieses stellt herkömmliche Kommunikationsansätze vor große Herausforderungen. Beim Design der semantischen Kommunikation werden diese mithilfe fortschrittlicher ML-Techniken gelöst, wie mit dem SINFONY-Ansatz demonstriert.

Zusammen zeigen diese Beiträge, wie ML-Fortschritte, z. B. DNNs, bestehende Beschränkungen in Bezug auf Modell- und Algorithmusdefizite überwinden und die Effizienz und Fähigkeiten zukünftiger Kommunikationssysteme erheblich verbessern können.

Preface

This dissertation was completed during my time as a research engineer at the Department of Communications Engineering, Universität Bremen, Germany.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Dr.-Ing. Armin Dekorsy, for his invaluable guidance and continuous support throughout this journey. His enthusiasm for machine learning and semantic communication greatly influenced my work, and his trust in my abilities allowed me the freedom to explore my ideas. Our many discussions, though at times challenging, pushed me to think critically and helped me grow as a researcher. I am also grateful for his encouragement to publish my findings in prestigious conferences and journals, which played an essential role in my academic development.

I would also like to extend my special thanks to Prof. Dr. Petar Popovski from Aalborg University, Denmark, for his keen interest in my work including stimulating discussions and for serving as the second reviewer. Likewise, I am grateful to Prof. Dr.-Ing. Karl-Ludwig Krieger and Prof. Dr.-Ing. Björn Lüssem for their roles as examiners.

I would like to express my deep gratitude to Dr.-Ing. Carsten Bockelmann, who was an essential pillar of support alongside my professor. His guidance, both in technical discussions and in soft skills, was invaluable, even when not deeply involved in the subject matter. Our trusted collaboration extended beyond the professional realm, and his broad knowledge and mentorship on both personal and professional levels were key to my development.

My gratitude extends to my former colleagues at the department, whose camaraderie and collaborative spirit created a unique working environment that made the entire experience both enjoyable and intellectually stimulating, helping me grow on a personal level. Special thanks go to Ban-Sok Shin and Matthias Hummert, whose support, discussions, and jokes in the beginning of my journey were invaluable. I am also grateful to the latter, Tim Düe, and Maximilian Tillmann for thorough proofreading of this thesis.

My deepest thanks go to my family, especially my parents for their unwavering belief in my abilities and their constant support throughout this journey. They were always there to listen and offer comfort, especially during the most challenging moments. I am also incredibly grateful to my friends, Johannes, Lorenz and Matthias, who stood by me from start to finish. Their optimism during the COVID period was particularly appreciated. Thank you for the many shared moments of laughter and for reminding me what matters most in life. Special thanks go to my karate club, Bushido Verden, whose balance and strength helped me remain grounded and resilient throughout this journey.

Finally, I would like to express my deepest gratitude to my girlfriend Khatia who stepped into my life at the very end of this PhD journey. She sweetened my life, was a great support, and inspirational source for the final meters.

As I reflect on this long journey, a few timeless philosophical insights come to mind. The following quotes, one for each main part of the thesis, give voice to these insights that gradually became clear to me as I explored the broader context of the respective topics:

Part I

"I neither know nor think I know."

– Socrates, 5th century BC

Part II

"The whole is more than the sum of the parts."

– Aristotle, 4th century BC

In closing, I hope that the insights gained through this work may offer a small contribution to the *enduring, mysterious puzzle of the universe.*

Bremen, November 2025

Edgar Beck

Contents

Chapter 1

Introduction

1.1 Motivation and Background

Artificial Intelligence (AI) is becoming increasingly prevalent in everyday life [HB23]. This surge is largely due to rapid advancements in Machine Learning (ML), a subfield of AI, since 2010, which have led to significant breakthroughs. Special hardware and software such as Graphics Processing Units (GPUs) and automatic differentiation systems, innovations in Deep Neural Network (DNN) models and advances in training have enabled the creation of algorithms that match or even surpass human performance in certain tasks [CMS12]. These achievements include advancements in image processing [KSH12], enabling applications such as autonomous driving and medical diagnostics, as well as the development of autonomous AI systems like AlphaGo, which have defeated professional players in complex games like chess and Go [SHM+16]. Notably, Go was once considered a game that required uniquely human intuition and strategic thinking to win.

For example, doctors are now assisted by expert systems in evaluating medical image data for disease diagnosis, with some systems surpassing human expertise [CGGS13]. Moreover, the fundamental open problem of protein folding which saw little progress for half a century, as traditional methods were slow, expensive and required extensive laboratory work or simulative resources, could be solved efficiently by recent DNN approaches [BB23]. Biologists can now predict protein structures from amino acid sequences, to understand cellular processes and for practical design such as optimizing vaccine antigens for stability and effectiveness. This breakthrough has significantly contributed to the fast development of mRNA vaccines, particularly during the COVID-19 pandemic. Recent advancements in generative AI and

Figure 1.1: Image generated by the AI tool "DALL-E", developed by OpenAI, to depict: "How this PhD thesis contributes to be one puzzle piece in the mystery of the universe in 16:9 format".

Natural Language Processing (NLP) [VSP+17] have also led to the development of powerful chatbots, e.g., Chat Generative Pre-Trained Transformer (ChatGPT), which help users master their daily tasks. Similarly, major tech companies like Apple, Google, Microsoft, and Amazon have introduced their own voice assistants—Siri, Google Assistant, Cortana, and Alexa—further integrating AI into everyday life. These advancements have significantly increased public awareness and visibility of AI technology.

In Fig. 1.1, we present an impressive example of how AI can generate a painting with philosophical depth based on a simple command prompt. This is only one illustration of how AI paves the way for many new possibilities.

However, image generation is a completely different task compared to solving problems with AI in a technical context: ML techniques must be adapted carefully to address challenges in communications. These challenges were tackled within the scope of a series of research projects, during which the content of this thesis was developed[1]:

- Momentum—Mobile Medizintechnik für die integrierte Notfallversorgung und Unfallmedizin: The goal was to enable rapid medical diagnosis in ambulances during emergency transport. Reliable and fast wireless resource allocation and transmission of medical information to doctors is of the highest priority.

[1]This work was partly funded by the Federal State of Bremen and the University of Bremen as part of the Humans on Mars Initiative in the seed project HiSE, and by the German Ministry of Education and Research (BMBF) under grant 16KIS1028 (MOMENTUM) and grant 16KISK016 (Open6GHub).

- Open6GHub—6G für souveräne Bürgerinnen und Bürger in einer hochvernetzten Welt: Germany's contribution to the global 6G harmonization and standardization process aims to ensure European technological sovereignty with a broad scope, encompassing security and resilience, 3D networks including satellites and airplanes, connected intelligence for enhanced resource efficiency and joint communication and sensing, flexible network topologies, and the utilization of higher frequency bands. Under the keyword of connected intelligence, one specific goal was to explore new PHY/Multiple Access Channel (MAC) layer concepts such as exploiting the semantic content of transmitted messages to ensure flexibility across diverse 6G transmission scenarios.

- HiSE—Human-integrated Swarm Exploration: The goal was to develop effective communication to facilitate accurate Human Decision-Making (HDM) in human interaction with (mobile) AI robotic systems, reducing the risk of critical failures and enabling efficient remote operation for production and habitation on Mars.

The motivation behind all these projects stems from a broader trend: New applications such as autonomous driving, medical diagnostics, and virtual reality are driving rapid data traffic growth and increasingly specific application demands. These challenges were addressed in the Momentum project in a medical context by prioritization and/or new communication architectures.

However, the increasing data demands cannot be effectively managed through content-agnostic, i.e., digital, communication alone, as it constrains efficiency in terms of bandwidth, power, latency, and complexity trade-offs [XYN+23]. Current systems already operate near the Shannon limit, necessitating a paradigm shift towards integrating semantic content into system design for future wireless communication standards, such as 6G, as explored in the Open6GHub project.

By focusing on the semantic content of the data—the meaning or essential information as required by the application and/or the user to execute a task rather than the exact bits themselves—, semantic communication introduces a novel approach that goes beyond traditional semantics-agnostic transmission methods [Wea49; PSB+20; LWZ+21; SB21; UKE+22; GQA+23]. This shift enables compression and coding techniques that can significantly reduce bandwidth, power consumption, and latency, leading to more efficient wireless communication systems.

The need for careful integration of semantic content into communications becomes evident when humans interact with (mobile) assistance systems, especially in hazardous environments like nuclear accident sites, extraterres-

trial exploration, and automated Industry 4.0 production facilities. Accurate HDM is vital for reducing the risk of critical failures and relies on effective and efficient communication, as investigated in the HiSE project. This includes evaluating the condition of a tool in a manufacturing plant to prevent costly shutdowns or utilizing robotic swarms in extraterrestrial settlements to identify essential resources for survival [SBD+24; BLR+25].

1.2 State of the Art and Open Challenges

The successes of ML have not gone unnoticed by communications engineers [WWW+17], meaning that ML is becoming increasingly important in communications engineering.

Traditionally, engineers build an abstract mathematical model of the environment and derive algorithms within the framework of statistics and information theory [Sim18b]. In wireless communications, we have in general linear models that represent reality well, i.e., that are empirically well-founded and abstract the underlying physical processes up to standard processing steps. This is because the electromagnetic propagation medium of air can be regarded as linear and radio frequency frontend devices including non-ideal or non-linear components like amplifiers, mixers, oscillators, filters, and analog-to-digital converters are designed to create conditions where linearity holds approximately true. Nowadays, there exist already many solutions for the design of transmitters and receivers in wireless communication systems based on linearity, stationarity and Gaussian distributions [OH17].

However, this traditional engineering approach faces limitations when one of the following deficits is present:

- **Model deficit:** The traditional approach struggles to account for non-idealities in models, such as those of radio frequency frontend devices, and constrains design choices toward achieving linearity, potentially resulting in suboptimal efficiency. More pronounced is the deficit in modeling fiber-optical and molecular channels [FG18; KCT+18], complicating the development of effective transceiver solutions. Moreover, current digital transceiver technologies are designed without consideration of semantic aspects [Sha48]. Exploiting these aspects requires knowledge of the semantic relationship between an observation, such as an image, and its meaning—like identifying a dog or a cat. These real-world tasks such as pattern recognition can be difficult to model analytically and may be only described by data samples, further highlighting the *model deficit*.

- **Algorithm deficit:** As outlined, in digital wireless communications,

established models, such as Additive White Gaussian Noise (AWGN), accurately represent reality and enable the development of optimized algorithms. However, these optimized algorithms can be too complex for practical implementation, leading to what is termed an *algorithm deficit*. For instance, in massive or large Multiple Input Multiple Output (MIMO) detection, the complexity of calculating the optimal solution increases exponentially with system size [SDW17; SDW19; BBD21]. Even the efficient implementation of the so-called Maximum A Posteriori (MAP) detector, the Sphere Detector (SD), becomes impractical due to high energy consumption and latency. Further, it is difficult to find a trade-off sweet spot with approximative solutions resulting in compromises.

If the transmitted data is also protected by a channel code, the code structure must be considered for MAP detection. Since the code length usually exceeds the number of antennas, the problem dimensions grow significantly, making it evident that these problems should be addressed separately. As a result, communication systems typically break down the overall problem into smaller, mostly independent sub-problems [OH17]. These sub-problems can then be solved efficiently and optimally. While the potential for improvement through joint design is partially exploited—e.g., via joint iterative equalization and decoding with soft information propagation—a unified, end-to-end optimization approach is usually not considered.

These cases are precisely where ML approaches can play to their strengths: ML algorithms in their most basic form do not need a model to capture their environment, these rely solely on data for design. First, an architecture such as a DNN and the training settings are selected. Then, the optimal parameters of the model are learned based on the available data that can cover unforeseen imperfections. This adaptability is a key advantage of ML, but it also offers a second benefit targeting the model deficit: By choosing an architecture with low complexity, it is possible to design practical algorithms that are energy-efficient and have low latency.

Meanwhile, complete transmitter and receiver structures have been learned by interpreting transmitter, channel and receiver as an AutoEncoder (AE) which is trained end-to-end similar to one DNN [OH17]. The insights gained from a simple AWGN scenario showing Bit Error Rate (BER) performance equal to that of handcrafted systems have aroused great interest in communications engineering.

A downside of ML approaches is the increased demand for data and computing power: If existing, highly optimized solutions are simply replaced by AI applications—even when using ever-improving GPUs for efficient

training—this can greatly increase power consumption and latency. Another drawback of ML is that it does not align with the traditional, systematic methodology of engineering. Training DNNs is often perceived as an art, lacking clear guidelines for selecting architectures and training settings, both of which significantly impact the achievable performance and training convergence. Furthermore, in the end it is not clear how the solution was arrived at, if it is the global solution, and what it actually means. The lack of in-depth interpretability, also in the mathematical sense, is a problem that, judging by the current state of research, is still very poorly understood.

Considering the strengths and limitations of both traditional and ML techniques, the question arises whether rethinking conventional wireless communication systems or transmitter and receiver structures through the lens of ML could lead to new architectures that achieve a better balance between performance, energy consumption, and latency.

For a comprehensive review on the State of the Art (SotA) in applying ML techniques to communications since the beginning of this dissertation project, we refer the reader to Sec. 3.2 and Sec. 4.2.

1.3 Thesis Objectives

We have seen that one of the key challenges lies in the careful integration of new ML concepts into communication systems, as their blind application does not lead to success. Well-thought-out approaches, developed over multiple generations of mobile communication standards, cannot be simply replaced by a "data-is-everything" mindset without sacrificing some of their benefits.

In this thesis, therefore, we aim to explore how recent promising ML approaches, such as DNNs, can be leveraged to enhance traditional communication design. The primary research objectives identified are:

1. **Establish a Common Theoretical Framework:** Identify a unified theoretical foundation for integrating communications and ML techniques, such as Variational Inference (VI) and DNNs.

2. **Optimize Model-Based Communication Design:** Investigate ways to effectively improve traditional model-based digital communication design by incorporating ML techniques while avoiding common drawbacks, such as increased data and computational requirements, which can lead to higher latency. Explore how to combine the strengths of both model-based and data-driven approaches.

3. **Incorporate Semantic Aspects into Communication Design:** Develop a theoretical framework for integrating semantic aspects into communication systems and leverage ML to design a semantic communication system. Validate the potential of the system in diverse application areas.

1.4 Thesis Style and Structure

This dissertation adopts a *cumulative format*, with its core built around multiple original publications that directly address the stated objectives. To reflect on the larger context and uncover the interdependencies of the included publications, a detailed discussion of the overall contributions is provided in Sec. 1.5. The chapters of this thesis are classified into two categories:

- Chapters starting with *Publication* are a faithful reproduction of the original papers reformatted to A5.
- The other chapters, including the appendices, extend the contents of the publications and place them in a larger context.

Furthermore, in line with the objectives, the thesis can be divided into two thematic parts

- Part I: Machine Learning for Digital Communications
- Part II: Semantic Communication

each dedicated to the challenge of an algorithm deficit and a model deficit, respectively.

Fig. 1.2 illustrates the logical structure of the dissertation as well as the interpendencies within it, presented in the form of a block diagram. We show the main contributions, i.e., innovative approaches, inside green boxes. These exploit, enrich, and extend preexisting approaches, represented within gray boxes. These include original ML techniques, communication pioneering works, i.e, SotA approaches where ML is applied to communications, early semantic information theory, and HDM models from psychology. Preexisting approaches and innovations can be categorized thematically into the two dissertation parts, i.e., different communications areas, and into different ML branches, i.e., *Supervised*, *Unsupervised* and *Reinforcement* Learning as well as the broader category *Tricks*. Arrows indicate the influences/flow of new ideas starting from preexisting works. On these arrows, crucial changes in model assumptions are presented via formulas. Surface-level advantages

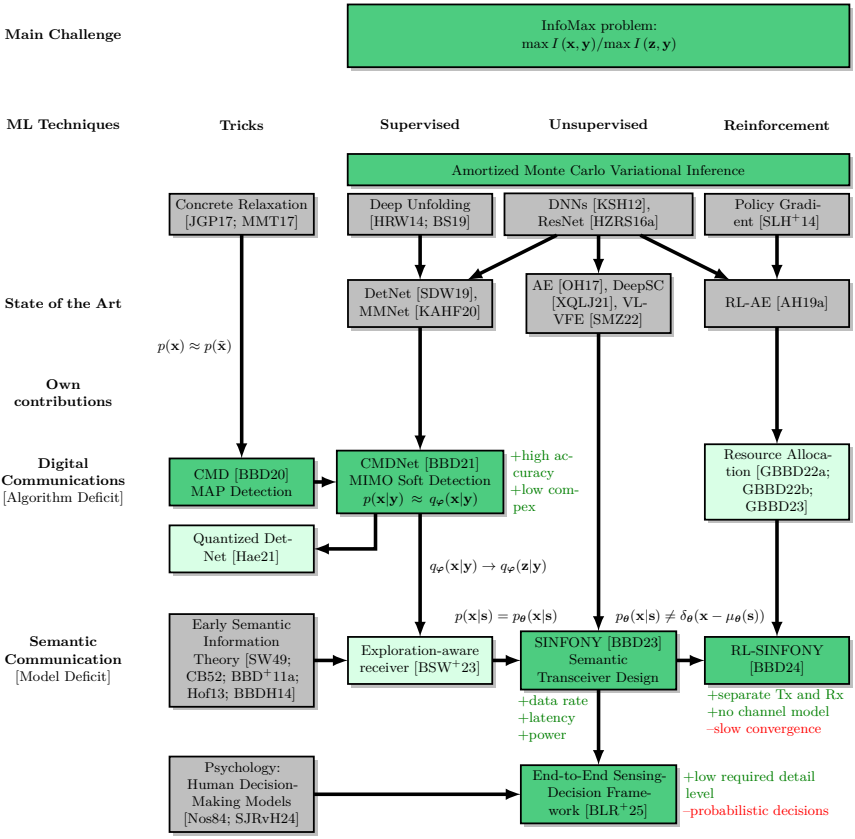

Figure 1.2: Thesis overview: Logical structure with categories and main challenge as well as interdependencies of ideas. Main thesis contributions: green. Influencing ideas: gray. Arrows: flow of ideas and model changes.

and drawbacks are given next to the green contribution boxes. Light green boxes indicate contributions that are inside the scope of the dissertation but are not shown here for the sake of extent. To understand the details such as formulas and abbreviations, we expect the reader to be familiar with the contents of the chapters.

1.5 Contributions

1.5.1 Main Contributions

The main contributions of this thesis, directly aligned with the stated objectives and the block diagram from Fig. 1.2, are:

1. **Information Maximization (InfoMax) framework:** We identified the InfoMax problem as the main challenge when optimizing components of a communication system [BBD21; BBD23]. As the central solution approach, we elaborate a lower bound and Amortized Monte Carlo variational inference from a theoretical perspective.

2. **Data-driven theory of MIMO soft detection + CMDNet approach**: We extend model-based massive MIMO soft detection towards a hybrid approach Concrete MAP Detection Network (CMDNet) combining strengths of model-based and data-driven approaches using the concrete relaxation and deep unfolding, while providing a unified theoretical view on soft detection [BBD20; BBD21; Bec23]. Furthermore, we shed light on how common ML training practices are challenged by communication theory and how an algorithm deficit instead of a model deficit changes the view.

3. **Semantic communication theoretical foundation + SINFONY approach:** Extending beyond former pioneering efforts [Wea49; BBD$^+$11a; BBDH14], we enrich the familiar concepts of information theory in communications to include semantic information and identify the InfoMax principle and its variation, the Information Bottleneck (IB) problem, to be proper design criteria [BBD23; Bec24]. To tackle the model deficit in semantic communication and to remove the information barrier caused by hard Variable-Length Codes (VLC) source coding in the classic block-wise digital design, we propose a new data-driven approach Semantic INFOrmation TraNsmission and RecoverY (SINFONY).

4. **Diverse semantic communication application areas:** We use the theoretical insights and data-driven approach to make distributed

exploration of a physical process communication-aware and vice versa [BSW+23] and to create a probabilistic end-to-end sensing-decision framework that wirelessly links the sensed data of an assistance system with human decision-making by semantic communication [BLR+25]. To enable online refinement, we extend the data-driven semantic communication design towards separately optimized transmitter and receiver using Reinforcement Learning (RL) techniques [BBD24]. Many simulative investigations underline that SINFONY is superior to classic digital design requiring less bandwidth, power, and latency.

1.5.2 Discussion of Contributions

Since this is a cumulative dissertation, a comprehensive discussion of its contributions and interdependencies is presented. The contributions are diverse and manifold:

- In Chapter 2, we lay the foundation of this thesis by introducing the most important concepts from ML such as Amortized Monte Carlo (MC) VI and powerful DNN models. These fundamentals are presented from a unique view with a unique notation connecting ML concepts and communications design closely through the lens of information theory instead of traditional ML practices. One of our central insights of this thesis is that we can use the InfoMax principle as an overall learning framework for receiver learning and transceiver design. Based on this principle, we derive the Maximum Likelihood (MaxL) criterion — commonly used for optimization in communications — as a special case of the InfoMax criterion and Kullback–Leibler (KL) divergence minimization. Most notably, we reflect on the approaches used in this thesis in the more general context of ML theory — illuminating their background and interconnections within the web of ML concepts — thereby motivating both their choice and possible alternatives. For instance, we derive the Mean-Field Variational Inference (MFVI) solution for MIMO detection providing numerical results that reveal the suboptimality of the Information (I)-compared to the Moment (M)-projection and argue that overfitting — i.e., decreased generalization performance due to a Monte Carlo (MC) approximation with too few samples — remains a valid concern, even though the recently discovered double descent phenomenon suggests improved generalization for large-capacity models.
- In Chapter 3, i.e., the publication [BBD21], we propose a hybrid approach to overcome the algorithm deficit in soft detection with large

system dimensions, e.g., in digital massive MIMO systems. Inspired by recent ML research, we first introduce a CONtinuous relaxation of the prior probability mass function (pmf) of the disCRETE Random Variables (RVs) by a probability density function (pdf) from [JGP17; MMT17] to the MAP detection problem [BBD20]. The benefit of this concrete relaxation is that the degree of approximation, which becomes exact for a limit value, can be controlled directly via a hyperparameter. With our new approach Concrete MAP Detection (CMD) [BBD20], we replace exhaustive search by computationally cheaper continuous optimization to approximately solve the MAP problem in any differentiable probabilistic non-linear model by gradient descent. Second, we combine the strengths of model- and ML-based approaches to unfold the gradient descent iterations into our new DNN-like model Concrete MAP Detection Network (CMDNet) [BBD21]. Following this so-called idea of deep unfolding, we can optimize a few parameters given a fixed number of iterations to further improve detection accuracy while limiting complexity. The low parameter number allows dynamic adjustment to different working points and potentially fast online training. Thirdly, given the information-theoretic perspective from Chapter 2, we derive the classification optimization criterion from the KL divergence. We show that we learn offline an approximation of the Individual Optimal (IO) detector avoiding the drawback of high computational training complexity typical for purely data-driven ML approaches. By doing so, we are able to provide detection probabilities, i.e., reliable soft outputs, to account for subsequent decoding, e.g., in MIMO systems, in contrast to literature [SDW19; KAHF20]. Numerical results of MIMO systems including a variety of simulation setups, e.g., correlated channels, demonstrate CMDNet to be a generic and promising approach competitive to SotA and superior to other recently proposed ML-based approaches. Notably, simulations in coded systems reveal CMDNet's soft outputs to be reliable for decoders in contrast to [SDW19]. In conclusion, the hybrid CMD approach features a promising trade-off between detection accuracy and complexity.

- In Appendix A, we extend the analysis on CMDNet providing the complete derivation of binary CMD extending beyond [BBD20], and prove that CMD and binary CMD are different algorithms for BSPK symbols. Most notably, we provide insights how we proceed systematically in ML optimization to arrive at the CMDNet solutions of [BBD21]. We discuss different training aspects such as the influence of hyperparameter choices including the optimization algorithm, batch size and number of layers in the context of established ML practices. We shed light

on how these practices are challenged by communication theory and how an algorithm deficit instead of a model deficit changes the view. For example, we figure out that using Adaptive Moment Estimation (Adam) as the optimization approach — known to generalize worse than Stochastic Gradient Descent (SGD) — indeed leads to comparable performance with infinite model-generated data, i.e., an algorithm deficit. Further, after introducing the cross-entropy loss as a measure of soft information to enable deeper investigations, we state that the main benefit of validation loss, i.e., tracking the generalization performance, transforms into being a less noisy observation of the current training progress having an algorithm deficit with infinite model-based generated data. Moreover, we show the unexpected result that optimization of CMDNet with respect to (w.r.t.) Mean Square Error (MSE) loss like related approaches leads to comparable performance — mainly with a different Signal-to-Noise Ratio (SNR) weighting with an excellent BER in high SNR regions. This explains why corresponding approaches do work well, although this practice is not properly motivated from a theoretical view. Further, we reveal CMDNet's sensitivity to starting weight initializations, requiring heuristics different from standard DNN practice, and that evaluating its training convergence in combination with offline training is not insightful. A first simple online learning investigation shows that a default low-complex DNN is not able to achieve competitive performance with reasonable training complexity and only a small performance increase with CMDNet. We explain that a mismatch of CMDNet parameters can be considered as overfitting and show CMDNet's robustness. Finally, we propose two extensions of CMDNet to overcome design flaws of non-convexity, i.e., Parallel Concrete MAP Detection (CMDpar) and Hypernetwork-based Concrete MAP detection (HyperCMD), to improve accuracy.

- In Chapter 4, i.e., the fundamental publication [BBD23], we contribute to the theoretical modeling and problem formulation in semantic communication, extending beyond former pioneering efforts [Wea49; BBD+11a; BBDH14]. In particular, we adopt the terminus of a semantic source and include it in communications' complete Markov chain, including semantic source, communications source, transmit signal, communication channel, and received signal. By doing so, we enrich the familiar concepts of information theory in communications to include semantic information and identify the InfoMax principle and its variation, the IB problem, to be proper design criteria, as outlined in Chapter 2. Our take on semantic communication does not differentiate but merges all Levels A, B, and C according to Weaver compared

to literature [XQLJ21; BSW+23] and aims to encode the semantic information contained in an observation such that the meaning at the receiver is best preserved. This perspective is different from the IB view in [SMZ22], and we further exploit different ML-based solution approaches, i.e., we maximize approximately the Mutual Information (MI) for a fixed encoder output dimension that bounds the information rate. To tackle the model deficit in semantic communication, we propose a new data-driven approach Semantic INFOrmation TraNsmission and RecoverY (SINFONY). We design and evaluate it for a distributed multipoint scenario where the meaning behind multiple observations, i.e., images, at different senders are communicated to a single receiver for semantic recovery of its contents. This scenario is different compared to [AZ21; SMZ23] in that we include the communication channel. Numerical experiments demonstrate that SINFONY achieves significant savings in both data rate and power consumption compared to classic digital communication.

- In Chapter 5, the publication [BBD24], we tackle the practical problem that data-driven ML approaches such as the AE or SINFONY typically used in semantic communication exhibit a high online learning complexity and are difficult to adapt with an unknown channel once deployed. Hence, we extend the data-driven semantic communication design towards separately optimized transmitter and receiver using RL techniques, i.e., the Stochastic Policy Gradient (SPG). In particular, we derive the application of the SPG for both classic and semantic communication from the InfoMax principle in contrast to [AH19a]. Additionally, no known or differentiable channel model is required for Reinforcement Learning-based SINFONY (RL-SINFONY). This allows for online refinement of the semantic design. Numerical evaluations in the distributed SINFONY scenario show performance comparable to a channel model-aware approach, albeit with a decreased convergence rate.

- In Appendix B, we delve deeper into key aspects of semantic communication from Chapter 4 and Chapter 5. Considering philosophical and interdisciplinary views, we show that meaning, i.e., semantics, is closely related to phenomenon of emergence in the universe and that there are multiple semantic hierarchical levels that can be viewed through the lens of Shannon's information theory. Further, we extend the simulative comparison of SINFONY and classical digital communication design to the exemplary dataset CIFAR10. The performance gap w.r.t. all competing approaches prove semantic communication to be even

more effective in more challenging scenarios. Moreover, we shed light on our design choice by showing performance results of alternative SINFONY designs. For example, using separate Rx modules for each received signal only lead to small gains reflecting that we made proper assumptions about image processing and channel models. Most notably, we reveal where the difficulties in introducing semantics to a classic digital design lie, i.e., an information barrier due to hard VLC source coding. The key insight follows to remove the block-wise structure like in the semantics-tailored design SINFONY. Lastly, we elaborate on the differences between the system models of Chapter 4 and [BSW+23], provide a different semantics example of floating-point bits transmission validating the numerical results of [BSW+23], and present ideas to overcome the slow training convergence of RL-SINFONY as well as how to train it with a non-differentiable objective function measuring semantic similarity.

- In Chapter 6, the publication [BLR+25], we propose a probabilistic end-to-end sensing-decision framework that wirelessly links the sensed data of an assistance system to HDM by semantic communication. In this context, semantic communication conveys the meaning behind the sensed information relevant to HDM when humans perform assistance-supported tasks. Integrating interdisciplinary perspectives from communications and psychology, the framework aims to enhance our understanding of how semantic communication impacts HDM and can improve its effectiveness. To investigate this interplay, we model HDM as a cognitive process and reveal both in theory and simulations the fundamental design trade-off between maximizing the relevant semantic information and matching the cognitive capabilities of the HDM model. Using SINFONY and the HDM model of Generalized Context Models (GCMs) on specific datasets, our initial analysis demonstrates how semantic communication can balance the level of transmitted information detail in feature extraction with human cognitive capabilities while demanding less bandwidth, power, and latency compared to classical Shannon-based methods. Notably, our findings reveal that increasing information does not always enhance decision accuracy. Finally, we discuss challenges for future research, including the design of effective information presentation through visualization and exploring game-theoretic approaches to address sender-receiver conflicts of interest. **Own contribution:** In [BLR+25], as the first authors, we provided the foundation by contributing the content on semantic communication and the overall theoretical framework with its extension to both presentation and HDM optimization. This includes its concep-

tualization, description and evaluation, along with the development of a software interface for the GCM [Bec24]. The psychology-related sections on HDM, the selection and simulation of the GCM, and a few identified challenges were developed collaboratively, with primary input from the other authors.

For reproducibility of our research, the simulation software of CMDNet and SINFONY is available at [Bec23; Bec24].

1.6 Thesis Publications

[BBD20] E. Beck, C. Bockelmann, and A. Dekorsy, "Concrete MAP Detection: A Machine Learning Inspired Relaxation," in *24th International ITG Workshop on Smart Antennas (WSA 2020)*, Hamburg, Germany: VDE VERLAG, Feb. 2020, pp. 1–5.

[BBD21] E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection With Low Complexity," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8214–8227, Dec. 2021. DOI: 10.1109/TCOMM.2021.3114682.

[BBD23] E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic Information Recovery in Wireless Networks," *Sensors*, vol. 23, no. 14, p. 6347, Jul. 2023. DOI: 10.3390/s23146347.

[BBD24] E. Beck, C. Bockelmann, and A. Dekorsy, "Model-free Reinforcement Learning of Semantic Communication by Stochastic Policy Gradient," in *1st IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2024)*, Stockholm, Sweden, May 2024, pp. 367–373. DOI: 10.1109/ICMLCN59089.2024.10625190.

[Bec23] E. Beck, *Concrete MAP Detection Network (CMDNet) Software*, Zenodo, version v1.0.2, Oct. 2023. DOI: 10.5281/zenodo.8416507.

[Bec24] E. Beck, *Semantic Information Transmission and Recovery (SINFONY) Software*, Zenodo, version v1.2.2, Dec. 2024. DOI: 10.5281/zenodo.8006567.

[BLR+25] E. Beck, H.-Y. Lin, P. Rückert, Y. Bao, B. von Helversen, S. Fehrler, K. Tracht, and A. Dekorsy, *Integrating Semantic Communication and Human Decision-Making into an End-to-End Sensing-Decision Framework*, arXiv preprint: 2412.05103, Mar. 2025. DOI: 10.48550/arXiv.2412.05103.

Other notable works

Last but not least, we shortly mention other notable works we published during the time of the dissertation:

- We published the results of my master thesis [Bec17] on Compressive Sensing – Spectral Estimation for Cognitive Radios in two conferences [BBD17; BBD18] and one journal article [BBD19]. There, we exploit the sparseness of edges in the power spectrum to define a new compressive sensing problem demonstrating high detection accuracy of occupied given a fraction of the original samples [BBD17; BBD18]. Further, we present practical results using Software Defined Radio (SDRadio) hardware with over-the-air-measurements, as well as a demonstration on Wi-Fi and Bluetooth signals [BBD19].

- In the supervised master thesis [Hae21], we exploit the learning compression algorithm to constrain the weights of Detection Network (DetNet) [SDW19], a generic DNN MIMO detector, to be within a predefined codebook that effectively translates into bit-shift operations. In particular, we use a Lagrangian augmentation with a quadratic penalty. In [AH19b], the idea is applied to generic DNN-based AE transceiver design. Results show that we can maintain the detection accuracy high while significantly decreasing complexity using bit-shift operations and coarse fixed-point arithmetic.

- Further, the results of a supervised master thesis [Gra20] on the application of RL techniques to resource allocation problems were published in [GBBD22a; GBBD22b; GBBD23]. Deep Q-Networks are used to optimize switching between different model-based resource scheduling algorithms [GBBD22a] whereas Deep Deterministic Policy Gradient (DDPG) with continuous action spaces is used to learn entirely new scheduling algorithms with special regard to priority users [GBBD22b]. Both achieve high performance on a flexible sum-utility goal. The technique of weight anchoring is used in [GBBD23] to find a solution that is nearby the solution of another learning problem to fixate desired behavior. Thereby, infrequent priority messages are not unlearned as proven by simulations.

- In the publication [BSW+23], we propose a framework to couple "exploration-aware" communication and a "communication-aware" exploration tightly using probabilistic ML techniques. To obtain a mutually-aware design, we model the physical process to be explored by multiple agents by means of Factor Graphs (FGs) and design

ML-based "communication-aware" swarm exploration algorithms that follow active inference principles. To link exploration to communication, we transmit the semantics, i.e., messages of the factor node describing key distribution parameters of the exploration RV to be exchanged between neighboring agents. Since we provide a probabilistic estimate instead of raw data that the message passing algorithm can integrate seamlessly, this can be seen as a first step towards a semantic design. By a "tight" integration of the communication chain, we enable the exploration strategy to balance the inference objective of the swarm with inter-agent communication. The design considerations can also be applied to semantics-agnostic settings and seen as an instance of Joint Source-Channel Coding (JSCC), a view meanwhile supported by semantic communication research [XQLJ21; GQA+23]. Based on a first numerical example with a semantic receiver tailored to digitally transmitted data from distributed full waveform inversion, we demonstrate that simply adapting the receiver to account for semantics yields a notable semantic performance gain. Further, we can achieve near-optimal semantic performance with a DNN of low complexity. Replacing the "classical" transmission, we can thus reduce the cost in terms of required data rate, latency, power and complexity while preserving the desired functionality of the whole distributed system.

- In the scope of the HiSE project [SBD+24], we propose a conceptual framework for integrating humans with a multi-agent robotic system for hazardous remote exploration and maintenance tasks, e.g., on Mars. Key challenges include the scarcity and unknown distribution of resources, limited processing power, and the need for fast decisions with minimal latency. To address these, the framework incorporates semantic communication to efficiently transmit and visualize relevant exploration data from rovers to human operators, aiding in decision-making.

- Semantic communication typically adapts communication to specific meanings or tasks, limiting its application to single use cases. To support multiple tasks, the work [HTB+25] extends the InfoMax framework of [BBD23] with SINFONY by multiple semantic interpretations in the semantic source from [HBD24]. To facilitate cooperative multitask processing and improve training convergence, the semantic encoders are divided into common and specific units, extracting common low-level features and separate high-level features. Simulation results on the numerical example of distributed sensed observations from [BBD23] highlight the effectiveness of this approach in scenarios

with statistical relationships, comparing cooperative and independent task processing.

1.7 Nomenclature

We present the basic notation we use throughout this thesis. For a complete list of all symbols, we refer the reader to the glossary at the end of this dissertation.

- A RV is denoted by a lower case italic letter a.
- A column vector is denoted by a lower case bold letter $\mathbf{a}$.
- A matrix is denoted by an upper case bold letter $\mathbf{A}$.
- We extract elements of a matrix by a_{nm} or $[\mathbf{A}]_{n,m}$. The columns are extracted by $\mathbf{A}_{*,j}$ and rows by $\mathbf{A}_{j,*}$. Sets of matrices are indexed using subscripts without comma separation, e.g., $\mathbf{A}_{nm}$.
- Realizations are denoted by sanserif letters, e.g., a.
- For better readability, we simplify the notation of pdf $p_{\mathbf{a}}(\mathbf{a}) = p(\mathbf{a} = \mathbf{a})$ of the RV $\mathbf{a}$ to $p(\mathbf{a})$ and use the same notation for a pmf.
- $\mathrm{E}_{\mathbf{a}\sim p(\mathbf{a})}[f(\mathbf{a})]$ denotes the expected value of $f(\mathbf{a})$ with regard to both discrete or continuous RVs $\mathbf{a}$.
- $\mathcal{H}(p(a))$ or $\mathcal{H}(a)$ denotes the Shannon entropy of $p(a)$ and $D_{\mathrm{KL}}(p \parallel q)$ the KL divergence between $p(a)$ and $q(a)$.
- $I(\mathbf{x}; \mathbf{y})$ denotes the MI between multivariate RVs $\mathbf{x}$ and $\mathbf{y}$.
- The natural logarithm is denoted as $\ln(\cdot)$ whereas $\log_2(\cdot)$ is the logarithm w.r.t. the basis 2. Both logarithm and exponential functions $\exp(\mathbf{a}) = e^{\mathbf{a}}$ are applied element-wise to vectors.
- $|\mathcal{A}|$ denotes the cardinality of the set $\mathcal{A}$.
- $|\mathbf{a}|$ denotes the absolute values of $\mathbf{a}$.
- $\|\mathbf{a}\|_j$ denotes the j-norm of $\mathbf{a}$, defaulting to $j = 2$.
- $\arg\max \bullet$ and $\arg\min \bullet$ denotes the argument that maximizes or minimizes the expression $\bullet$.

- $\mathbf{0}$ and $\mathbf{1}$ are the all-zero and all-one matrices or vectors, respectively. $\mathbf{I}$ is the identity matrix.
- $\mathbf{A}^T$ and $\mathbf{A}^H$ denote the transpose and hermitian of a matrix, respectively.
- The partial derivative of function $f(\mathbf{a})$ w.r.t. $\mathbf{a}$ is denoted as $\frac{\partial f(\mathbf{a})}{\partial \mathbf{a}}$.
- $\operatorname{diag}\{\mathbf{a}\}$ denotes the diagonal operator placing the vector entries on the diagonal of an all-zero matrix.

Chapter 2

Fundamentals of Machine Learning for Communications

In this chapter, we lay the theoretical foundation of the thesis. We explain basic Machine Learning (ML) concepts crucial to understand the contents and contributions of this thesis.

2.1 Overview

Machine Learning (ML) is a broad research area involving lots of different techniques [Bis06; Sim18a] and a subset of Artificial Intelligence (AI). While AI refers to the broader concept of machines or systems that can perform tasks that typically require human intelligence, such as reasoning, problem-solving, and understanding language, ML focuses specifically on enabling systems to learn from data.

At a first glance, the different ML techniques seem like a mixed collection of unrelated approaches. Looking more closely, a common foundation with their root in probability theory can be revealed: *information theory*. Originally, Shannon introduced this theory to describe communication of information in 1949 and thus termed it communication theory [Sha48]. Therefore, information theory lies at the heart of both ML and communications. The probabilistic view of information theory crucially influenced research on AI and has paved the way for many advances in both theory and practice in recent years. A detailed description of how recent ML advances influenced

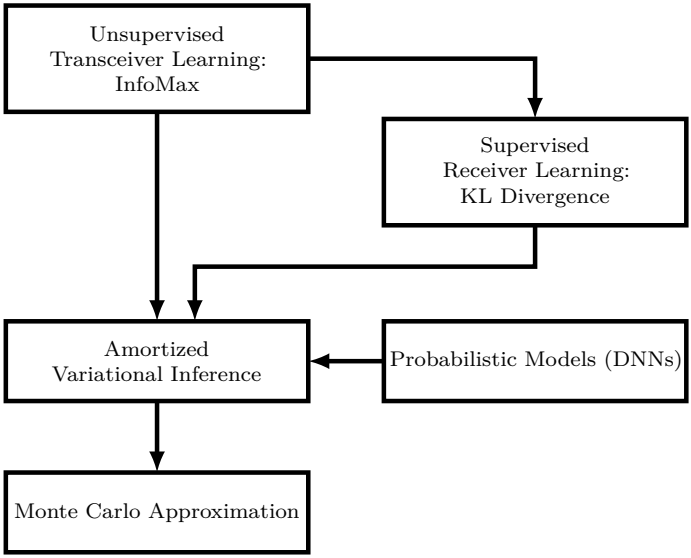

Figure 2.1: Main ML concepts of this thesis and their relation to each other.

communications research can be found in Chapter 3.

In the following section, we explain the most important terms in ML and in particular those used in this thesis to equip the reader with what is needed to understand the fundamentals and insights provided here. Then, we want to answer the question of what learning actually is and describe the basic problem of learning, the task at the core of ML, via information-theoretic measures.

The key principles of ML for communications applied in this thesis are shown in Fig. 2.1. For the learning of a whole transceiver, we propose to use the Information Maximization (InfoMax) principle. From InfoMax, we can derive the Kullback–Leibler (KL) divergence minimization as the receiver learning criterion. Deviating from the shown scheme, we start with the latter concept in Sec. 2.3 and crucial ML terms in Sec. 2.2 for a better introduction of the key principles, and extend these towards the more general InfoMax principle in Sec. 2.6. We use amortized Variational Inference (VI) (see Sec. 2.4) and probabilistic models including Deep Neural Networks (DNNs) (see Sec. 2.7) to overcome *algorithm deficits* when it comes to digital communications (see Chapter 3, Appendix A) and *model deficits* for semantic communications (see Chapter 4, Chapter 5, Appendix B). This combination further requires Monte Carlo (MC) methods from Sec. 2.4.5

which can be used if only a finite number of samples is available.

2.2 Taxonomy

ML problems rely on *inference*, i.e., the process of using a trained model to make predictions based on new, unseen data. In ML, there are two approaches for modeling [Sim18a]:

- **Generative Models:** The observation distribution $p(y)$ or the joint distribution $p(x, y)$ between target variable $x \in \mathcal{M}_x$ and observation variable $y \in \mathcal{M}_y$ from domain $\mathcal{M}_x$ and $\mathcal{M}_y$ is modeled and learned. In both cases, we model the observation distribution $p(y)$ such that we can *generate* realizations of observations. The predictive distribution $p(x|y)$ is inferred from the joint distribution by application of Bayes' theorem. We note that in communication designs a *directed generative* [Sim18a] and probabilistic system model $p(x, y) = p(y|x) \cdot p(x)$ is often used.
- **Discriminative Models:** The predictive inference distribution $p(x|y)$ is directly modeled and learned. We can directly *discriminate* the target variable x based on the posterior.

The main three types of ML problems are [Bis06; Sim18a; BB23]:

- **Supervised Learning:** An observation or input variable y is mapped onto a label or target variable x. The goal is to learn this mapping given a dataset or probabilistic model $p(x|y)$. We explain this in more depth in Sec. 2.3, Sec. 2.5 and Sec. 2.7. Supervised learning distinguishes two type of problems with respect to (w.r.t.) the target variable x, which have a counterpart in the communications domain (the communications term is given in brackets):
  - **Classification (Detection):** The target variable x is from a discrete set, i.e., $x \in \mathcal{M}$, and the task is to classify the observation y into one of the categories in $\mathcal{M}_x = \mathcal{M}$.
  - **Regression (Estimation):** The target variable x is continuous, e.g., $x \in \mathbb{R}$ with $\mathcal{M}_x = \mathbb{R}$, and the task is to estimate it from y.
- **Unsupervised Learning:** Only unlabeled data of the observation y is available, and the goal is to leverage similarities and dissimilarities among data points to learn patterns and structures. This can be achieved through techniques such as clustering, dimensionality reduction, representation learning and generative modeling, which includes

learning the properties of the underlying generative mechanism, e.g., $p(y)$. We refer the reader to Sec. 2.6 about the InfoMax principle that concludes with a comparison between discriminative and directed generative modeling in unsupervised learning.

- **Reinforcement Learning (RL):** The goal is to learn optimal sequential actions x given an observation y of the environment to maximize a reward [GBBD22b; GBBD23]. Interacting with the environment changes its state, meaning the history of actions must be considered. In this thesis, the use of Reinforcement Learning (RL) is derived from unsupervised learning in Chapter 5, revealing how close the underlying ML concepts are interrelated [BBD24].

Furthermore, there are two essential learning paradigms, each differentiated and explicitly leveraged in Chapter 3 and Appendix A [Sim18a]:

- **Offline learning:** The model is trained offline in a single pass with high training complexity using an entire dataset, capturing statistical properties of all use cases. Once trained, it is deployed for inference with low runtime complexity.

- **Online learning:** The model is trained incrementally, updating with small batches of data as new data arrives. This approach enables the model to adapt to changes in the data distribution or to specific subset statistics, reducing potential mismatch and improving performance, respectively. However, it introduces additional training complexity during inference and carries the risk of catastrophic forgetting of prior knowledge [GBBD23]. The performance difference between online and offline training is known as the *amortization gap* (see Sec. 2.4.2) — the price paid for achieving efficient deployment with offline learning.

The two main motivations to apply ML techniques in lieu of the conventional engineering flow are [Sim18b]:

- **Model deficit:** There is insufficient domain knowledge or a lack of a physics-based mathematical model, making it difficult to apply model-based approaches.

- **Algorithm deficit:** Even if a (complex) model exists, the algorithms derived to solve the specific problem may be computationally intractable. Using efficient learning models such as Deep Neural Networks (DNNs), may yield algorithms of low complexity.

2.3 Basic Problem of Learning: Approximate Inference

What is learning? For humans, learning at an abstract level means improving in a certain task after observing several trials of it. For example, a child gradually gains control over their feet by interacting with them, a process known as learning by doing. In the physical world, a task is embedded into a complex environment with many interdependencies, most of which we as human beings are unable to fully observe due to limited sensory information or time. Even if we could observe everything, processing all this information would require an infinitely complex brain. However, probabilistic models can capture the physical world with sufficient accuracy by focusing on the most relevant factors while managing uncertainties, noise, and variations without needing to account for every detail. Thus, a probabilistic description of the task's underlying phenomena, excluding minor influences as stochastic variations, naturally suggests itself. Besides natural intelligence, this is also true for AI.

Following this discussion, uncertainty can be decomposed into *aleatoric* uncertainty, arising from irreducible noise that cannot be attributed to any cause, and *epistemic* uncertainty, which reflects a lack of knowledge and can be reduced with sufficient learning. In an idealized world with no inherent randomness, aleatoric uncertainty vanishes, and all uncertainty becomes epistemic—and thus fully learnable. *In conclusion, reducing uncertainty in the probabilistic description is the essence of learning, leading to more reliable inferences and actions.*

Interestingly, probabilistic descriptions are also commonly used in communications, where the uncertainty about data and noise sources must be managed effectively. This approach dates back to 1948, when Claude Shannon proposed a landmark paper titled "A Mathematical Theory of Communication" [Sha48]. Today, his ideas have formed the broad field of information theory which has driven research in the past and made several technologies such as wireless communications possible. The common probabilistic viewpoint shows the close connection between communications and ML in their theoretical foundation and roots.

2.3.1 Inference

Probabilistic View: To capture the thoughts about the probabilistic viewpoint on learning more precisely and mathematically, let us assume a Random Variable (RV) $x \in \mathcal{M}_x$ from domain $\mathcal{M}_x$ distributed according to a probability density function (pdf) or probability mass function (pmf)

$p(x)$. In the following, we will use the term pdf interchangeably with pmf for the sake of concise explanation. Our task shall be *inference*, i.e., to make predictions or to infer the value of x.

In communications, this could for example mean to infer the transmitted signal x at the receiver. For a discrete symbol alphabet, i.e., $x \in \mathcal{M}$, given the Maximum A Posteriori (MAP) criterion, this is done by choosing the element x of the set $\mathcal{M}$ with the highest probability which is commonly referred to as the task of detection. In order to detect a symbol successfully, we have to know the pdf $p(x)$.

Information-theoretic View: Successful detection depends on the degree of uncertainty: If uncertainty is maximum, i.e., all symbols from $\mathcal{M}$ have equal probability, we cannot make any prediction about x. If we know the symbol, uncertainty is zero. A measure of uncertainty reflecting these thoughts is given by the Shannon entropy

$$\mathcal{H}\left(p(x)\right) = -\sum_{x\in\mathcal{M}} p(x)\log_2 p(x) \tag{2.1}$$

$$= \mathrm{E}_{x\sim p(x)}[-\log_2 p(x)] \tag{2.2}$$

from information theory and is usually measured in "bits". In (2.2), $\mathrm{E}_{x\sim p(x)}[f(x)]$ denotes the expected value of $f(x)$ with regard to both discrete or continuous RV x. In the following, we will make use of the natural logarithm and of the unit "nats", as this simplifies the upcoming analysis. Furthermore, $\mathcal{H}\left(p(x)\right)$ and $\mathcal{H}\left(x\right)$ will be used interchangeably throughout this thesis. In the example of detection or estimation, the entropy should be rather low to decrease uncertainty.

Bayesian Inference: For brevity, we have so far defined distributions over single RVs x. However, all considerations extend naturally to distributions over multivariate RVs $\mathbf{x} \in \mathcal{M}_x^{N_x\times 1}$ and $\mathbf{y} \in \mathcal{M}_y^{N_y\times 1}$ from domain $\mathcal{M}_x$ and $\mathcal{M}_y$.

Probabilistic inference in communications is typically based on a *directed generative model* — a known joint distribution $p(\mathbf{x},\mathbf{y}) = p(\mathbf{y}|\mathbf{x})\cdot p(\mathbf{x})$ with forward model or likelihood $p(\mathbf{y}|\mathbf{x})$ and prior distribution $p(\mathbf{x})$. To infer the unobserved variable $\mathbf{x}$ given an observed variable $\mathbf{y}$, we need to compute the posterior distribution $p(\mathbf{x}|\mathbf{y})$ according to Bayes' theorem:

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x})\cdot p(\mathbf{x})}{p(\mathbf{y})}\,. \tag{2.3}$$

The latter is used in communication systems for optimal Maximum A Posteriori (MAP) design to estimate the probability of each possible sent symbol to be the original symbol given the received signal. This stochastic approach enables to not only make point estimates of the most probable symbols, but to reflect also the uncertainty via distribution $p(\mathbf{x}|\mathbf{y})$, which can be exploited by advanced algorithms, e.g., soft decoding, as soft information.

Posterior calculation — that we will also use in (2.49) — requires calculation of the marginal

$$p(\mathbf{y}) = \sum_{\mathbf{x}\in\mathcal{M}^{N_x}} p(\mathbf{x},\mathbf{y})\,. \tag{2.4}$$

Marginalization becomes computationally demanding if the latent variable $\mathbf{x}$ is from a large alphabet $\mathcal{M}_x^{N_x} = \mathcal{M}^{N_x}$. Even when the calculation of the posterior is not required, we still do not avoid the potential intractable summation or integration w.r.t. $\mathbf{x} \in \mathcal{M}$, which is a common challenge in learning and inference tasks.

In communications, when it comes to symbol detection, this could correspond to having a symbol alphabet of 16-Quadrature Amplitude Modulation (QAM). If we have a vector of symbols $\mathbf{x} \in \mathcal{M}^{N_x}$, the number of symbol combinations grows to $|\mathcal{M}|^{N_x}$ and even a moderate vector length N_x makes it impossible to compute the posterior exactly. The same applies in coding theory, where the size of an alphabet grows exponentially with the length of the code word. Since even MAP detection of moderately sized code words becomes computationally intractable and as most codes work well for long code word lengths, the necessity of an approximation becomes evident.

2.3.2 Approximate Inference and Learning

So far, we only considered the task of inference or prediction based on the model pdf $p(\mathbf{x},\mathbf{y})$ and identified its two main problems:

1. We may *not have exact knowledge* of the true joint pdf $p(\mathbf{x},\mathbf{y})$ and have to learn it, as outlined in Sec. 2.3.

2. Even if we know $p(\mathbf{x},\mathbf{y})$, performing Bayesian inference on it can be *computationally intractable*.

Hence, we have to approximate $p(\mathbf{x}|\mathbf{y})$ in (2.3) by an alternative pdf $q(\mathbf{x}|\mathbf{y})$. This process constitutes *approximate inference* and inherently introduces the notion of *learning*: The closer $q(\mathbf{x}|\mathbf{y})$ aligns with the true pdf $p(\mathbf{x}|\mathbf{y})$, the more we reduce epistemic uncertainty in inference — thereby capturing the essence of learning, as discussed in Sec. 2.3.

2.3.3 Kullback-Leibler Divergence Measure

To quantify how well $p(\mathbf{x}|\mathbf{y})$ is approximated by $q(\mathbf{x}|\mathbf{y})$, we can use information-theoretic divergence measures [Sim18a]. The Kullback–Leibler (KL) divergence between two pdfs $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as:

$$D_{\mathrm{KL}}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) = \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\ln \frac{p(\mathbf{x})}{q(\mathbf{x})}\right] . \tag{2.5}$$

It can be interpreted as the amount of extra information needed to represent data $\mathbf{x} \sim p(\mathbf{x})$ from the true pdf $p(\mathbf{x})$ when assuming $\mathbf{x} \sim q(\mathbf{x})$ and has important properties: Application of Gibbs' inequality reveals non-negativity [Sim18a], i.e.,

$$D_{\mathrm{KL}}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) \geq 0 , \tag{2.6}$$

and that $D_{\mathrm{KL}}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) = 0$ holds if and only if $p(\mathbf{x}) = q(\mathbf{x})$. Furthermore, it is neither symmetric, i.e., $D_{\mathrm{KL}}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) \neq D_{\mathrm{KL}}\left(q(\mathbf{x}) \parallel p(\mathbf{x})\right)$, nor does it satisfy the triangle inequality, which implies that it is not a valid distance metric. Non-symmetry has several theoretical and practical implications, which will become important in Sec. 2.4.1.

Just like the Mean Square Error (MSE) for deterministic functions, the KL divergence can be used to define an optimization criterion for deriving a tight probabilistic approximation $q(\mathbf{x})$ of $p(\mathbf{x})$. **We conclude that minimization of the KL divergence is a suitable optimization criterion for supervised learning of $q(\mathbf{x})$, i.e.,**

$$q^*(\mathbf{x}) = \arg\min_{q(\mathbf{x})} D_{\mathrm{KL}}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) . \tag{2.7}$$

The connection of a learning criterion based on the KL divergence to the fundamental information-theoretic measure of entropy becomes clear if we rewrite the KL divergence into a sum of cross-entropy $\mathcal{H}\left(p(\mathbf{x}), q(\mathbf{x})\right)$ and entropy $\mathcal{H}\left(p(\mathbf{x})\right)$:

$$\begin{aligned} D_{\mathrm{KL}}\left(p(\mathbf{x}) \parallel q(\mathbf{x})\right) &= \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\ln q(\mathbf{x})] - \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\ln p(\mathbf{x})] && (2.8)\\ &= \mathcal{H}\left(p(\mathbf{x}), q(\mathbf{x})\right) - \mathcal{H}\left(p(\mathbf{x})\right) . && (2.9) \end{aligned}$$

The cross-entropy can be interpreted as a measure of uncertainty when assuming q while p is true. Since we defined the basic learning problem (2.7) w.r.t. its approximation q, we can neglect the entropy term $\mathcal{H}\left(p(\mathbf{x})\right)$ independent of q and use the cross-entropy as the learning criterion. The optimization problem underlying learning now reads:

$$q^*(\mathbf{x}) = \arg\min_{q(\mathbf{x})} \mathcal{H}\left(p(\mathbf{x}), q(\mathbf{x})\right) . \tag{2.10}$$

Throughout this thesis, when the distributions p and q are clear from the context, we will adopt the notations $D_{\mathrm{KL}}(p \parallel q)$ and $\mathcal{H}(p, q)$ for brevity.

2.4 Approximate Inference Techniques

Several approaches exist to derive pdf approximations that address the aforementioned inference challenges. In the following sections, we briefly introduce some of the most fundamental *approximate inference* techniques.

2.4.1 Variational Inference

Moment-Projection

As we have seen in Sec. 2.3 from an information-theoretic viewpoint, learning involves minimization of a divergence between a true and approximating pdf. And in fact, the general learning problem of the form (2.7) is the general idea behind Variational Inference (VI): An additional auxiliary distribution, the so-called variational posterior $q(\mathbf{x}) = q(\mathbf{x}|\mathbf{y} = \mathbf{y}_i)$, is introduced and optimized — for each realization $\mathbf{y}_i$ with $i = 1, \ldots, N$ — in order to approximate the true and maybe intractable posterior $p(\mathbf{x}|\mathbf{y})$. By this means, we approximate the posterior computation (2.3) by an optimization problem avoiding the marginalization (2.4). In its basic form, the learning problem in (2.7) now reads

$$q^*(\mathbf{x}) = \arg\min_{q(\mathbf{x})} D_{\mathrm{KL}}\left(p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i) \parallel q(\mathbf{x})\right) \tag{2.11}$$

and holds for a fixed value or observation realization $\mathbf{y} = \mathbf{y}_i$. If this optimization problem is unconstrained, i.e., no constraints are imposed on $q(\mathbf{x})$, the unique and trivial solution is $q^*(\mathbf{x}) = p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i)$.

The key idea of Variational Inference (VI) is to choose a suitable model $q(\mathbf{x}|\boldsymbol{\varphi})$ with advantageous properties, such as being member of the exponential family (see Sec. 2.7) and parametrized by a vector $\boldsymbol{\varphi} \in \mathbb{R}^{N_{\boldsymbol{\varphi}} \times 1}$, allowing to solve (2.11) with limited complexity. Note that the optimization problem is now constrained by the set of distributions $\{q(\mathbf{x}|\boldsymbol{\varphi})\}$ defined by the given variational parametrization $\boldsymbol{\varphi}$:

$$\begin{aligned}
\boldsymbol{\varphi}_i^* &= \arg\min_{\boldsymbol{\varphi}} D_{\mathrm{KL}}\left(p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i) \parallel q(\mathbf{x}|\boldsymbol{\varphi})\right) && (2.12)\\
&= \arg\min_{\boldsymbol{\varphi}} \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y}=\mathbf{y}_i)}[-\ln q(\mathbf{x}|\boldsymbol{\varphi})] && (2.13)\\
&= \arg\min_{\boldsymbol{\varphi}} \mathcal{H}\left(p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i), q(\mathbf{x}|\boldsymbol{\varphi})\right) . && (2.14)
\end{aligned}$$

The solution $q(\mathbf{x}|\boldsymbol{\varphi}_i^*)$ of (2.12) is commonly known as the Moment (M)-projection of the posterior $p(\mathbf{x}|\mathbf{y})$ into the set $\{q(\mathbf{x}|\boldsymbol{\varphi})\}$ [Sim18a]. If the set is large enough to contain distributions close to the true posterior, then approximate equality $q(\mathbf{x}|\boldsymbol{\varphi}_i^*) \approx p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i)$ is guaranteed. The objective function in (2.13) is the cross-entropy and derived by (2.9) and (2.10). Note that optimization in (2.13) requires knowledge of the true posterior. Thus, the optimization problem does not appear to be solvable. Fortunately, this is not true if $q(\mathbf{x}|\boldsymbol{\varphi})$ belongs to the exponential family. Then, the M-projection can be obtained by moment matching to the moments of $p(\mathbf{x}|\mathbf{y})$ [Sim18a]. Most notably, we can exploit amortized inference to exchange the dependence on $p(\mathbf{x}|\mathbf{y})$ by one on $p(\mathbf{x}, \mathbf{y})$ — explained in Sec. 2.4.2 and being extensively used throughout this thesis — to enable optimization with the M-projection in general.

Information-Projection

Recalling that the KL divergence is non-symmetric, we can change the order of $p(\mathbf{x}|\mathbf{y})$ and $q(\mathbf{x})$ in (2.11) to arrive at a different optimization problem [Bis06; Sim18a]:

$$\begin{aligned}
\boldsymbol{\varphi}_i^* &= \arg\min_{\boldsymbol{\varphi}} \; D_{\mathrm{KL}}\left(q(\mathbf{x}|\boldsymbol{\varphi}) \parallel p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i)\right) && (2.15)\\
&= \arg\min_{\boldsymbol{\varphi}} \; \mathrm{E}_{\mathbf{x}\sim q(\mathbf{x}|\boldsymbol{\varphi})}[\ln q(\mathbf{x}|\boldsymbol{\varphi})] - \mathrm{E}_{\mathbf{x}\sim q(\mathbf{x}|\boldsymbol{\varphi})}[\ln p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i) \cdot p(\mathbf{y} = \mathbf{y}_i)] \\
&\quad + \ln p(\mathbf{y} = \mathbf{y}_i) && (2.16)\\
&= \arg\min_{\boldsymbol{\varphi}} \; \mathrm{E}_{\mathbf{x}\sim q(\mathbf{x}|\boldsymbol{\varphi})}[\ln q(\mathbf{x}|\boldsymbol{\varphi})] - \mathrm{E}_{\mathbf{x}\sim q(\mathbf{x}|\boldsymbol{\varphi})}[\ln p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)] && (2.17)\\
&= \arg\min_{\boldsymbol{\varphi}} \; D_{\mathrm{KL}}\left(q(\mathbf{x}|\boldsymbol{\varphi}) \parallel p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)\right) . && (2.18)
\end{aligned}$$

Its solution is known as the Information (I)-projection. The unique solution without constraints on $q(\mathbf{x})$ is again $q^*(\mathbf{x}) = p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i)$. Moreover, problem reformulation (2.18) avoids the need of computing the desired posterior $p(\mathbf{x}|\mathbf{y})$, only requiring the joint generative distribution $p(\mathbf{x}, \mathbf{y})$. The optimization term in (2.17) is the so-called *variational free energy* [Sim18a]. The I-projection tends to underestimate the support of $p(\mathbf{x}|\mathbf{y})$ and places mass on one of its modes [Bis06; Sim18a]. This can be explained by the fact that $q(\mathbf{x}|\boldsymbol{\varphi})$ needs to be 0 whenever $p(\mathbf{x}|\mathbf{y}) = 0$ for the KL divergence to remain finite. The opposite argumentation of an overestimating support applies to the M-projection. For an illustrative comparison of I- and M-projection, we refer the reader to [Bis06; Sim18a].

Lastly, we note that the *Laplace approximation* provides a simpler alternative for approximating the posterior of continuous RVs, fitting a Gaussian

distribution with a mean equal to one of the true distribution's modes — such as the MAP estimate — and a local precision observed around it [Bis06].

Beyond Conventional Divergence Measures

The KL divergence is only one of several divergence measures to quantify the distance between two distributions. It belongs, like the symmetric Jensen-Shannon divergence, to the more general α-divergence [Sim18a]. Changing the value of α, we are able to find projections between that of $D_{\mathrm{KL}}(p \parallel q)$ and $D_{\mathrm{KL}}(q \parallel p)$, i.e., between the I- and M-projection. In fact, the α-divergence itself is part of the larger class of f-divergences $D_f(p \parallel q)$ which include Generative Adversarial Networks (GANs) [Sim18a]. GANs learn the divergence measure through data-based optimization of a discriminator leading to State of the Art (SotA) performance, e.g., in image generation.

2.4.2 Amortized Inference

A major insight of solving problem (2.12) and (2.15) is that the variational posterior $q(\mathbf{x}|\boldsymbol{\varphi})$ must be derived for each observation $\mathbf{y} = \mathbf{y}_i$ independently. This may become computationally inefficient. To overcome this problem, the idea of *amortized inference* introduces an inference variational distribution $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ conditioned on $\mathbf{y}$ and valid for each observation $\mathbf{y} = \mathbf{y}_i$ [Sim18a; BB23]. This can be for example parametrized by a Deep Neural Network (DNN) model capable of approximating arbitrarily well (see Sec. 2.7). Once $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ is learned, only inference but no optimization is required anymore. With conditioning of the M-projection in (2.12) on $\mathbf{y}$, we arrive at the following problem:

$$\boldsymbol{\varphi}^* = \arg\min_{\boldsymbol{\varphi}} \; \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[D_{\mathrm{KL}}(p(\mathbf{x}|\mathbf{y}) \parallel q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi}))] \; . \tag{2.19}$$

Now, the KL minimization is amortized across multiple values of $\mathbf{y}$ — a crucial tool in this thesis, e.g., in Chapter 3 and Chapter 4.

Applying the marginalization across $\mathbf{y}$ to the KL divergence decomposition from (2.9) including conditioning on $\mathbf{y}$, we see that the marginalized entropy term $\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}(p(\mathbf{x}|\mathbf{y}))]$ is still independent of $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$. It follows that amortized minimization of the cross-entropy is equivalent to that of the KL divergence (2.19). Rewriting the amortized cross-entropy objective function

$$\begin{aligned} \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}(p(\mathbf{x}|\mathbf{y}), q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi}))] &= \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}\big[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})]\big] \\ &= \mathrm{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})] \end{aligned} \tag{2.20}$$

by the law of iterated expectations [Sim18a], reveals an important insight: *In* (2.20)*, or—as we will see later—in* (2.60)*, the expected value is calculated*

w.r.t. the joint distribution $p(\mathbf{x}, \mathbf{y})$: Therefore, through amortization, explicit knowledge of the true posterior $p(\mathbf{x}|\mathbf{y})$ which may be of intractable complexity is not required.

Note that in Sec. 2.5.2 we show that the MC approximation of amortized inference applied to the M-projection is equivalent to the Maximum Likelihood (MaxL) problem. Moreover, amortized inference can be defined for the I-projection (2.15) by conditioning on $\mathbf{y}$ as well.

2.4.3 Mean-Field Variational Inference

Another common and useful assumption in VI to make marginalization computationally tractable is that the variational posterior $q(\mathbf{x})$ factorizes into N_x distributions $q_n(x_n)$ according to

$$q(\mathbf{x}) = \prod_{n=1}^{N_x} q_n(x_n)\,, \tag{2.21}$$

such that the RVs x_n are statistically independent.

I-Projection

If we further perform an I-projection (2.15) iteratively for one target RV x_n at a time and assume the factors $q_m(x_m)$ for all other RVs x_m with $m \neq n$ to be fixed, we arrive at the method of Mean-Field Variational Inference (MFVI). Problem (2.15) w.r.t. each factor $q_n(x_n)$ then reads:

$$q_n^*(x) = \underset{q_n(x)}{\arg\min}\; D_{\mathrm{KL}}\left(q(\mathbf{x}) \parallel p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)\right) \tag{2.22}$$

$$= \underset{q_n(x)}{\arg\min}\; - \mathrm{E}_{x_n \sim q_n(x_n)} \left[\underbrace{\mathrm{E}_{\mathbf{x}_{\setminus n} \sim \prod_{\substack{m=1 \\ m \neq n}}^{N_x} q_m(x_m)} \left[\ln p(x_n, \mathbf{x}_{\setminus n}, \mathbf{y} = \mathbf{y}_i)\right]}_{\mathrm{E}_{\mathbf{x}_{\setminus n}}[\ln p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)]} \right]$$

$$- \sum_{m=1}^{N_x} \mathcal{H}\left(q_m(x_m)\right) \tag{2.23}$$

$$= \underset{q_n(x)}{\arg\min}\; D_{\mathrm{KL}}\left(q_n\left(x_n\right) \parallel \exp(\mathrm{E}_{\mathbf{x}_{\setminus n}}[\ln p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)])\right)\,, \tag{2.24}$$

where $\mathbf{x}_{\setminus n}$ is a vector containing all entries of $\mathbf{x}$ except for x_n. After solving the minimization problem of (2.24) [Bis06; Sim18a], we arrive at:

$$\ln q_n^*(x_n) = \mathrm{E}_{\mathbf{x}_{\setminus n}}[\ln p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)] + c \tag{2.25}$$

$$q_n^*(x_n) = \underbrace{\frac{\exp\left(\mathrm{E}_{\mathbf{x}_{\setminus n}}[\ln p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)]\right)}{\sum_{x_n \in \mathcal{M}_x} \exp\left(\mathrm{E}_{\mathbf{x}_{\setminus n}}[\ln p(\mathbf{x}, \mathbf{y} = \mathbf{y}_i)]\right)}}_{=-c} \tag{2.26}$$

with constant c. Finally, this results in an iterative algorithm: We are able to compute an approximate posterior $q(\mathbf{x})$ by cycling through computation of its factors (2.26) iteratively for all $n \in \{1, \ldots, N_x\}$. These *serial updates* correspond to solving the I-projection by coordinate descent, which requires choosing a starting point. Since each serial step ensures that the variational free energy (2.22) monotonically decreases, the convergence of the MFVI iterations to a stationary point of problem (2.24) is guaranteed [Bis06; Sim18a]. This is not true for parallel updates. Convergence must be checked via the remaining change in free energy. Most notably, the form of each factor $q_n(x)$ is not restricted — a crucial benefit. Typically, the true joint pdf $p(\mathbf{x}, \mathbf{y})$ is composed of distributions of the so-called exponential family, e.g., Gaussian and Bernoulli distributions, making the marginalization in (2.26) tractable. Note that the RVs are only interdependent through the update equations (2.26), while the statistical dependencies present in the true posterior are ignored in the mean-field approximation.

M-Projection

In Chapter 3, we show the application of MFVI to the M-projection [BBD21]. We derive by (3.4) and (3.31) that the KL divergence decomposes into a sum of individual cross-entropies between the Individual Optimal (IO) true posteriors $p(x_n|\mathbf{y})$ and the variational factors $q_n(x_n)$ and the constant entropy term $\mathcal{H}\left(p(\mathbf{x}|\mathbf{y})\right)$ [Bis06; BBD21]:

$$D_{\mathrm{KL}}\left(p(\mathbf{x}|\mathbf{y}) \parallel q(\mathbf{x})\right) = \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[\ln p(\mathbf{x}|\mathbf{y})] + \sum_{n=1}^{N_x} \mathcal{H}\left(p(x_n|\mathbf{y}), q_n(x_n)\right) . \tag{2.27}$$

Thus, the optimal solution without restrictions on $q(\mathbf{x})$ is the minimum of the individual cross-entropies $\mathcal{H}\left(p(x_n|\mathbf{y}), q_n(x_n)\right)$ [Bis06]:

$$q_n^*(x_n) = p(x_n|\mathbf{y} = \mathbf{y}_i) = \sum_{\mathbf{x}_{\setminus n} \in {\mathcal{M}_x}^{N_x - 1}} p(\mathbf{x}|\mathbf{y} = \mathbf{y}_i) . \tag{2.28}$$

This means that for the M-projection with MFVI, a closed-form solution exists, i.e., the IO posterior, and no iterative procedure, as it is the case for the I-projection, is required. However, as already outlined, the calculation of the posterior and its marginalization in (2.28) may be intractable.

Fano's Inequality: In detection problems, the MAP estimate of (2.28) is known as the IO detector. It minimizes the symbol/bit error probability or Symbol Error Rate (SER)/Bit Error Rate (BER) in communications. The IO posterior can be related via the conditional entropy $\mathcal{H}\left(x_n|\mathbf{y}\right) = \mathcal{H}\left(p(x_n, \mathbf{y}), p(x_n|\mathbf{y})\right)$ to the error probability $p(x_n \neq \hat{x}_n)$ of inferring $x_n \in \mathcal{M}$ by the detector $\hat{x}_n = f(\mathbf{y}) \in \mathcal{M}$ given the observation $\mathbf{y}$ with the posterior $p(x_n|\mathbf{y})$, e.g., by Fano's inequality [CT06]:

$$\mathcal{H}\left(p(x_n \neq \hat{x}_n)\right) + p(x_n \neq \hat{x}_n) \cdot \ln\left(|\mathcal{M}| - 1\right) \geq \mathcal{H}\left(x_n|\hat{x}_n\right) \geq \mathcal{H}\left(x_n|\mathbf{y}\right) . \tag{2.29}$$

For example, channel effects such as high noise and interference translate into high (aleatoric) uncertainty in the posterior and thus a high conditional entropy $\mathcal{H}\left(x_n|\mathbf{y}\right)$ — eventually increasing the minimum possible error probability $p(x_n \neq \hat{x}_n)$.

Using an amortized M-projection with approximation $q_n(x_n|\mathbf{y}, \boldsymbol{\varphi})$, we approach the conditional entropy $\mathcal{H}\left(x_n|\mathbf{y}\right)$ as a lower bound minimizing the cross-entropy in (2.27) amortized across $\mathbf{y}$. This lower bound $\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}\left(p(x_n|\mathbf{y}), q_n(x_n|\mathbf{y}, \boldsymbol{\varphi})\right)] \geq \mathcal{H}\left(x_n|\mathbf{y}\right)$ can be derived combining (2.6) and (2.9) and introducing conditioning on $\mathbf{y}$. The amortized cross-entropy $\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}\left(p(x_n|\mathbf{y}), q_n(x_n|\mathbf{y}, \boldsymbol{\varphi})\right)]$ — usually numerically calculated based on samples as the training or validation loss in ML toolboxes — can be used to compute a conservative estimate of the minimum error probability via (2.29).

2.4.4 Variational Inference: MIMO Detection Example

To show how MFVI can be used in communications, we apply MFVI to the example of detection in a Multiple Input Multiple Output (MIMO) system covered more deeply in Chapter 3 and Appendix A. We assume the prior $p(\mathbf{b}|\boldsymbol{\alpha})$ and the variational posterior $q(\mathbf{b}|\boldsymbol{\varphi})$ of bit sequences $\mathbf{b}$ to be Bernoulli pmfs and a Gaussian pdf for the generative and real-valued MIMO system model $p(\mathbf{y}|\mathbf{x}, \mathbf{H}, \sigma_{\mathrm{n}}^2)$ with transmit symbols $\mathbf{x} \in \mathcal{M}^{N_{\mathrm{T}} \times 1}$ from Binary Phase Shift Keying (BPSK) alphabet $\mathcal{M} = \{-1, 1\}$, received signal $\mathbf{y} \in \mathbb{R}^{N_{\mathrm{R}} \times 1}$, channel matrix $\mathbf{H} \in \mathbb{R}^{N_{\mathrm{R}} \times N_{\mathrm{T}}}$ with channel statistics $\mathbf{H} \sim p(\mathbf{H})$, and noise

variance $\sigma_{\mathrm{n}}^2 \in \mathbb{R}^+$ (see Chapter 3):

$$p(\mathbf{b}|\boldsymbol{\alpha}) = \prod_{n=1}^{N_{\mathrm{bit}}} p(b_n|\alpha_n) = \prod_{n=1}^{N_{\mathrm{bit}}} \alpha_n^{(1-b_n)} \cdot (1-\alpha_n)^{b_n} \tag{2.30a}$$

$$q(\mathbf{b}|\boldsymbol{\varphi}) = \prod_{n=1}^{N_{\mathrm{bit}}} q(b_n|\varphi_n) = \prod_{n=1}^{N_{\mathrm{bit}}} \varphi_n^{(1-b_n)} \cdot (1-\varphi_n)^{b_n} \tag{2.30b}$$

$$\mathbf{x} = 1 - 2 \cdot \mathbf{b} \tag{2.30c}$$

$$p(\mathbf{y}|\mathbf{x}, \mathbf{H}, \sigma_{\mathrm{n}}^2) = \mathcal{N}\left(\mathbf{H} \cdot \mathbf{x}, \sigma_{\mathrm{n}}^2 \cdot \mathbf{I}\right) . \tag{2.30d}$$

Factorization of the prior pmf $p(\mathbf{b}|\boldsymbol{\alpha})$ means we assume statistical independence between the incoming bits in $\mathbf{b}$. The prior probability of each bit being $b_n = 0$ is $p(b_n = 0|\alpha_n) = \alpha_n \in [0, 1]$ and the corresponding posterior probability to be learned is $q(b_n = 0|\varphi_n) = \varphi_n \in [0, 1]$. Note that $N_{\mathrm{T}} = N_{\mathrm{bit}}$ for BPSK and that all distributions belong to the exponential family.

I-Projection: We derive parallel update equations [LL09] for the I-projection with (2.26) and $\mathbf{y} = \mathbf{y}_i$ as:

$$\boldsymbol{\varphi}^{(j+1)} = \rho\left(\ln\left(\frac{\boldsymbol{\alpha}}{1-\boldsymbol{\alpha}}\right) + \frac{2}{\sigma_{\mathrm{n}}^2}\left[\mathbf{H}^T\mathbf{y} - \mathbf{D} \cdot \left(2 \cdot \boldsymbol{\varphi}^{(j)} - \mathbf{1}\right)\right]\right) \tag{2.31a}$$

$$\mathbf{D} = \mathbf{H}^T\mathbf{H} - \mathrm{diag}\left\{\mathbf{H}^T\mathbf{H}\right\} \tag{2.31b}$$

with the sigmoid activation function ρ (see Appendix C). To derive the serial updates in (2.26), we optimize one row of equation (2.31) at a time to compute $\varphi_n^{(j+1)}$, performing the optimization in a sequential, row-wise manner. The posterior probabilities $\boldsymbol{\varphi}^{(N_{\mathrm{it}})}$ of $q(\mathbf{b}|\boldsymbol{\varphi}^{(N_{\mathrm{it}})}) = q(\mathbf{b}|\mathbf{y}, \mathbf{H}, \sigma_{\mathrm{n}}^2, \boldsymbol{\alpha}, \boldsymbol{\varphi}^{(0)})$ from (2.31) after N_{it} iterations can be passed as soft information, e.g., transformed into Log-Likelihood Ratios (LLRs) $= \ln(\boldsymbol{\varphi}^{(N_{\mathrm{it}})}/(1-\boldsymbol{\varphi}^{(N_{\mathrm{it}})}))$, to a soft decoder closely interfacing equalization and decoding by iterative joint equalization and decoding. This resembles a factorization of both actually statistical dependent steps being too computational complex when solved together in one problem. As an extension, we could include the noise variance σ_{n}^2 as one of the parameters to be estimated leading to a variational Expectation Maximization (EM) algorithm [LL09].

Numerical Results: The results when MFVI is applied for detection in a 128×64 massive MIMO system with independent and identically distributed (i.i.d.) Gaussian channel matrix $p(\mathbf{H}) = \mathcal{N}(0, 1/N_{\mathrm{R}} \cdot \mathbf{I})$, $\boldsymbol{\varphi}^{(0)} = \boldsymbol{\alpha} = 0.5 \cdot \mathbf{1}$, and $N_{\mathrm{it}} = 64$ iterations (see Chapter 3), are shown in Fig. 2.2, and compared

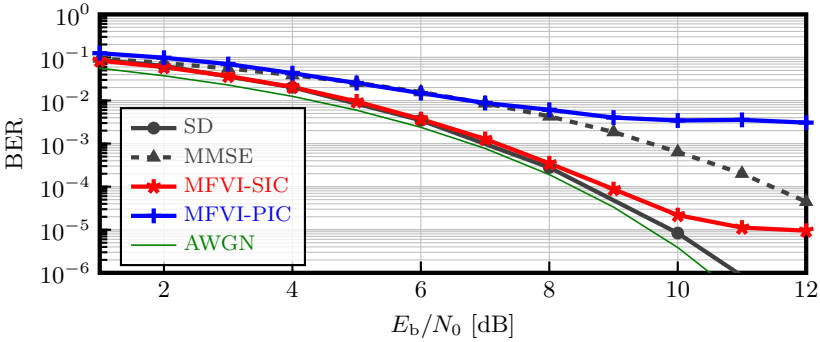

Figure 2.2: BER curves of MFVI-based detection methods in a 64×32 massive MIMO system with Quadrature Phase Shift Keying (QPSK) modulation. Effective dimension of the equivalent real-valued system with BPSK modulation is 128×64 and for iterative algorithms $N_{\text{it}} = 64$.

to MAP sequence detection using the Sphere Detector (SD), as well as the Minimum Mean Square Error (MMSE). We see that MFVI-Successive Interference Cancellation (SIC) with serial updates from (2.26) performs close to the SD for low Signal-to-Noise Ratio (SNR) since it is guaranteed to converge to a stationary point [Bis06; Sim18a]. In contrast, MFVI-Parallel Interference Cancellation (PIC) applying parallel updates in (2.31) cannot beat the MMSE equalizer. In a symmetric 64×64 MIMO system, we can observe a similar behavior, but now both approaches cannot compete with the MMSE equalizer hinting towards the suboptimality of the MFVI solution (2.26) for MIMO detection. At first, statistical independence, i.e., full factorization, in the posterior $q(\mathbf{b}|\boldsymbol{\varphi})$ or equivalently $q(\mathbf{x}|\boldsymbol{\varphi})$ w.r.t. BPSK symbols x_n with a full-entry channel matrix $\mathbf{H}$ seems to be a crude assumption. However, there are two major flaws specific to the I-projection:

1. As outlined in Sec. 2.4.1, the I-projection tends to underestimate the support and place all its mass on one mode of $p(\mathbf{x}|\mathbf{y})$. This becomes a drawback if there exist multiple symbol vectors with a similar probability, as even more likely symbol vectors may not be covered appropriately.

2. In fact, there is no guarantee that the serial updates converge to the global minimum of (2.22), and that the optimum coincides with the individual true posteriors, i.e., $q_n(x_n) = p(x_n|\mathbf{y})$.

M-Projection: Applying the MFVI assumption to the M-projection for MIMO detection, we can overcome these drawbacks: Via (2.27), we can show that the optimal solution $q_n^*(x_n)$ in (2.28) is the soft output, i.e., the symbol-wise posterior $p(x_n|\mathbf{y})$, of the IO detector (3.4), and that we can learn an approximation $q_n(x_n|\boldsymbol{\varphi}_n)$ of it. The IO detector minimizes the individual SER which directly translates into BER for BPSK symbols. Moreover, the M-projection tends to overestimate the support and thus covers better the uncertainty, i.e., the most likely symbol vectors, or the multi-modality caused by the interference.

In Chapter 3, we exploit these benefits to design a low-complex soft detector that achieves low BER [BBD21]: We define an inference distribution $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ for all $\mathbf{y} = \mathbf{y}_i$ for amortized inference, avoiding knowledge of the intractable individual posteriors $p(x_n|\mathbf{y})$ and constraining the solution space by incorporating model knowledge from the pdf $p(\mathbf{x}, \mathbf{y})$ into $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$. The detector obtained by data-driven optimization (see Sec. 2.5) performs close to the SD, e.g., in a symmetric 64×64 MIMO system, in contrast to the solution (2.31) of the I-projection.

Beyond Mean-Field: At this point, we note that as an alternative to MFVI we can assume the same factorization for $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ as the joint distribution $p(\mathbf{x}, \mathbf{y})$. This is the so-called Bethe approximation and can be solved by means of loopy belief propagation [Sim18a]. One simplification of the latter is Approximate Message Passing (AMP) which leads to competitive performance in MIMO detection as will be shown in Chapter 3.

2.4.5 Monte Carlo Methods

One crucial approximate inference technique besides VI techniques is Monte Carlo (MC) sampling [Bis06; Sim18a]. To make the explanation of MC methods more precise, let us apply them to our basic marginalization problem (2.4). To do so, we rewrite into the new form

$$p(\mathbf{y}) = \sum_{\mathbf{x} \in \mathcal{M}_x^{N_x}} p(\mathbf{y}|\mathbf{x}) p(\mathbf{x}) = \mathrm{E}_{\mathbf{x} \sim p(\mathbf{x})}[p(\mathbf{y}|\mathbf{x})] \,. \tag{2.32}$$

By drawing N i.i.d. samples $\mathbf{x}_i \sim p(\mathbf{x})$, $i = 1, \ldots, N$, we are able to approximate the expected value (2.32) by its consistent estimator, i.e., the empirical average

$$p(\mathbf{y}) \approx \frac{1}{N} \sum_{i=1}^{N} p(\mathbf{y}|\mathbf{x} = \mathbf{x}_i) \,. \tag{2.33}$$

We note that MC sampling is the origin of a phenomenon known as *overfitting*. We will uncover this in Sec. 2.5.5.

2.5 Data-driven Supervised Learning for Receiver Inference

Given the approximate inference techniques of MC methods and amortized inference, we can now formulate the supervised learning problem of approximating the true posterior distribution $p(\mathbf{x}|\mathbf{y})$ from a data-driven perspective. Doing so reveals its information-theoretic relation to the Maximum Likelihood (MaxL) and MAP principle and that no knowledge of the true posterior is required.

2.5.1 Monte Carlo Variational Inference

To learn a discriminative model in a supervised manner, we usually assume an inference variational distribution $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ according to amortized VI in Sec. 2.4.2. The optimization criterion to obtain the M-projection is then amortized minimization of the KL divergence (2.19) or equivalently of the cross-entropy in (2.20). *An important implication of using the objective function* (2.20) *is that explicit knowledge of the true posterior* $p(\mathbf{x}|\mathbf{y})$ *which may be of intractable complexity is not required through amortized inference.*

To learn $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ by minimizing (2.20), we can exploit a MC approximation of (2.20) from Sec. 2.4.5 and restrict to N pair of i.i.d. samples $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ from the joint distribution $p(\mathbf{x}, \mathbf{y})$ collected in a set $\mathcal{D}$. This combination of MC and VI techniques is referred to as *Monte Carlo variational inference*. Now, the empirical cross-entropy is the new optimization objective:

$$\mathrm{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})] \approx -\frac{1}{N}\sum_{i=1}^{N} \ln q(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\varphi})\,. \tag{2.34}$$

This learning criterion (2.34) based on samples resembles more closely what humans understand if they think of learning, since they are usually not aware of the underlying process $p(\mathbf{x}, \mathbf{y})$. Additionally, this criterion defines the concept of data-driven supervised learning: Based on pairwise observations $(\mathbf{x}_i, \mathbf{y}_i)$, the relation between both variables is learned.

2.5.2 Relation to Maximum Likelihood

Likewise, the approximate equality (2.34) shows the fundamental relation between cross-entropy, KL divergence and the Maximum Likelihood (MaxL) criterion. To explain this, let us take a step back and recall the MAP criterion for model selection: It selects the most probable model $q(\boldsymbol{\varphi}|\mathbf{x},\mathbf{y})$ or model parameters $\boldsymbol{\varphi}$ given realizations, i.e., a training set, of $\mathbf{x}$ and $\mathbf{y}$. With Bayes' theorem

$$q(\boldsymbol{\varphi}|\mathbf{x},\mathbf{y}) = \frac{q(\mathbf{x},\boldsymbol{\varphi}|\mathbf{y})}{q(\mathbf{x})} \sim q(\mathbf{x}|\mathbf{y},\boldsymbol{\varphi}) \cdot q(\boldsymbol{\varphi})\,, \tag{2.35}$$

we can relate to the variational distribution, i.e., likelihood $q(\mathbf{x}|\mathbf{y},\boldsymbol{\varphi})$, and the prior pdf $q(\boldsymbol{\varphi})$ on the models $\{q\}$ or model parameters $\boldsymbol{\varphi}$. Assuming a *non-informative prior* $q(\boldsymbol{\varphi}) = c$ in (2.35), it holds $q(\boldsymbol{\varphi}|\mathbf{x},\mathbf{y}) \sim q(\mathbf{x}|\mathbf{y},\boldsymbol{\varphi})$.

A so-called training set typical for supervised learning consists of the collection $\mathcal{D}$ of i.i.d. samples or data we mentioned so far. With this set $\mathcal{D}$, the MaxL problem w.r.t. the parameters $\boldsymbol{\varphi}$ now reads:

$$\boldsymbol{\varphi}^* = \arg\max_{\boldsymbol{\varphi}}\; q(\mathbf{x}_1,\ldots,\mathbf{x}_N|\mathbf{y}_1,\ldots,\mathbf{y}_N,\boldsymbol{\varphi}) \tag{2.36}$$

$$= \arg\max_{\boldsymbol{\varphi}}\; \prod_{i=1}^{N} q(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\varphi}) \tag{2.37}$$

$$= \arg\min_{\boldsymbol{\varphi}}\; -\frac{1}{N}\sum_{i=1}^{N} \ln q(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\varphi})\,. \tag{2.38}$$

After having reformulated the problem w.r.t. the negative log-likelihood, we observe that the MaxL objective function in (2.38) is an empirical approximation of the information-theoretic measure of cross-entropy in (2.34). In other words, (2.20) and (2.38) are asymptotically equivalent or approximately the same for large N:

$$\lim_{N\to\infty} -\frac{1}{N}\sum_{i=1}^{N} \ln q(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\varphi}) = \mathrm{E}_{(\mathbf{x},\mathbf{y})\sim p(\mathbf{x},\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y},\boldsymbol{\varphi})]\;. \tag{2.39}$$

Therefore, for large N, the MaxL problem can be interpreted to minimize the amortized KL divergence or cross-entropy between the true pdf $p(\mathbf{x}|\mathbf{y})$ and the approximating pdf $q(\mathbf{x}|\mathbf{y},\boldsymbol{\varphi})$.

2.5.3 MAP Criterion

To move beyond the MaxL principle towards the Maximum A Posteriori (MAP) criterion, we introduce a non-uniform prior pdf $q(\boldsymbol{\varphi})$ to (2.35). This

means we do not only regard the model's RVs, but also all parameters $\boldsymbol{\varphi}$ explicitly as RVs. Now, we can define the MAP criterion w.r.t. the parameters $\boldsymbol{\varphi}$:

$$\boldsymbol{\varphi}^* = \arg\max_{\boldsymbol{\varphi}} \ \ln q(\mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\varphi} | \mathbf{y}_1, \ldots, \mathbf{y}_N) \tag{2.40}$$

$$= \arg\min_{\boldsymbol{\varphi}} \ -\ln q(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \ldots, \mathbf{y}_N, \boldsymbol{\varphi}) - \ln q(\boldsymbol{\varphi}) \tag{2.41}$$

$$= \arg\min_{\boldsymbol{\varphi}} \ -\ln q(\boldsymbol{\varphi}) - \sum_{i=1}^{N} \ln q(\mathbf{x}_i | \mathbf{y}_i, \boldsymbol{\varphi}) \,. \tag{2.42}$$

The prior pdf $q(\boldsymbol{\varphi})$ acts like a regularization term. For example, if we choose $q(\boldsymbol{\varphi})$ to be a Gaussian or Laplace distribution, it gives lower probability to large values of $\boldsymbol{\varphi}$, which can help avoid overfitting (see Sec. 2.5.5) [Bis06; Sim18a]. Furthermore, with a Gaussian or Laplace distribution, we arrive at the l_2- and l_1-regularization, respectively. If the prior pdf $q(\boldsymbol{\varphi})$ is uniform/constant, MaxL and MAP criterion coincide.

2.5.4 Fully Bayesian Inference

We note that the MaxL and MAP criterion only select the most probable parameters $\boldsymbol{\varphi}$ for the given training set $\mathcal{D}$. Under the *strong assumption* that both observed data, data to be inferred and parameters $\boldsymbol{\varphi}$ follow the same distribution $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$, i.e., the true joint distribution $p(\mathbf{x}, \mathbf{y}, \boldsymbol{\varphi})$ is known, we can exploit fully Bayesian inference by marginalizing the model parameters $\boldsymbol{\varphi}$:

$$p(\mathbf{x}|\mathbf{y}, \mathcal{D}) = \int \frac{p(\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\varphi} | \mathbf{y}, \mathbf{y}_1, \ldots, \mathbf{y}_N)}{p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \ldots, \mathbf{y}_N)} \, \mathrm{d}\boldsymbol{\varphi} \tag{2.43a}$$

$$= \int p(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi}) \cdot \frac{p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \ldots, \mathbf{y}_N, \boldsymbol{\varphi}) \cdot p(\boldsymbol{\varphi})}{p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \ldots, \mathbf{y}_N)} \, \mathrm{d}\boldsymbol{\varphi} \tag{2.43b}$$

$$= \int p(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi}) \cdot p(\boldsymbol{\varphi} | \mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{y}_1, \ldots, \mathbf{y}_N) \, \mathrm{d}\boldsymbol{\varphi} \tag{2.43c}$$

with

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \ldots, \mathbf{y}_N) = \int p(\mathbf{x}_1, \ldots, \mathbf{x}_N | \mathbf{y}_1, \ldots, \mathbf{y}_N, \boldsymbol{\varphi}) \cdot p(\boldsymbol{\varphi}) \, \mathrm{d}\boldsymbol{\varphi} \tag{2.43d}$$

$$= \int p(\boldsymbol{\varphi}) \cdot \prod_{i=1}^{N} p(\mathbf{x}_i | \mathbf{y}_i, \boldsymbol{\varphi}) \, \mathrm{d}\boldsymbol{\varphi} \,. \tag{2.43e}$$

With this fully Bayesian perspective, *inference and learning are one process*. Thus, we can account for uncertainties of the parameter estimation during

inference and quantify epistemic uncertainty [Bis06; Sim18a]. A drawback is that marginalization of the model parameters $\boldsymbol{\varphi}$ adds computational cost and can become intractable as for example with DNNs. To find a tractable approximation, VI techniques can be exploited [Bis06].

Considering a generative model $p(\mathbf{x}, \mathbf{y}, \boldsymbol{\varphi})$ in communications, a fully Bayesian approach means that uncertainties in the system variables $\boldsymbol{\varphi}$, such as channel and noise variance, can also be taken into account and can subsequently be estimated. For example, the respective prior distributions of channel and noise as well as the uncertainty in their estimation from pilot data can be included for improved soft detection.

Prior Distributions: For a likelihood $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ from the exponential family, *conjugate priors* $p(\boldsymbol{\varphi})$ can be selected such that the posterior $p(\mathbf{x}, \boldsymbol{\varphi}|\mathbf{y})$ belongs to the same class of distributions. Then, computation of the posterior is analytically tractable [Sim18a]. However, this does not imply analytical tractability in computing the posterior predictive pdf $p(\mathbf{x}|\mathbf{y}, \mathcal{D})$ in (2.43).

Furthermore, we note that in a fully Bayesian approach the shape of the parameter prior $p(\boldsymbol{\varphi}|\boldsymbol{\psi})$ is defined via *hyperparameters* $\boldsymbol{\psi}$ which themselves have to be chosen carefully. If we further introduce a prior $p(\boldsymbol{\psi})$ on the hyperparameters, we enter the realm of *hierarchical models* [Sim18a]. As a hybrid approach, the *empirical Bayes* method estimates the hyperparameters of the prior from the data [Sim22].

2.5.5 Monte Carlo Methods and Overfitting

We made use of MC sampling when empirically approximating the amortized cross-entropy in (2.34) or (2.39): We replaced expected values by empirical averages over samples of the joint distribution $p(\mathbf{x}, \mathbf{y})$ to define a supervised learning problem in the form (2.34) that alleviates intractable marginalization.

At this point, we note that the equivalence of this approximation is only assured for sufficiently large N due to the law of large numbers. Additionally, the variance of the approximation error scales with $1/N$. In order for the MC approximation to work, the number of samples N therefore has to be sufficiently large. Otherwise, if N is too small, we fit $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ to a few samples or points which results in a well-known phenomenon in supervised learning, known as *overfitting*. If overfitting occurs, *generalization* to unseen data points becomes poor.

In particular, large-capacity models $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ such as DNNs from a larger hypothesis class, i.e., the set of all possible models $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$, are prone to overfitting, as these models can better fit to the training data. With such

models, the *estimation error* (or *variance*) in generalization dominates the *bias*, which is caused by the choice of a low-capacity model. This explains why low-capacity models $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ oftentimes generalize better when the training set is small. In contrast, when much data is available, low-capacity models tend to *underfit*. For illustration of overfitting and underfitting, we refer the reader to [Bis06; Sim18a].

Bias-Variance Trade-off: This relationship can be formulated mathematically as the *bias-variance trade-off* [Bis06; Sim18a] from statistical learning [Sim18a] and is discussed in [Bis06; Sim18a] w.r.t. the MSE loss. While a MSE loss permits a straightforward decomposition into bias and variance, recent research has shown that clean decompositions can be achieved with g-Bregman divergences [Hes25]. Since the KL divergence (2.5) is a g-Bregman divergence, it has a bias-variance decomposition [Hes25]. Its decomposition is also valid for the cross-entropy (2.10) or (2.20), that is widely used in this thesis for learning, and can be rewritten into a constant entropy and KL divergence term via (2.9).

2.5.6 Training, Validation, Test Datasets

To detect overfitting when training on samples, i.e., a finite dataset, based on (2.34), the dataset is typically split into *training set* $\mathcal{D}$, *validation set* $\mathcal{D}_{\text{Val}}$, and *test set* $\mathcal{D}_{\text{Test}}$. First, we select and train a model $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ and observe the validation loss to evaluate its generalization performance. The cycle of model selection and training continues until the validation loss no longer decreases. We explain this for the example of [BBD21] in Appendix A.3.2 in more detail.

In fact, after training and validation of a model $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$, we need to evaluate the generalization capability once again with a test set. This is because we selected the model to minimize the validation error itself computed with a finite dataset $\mathcal{D}_{\text{Val}}$.

Since a well-defined model is typically available in digital wireless communications, it is possible to generate virtually unlimited training data, thereby mitigating overfitting and enabling high generalization performance without the use of validation or test datasets. We elaborate on this point in Appendix A.

As a final remark, we note that the fully Bayesian approach can partly mitigate overfitting by accounting for high epistemic uncertainty in parameter estimation when data is limited. However, validation remains necessary w.r.t. the hyperparameters $\boldsymbol{\psi}$.

2.6 Information Maximization Principle for Unsupervised Learning of Communications Design

Moving beyond learning how to infer or predict in a supervised manner, e.g., how it would be done on the receiver side, let us extend the previous model:

- We now assume a typical communications Markov chain $\mathbf{s} \to \mathbf{x} \to \mathbf{y}$ with input signal $\mathbf{s} \in \mathcal{M}_s^{N_s \times 1}$, i.e., the source, transmit signal $\mathbf{x} \in \mathcal{M}_x^{N_x \times 1}$, and received signal $\mathbf{y} \in \mathcal{M}_y^{N_y \times 1}$: We describe this chain by a probabilistic encoder — a pdf $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) = p(\mathbf{y}|\mathbf{s}, \boldsymbol{\theta})$, that comprises both transmitter encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ and communication channel $p(\mathbf{y}|\mathbf{x})$, and is parametrized by parameters $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}} \times 1}$.
- We note that the role of the RVs changes compared to supervised learning at the receiver side. When considering the complete transceiver, now the observation is not the received signal but the input signal $\mathbf{y} \rightsquigarrow \mathbf{s}$. The target RV changes from the transmit signal to the received signal, i.e., $\mathbf{x} \rightsquigarrow \mathbf{y}$. Since it is not labeled, it further becomes a latent representation, and we have an *unsupervised learning* problem. Moreover, the probabilistic encoder $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ is a discriminative model, as it maps the observation $\mathbf{s}$ directly to representation $\mathbf{y}$.
- In communications, we assume the input prior pdf $p(\mathbf{s})$ or samples from it to be known. Therefore, we model the joint pdf $p_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{s}) = p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) \cdot p(\mathbf{s})$ between observation RV $\mathbf{s}$ and its representations $\mathbf{y}$.

Classic InfoMax: Our typical aim for communications is to optimize or learn the encoder to maximize throughput, i.e., to maximize the Shannon Mutual Information (MI) $I(\mathbf{s}; \mathbf{y})$ w.r.t. the probabilistic encoder $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ or its parameters $\boldsymbol{\theta}$. In other words, we want to find a received representation $\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ in an *unsupervised* manner that retains a significant amount of information about the input $\mathbf{s}$:

$$p^*_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) = \arg\max_{p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})} I(\mathbf{s}; \mathbf{y}) \tag{2.44}$$

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} I_{\boldsymbol{\theta}}(\mathbf{s}; \mathbf{y}) \tag{2.45}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{s},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})} \left[\ln \frac{p_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{y})}{p(\mathbf{s}) p_{\boldsymbol{\theta}}(\mathbf{y})} \right] \tag{2.46}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathcal{H}(p(\mathbf{s})) - \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) \tag{2.47}$$

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \; \mathcal{H}\left(p(\mathbf{s})\right) - \mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})\right) \tag{2.48}$$

$$= \arg\max_{\boldsymbol{\theta}} \; \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})] \; . \tag{2.49}$$

In the ML domain, this problem is known as the *InfoMax principle* [Sim18a]. We note that it does not only fit well as a design criterion for *unsupervised learning* of discriminative models, but naturally also as a general design criterion for the design of communication systems. This brings us to a crucial viewpoint of this thesis: **Learning can be defined in communication systems as the optimization process aiming to maximize the information contained in the received signal about the signal of interest to be transmitted**.

Note independence from $\boldsymbol{\theta}$ in $\mathcal{H}\left(p(\mathbf{s})\right)$ and dependence in $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) \cdot p(\mathbf{s})$. Notably, the form of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ has to be constrained implicitly to avoid learning a trivial identity mapping $\mathbf{s} = \mathbf{y}$. Since the communication channel $p(\mathbf{y}|\mathbf{x})$ is included in $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ and introduces noise, this is usually true for communications. Additionally, note that $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})$ is the posterior and thus the optimal decoder given the encoder forward model $p_{\boldsymbol{\theta}}(\mathbf{y},\mathbf{s})$.

For an important variation of the InfoMax principle that introduces an information constraint on the forward model explicitly, i.e., the Information Bottleneck (IB) problem, we refer the reader to Chapter 4 and [TPB99; GP20; ZES20; Has22].

InfoMax for Transmitter Encoder: So far, in the classic InfoMax problem (2.44), we optimized for the probabilistic encoder $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$, encompassing the whole forward model. To directly obtain the transmit encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ for the full communications Markov chain $\mathbf{s} \to \mathbf{x} \to \mathbf{y}$, we exploit the affine transformations

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) = \sum_{\mathbf{x}\in\mathcal{M}_x^{N_x}} p(\mathbf{y}|\mathbf{x}) \cdot p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) \tag{2.50a}$$

$$p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) = \int_{\mathbf{x}\in\mathcal{M}_x^{N_x}} p(\mathbf{y}|\mathbf{x}) \cdot p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) \, \mathrm{d}\mathbf{x} \tag{2.50b}$$

for discrete and continuous sets $\mathcal{M}_x^{N_x}$, respectively. By means of this affine transformation, we can rewrite (2.44) into the equivalent optimization problem w.r.t. the transmitter encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ that we aim to optimize:

$$p^*_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) = \arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} \; I\left(\mathbf{s};\mathbf{y}\right) \; . \tag{2.51}$$

All other optimization problems from (2.45) to (2.49) are also valid in this case, since the parametrization of $I_{\boldsymbol{\theta}}\left(\mathbf{s};\mathbf{y}\right)$ w.r.t. $\boldsymbol{\theta}$ is unchanged.

Convexity Analysis: The InfoMax problem (2.44) is convex with regard to the encoder functional $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ for fixed $p(\mathbf{s})$ [CT06]. This holds for both discrete and continuous RVs $\mathbf{s}$ and $\mathbf{y}$, as the expectation operator in terms of sums and integrals is an affine mapping that is composed with a convex function which preserves convexity [BV04, Sec. 3.2.2.]. However, the MI $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ is not necessarily convex with regard to the encoder parameters $\boldsymbol{\theta}$. For example, it is non-convex if the encoder function is non-convex with regard to its parameters being typically the case with DNN encoders.

Since the affine transformations in (2.50) preserve convexity, the InfoMax problem w.r.t. the encoder (2.51) is still convex in $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ for fixed $p(\mathbf{s})$. Once the global maximum $p^*_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ is found the corresponding maximum $p^*_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ of (2.44) can be calculated by (2.50).

Upper Bound on the InfoMax Problem: It remains the question of how large the MI $I(\mathbf{s};\mathbf{y})$ can be at maximum. Through the information processing inequality [CT06], we know:

$$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq \min\left\{I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}), I_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y})\right\}. \tag{2.52}$$

In case of negligible encoder compression $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}) > I_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y})$, the capacity is the upper bound on the achievable information rate:

$$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq I_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y}) \leq \max_{p(\mathbf{x});\ \mathrm{E}\left[|x_n|^2\right]\leq 1} I_{\boldsymbol{\theta}}(\mathbf{x};\mathbf{y}) = C \tag{2.53}$$

for $n = \{1,\ldots,N_x\}$. For a more detailed analysis on the bounds in the context of semantic communication, we refer to Sec. 4.5.5.

Moreover, the MI in (2.44) can be related to the error probability $p(\mathbf{s} \neq \hat{\mathbf{s}})$ of inferring $\mathbf{s}$ given the representation $\mathbf{y}$ by Fano's inequality (2.29) using $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) = \mathcal{H}(p(\mathbf{s})) - I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ [Sim18a].

2.6.1 Mutual Information Lower Bound

If calculation of the posterior $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})$ in (2.49) is intractable, we are able to replace it by a variational distribution $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) = q(\mathbf{s}|\mathbf{y},\boldsymbol{\varphi})$. This approach yields a lower bound on the MI, referred to as the MI Lower BOund (MILBO):

$$\begin{aligned} I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) &= \mathcal{H}(p(\mathbf{s})) - \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) && (2.54)\\ &= \mathcal{H}(p(\mathbf{s})) + \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})] && (2.55)\\ &\geq \mathcal{H}(p(\mathbf{s})) + \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})], && (2.56)\end{aligned}$$

where the last step follows from non-negativity (2.6) of the KL divergence $\mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})/q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})] \geq 0$. The MILBO problem (2.56) can be

solved using a Majorization Minimization (MM) approach, a class of algorithms to which the EM algorithm from ML belongs as a specific instance [Sim18a]. Alternatively, optimization w.r.t. both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ can now be done directly w.r.t. this lower bound:

$$\arg\max_{\boldsymbol{\theta},\boldsymbol{\varphi}} \mathcal{H}\left(p(\mathbf{s})\right) + \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})] \tag{2.57}$$

$$=\arg\max_{\boldsymbol{\theta},\boldsymbol{\varphi}} \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})] \tag{2.58}$$

$$=\arg\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} -\mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})]\ . \tag{2.59}$$

This is known as the *variational InfoMax* problem [Sim22]. Reformulation from (2.56) via (2.59) to

$$\begin{aligned} -\mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})] =& \mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y}), q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})\right) && \text{(2.60a)}\\ =& \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[\mathrm{E}_{\mathbf{s}\sim p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})]\right] && \text{(2.60b)}\\ =& \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}), q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})\right)] && \text{(2.60c)}\\ =& \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} && \text{(2.60d)} \end{aligned}$$

reveals that maximization of the MILBO is equivalent to minimization of the cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ with amortization across $\mathbf{y}$ from (2.20). The only difference between (2.20) and (2.60) is that now also encoder optimization parameters $\boldsymbol{\theta}$ are included. *We conclude that approaches that rely on the minimization of the amortized cross-entropy* $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ *in* (2.60) *approximately maximize the MI.* One example where this optimization criterion is usually implemented is the popular AutoEncoder (AE) approach that consists of an encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ and a decoder $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$, both typically parametrized by a DNN and optimized as one entity [OH17; BBD23].

Further rewriting the amortized cross-entropy — as shown in [SAH19; CAD+20] or will be shown in Sec. 4.5.3 for the case of semantic communication [BBD23] — reveals that it includes a new decoder optimization term in addition to the MI:

$$\begin{aligned} \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} &= \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[\mathrm{E}_{\mathbf{s}\sim p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})]\right]\\ &= \mathcal{H}\left(p(\mathbf{s})\right) - \underbrace{I_{\boldsymbol{\theta}}\left(\mathbf{s};\mathbf{y}\right)}_{\text{encoder objective}} + \underbrace{\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) \parallel q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})\right)]}_{\text{decoder objective}}\ . \end{aligned} \tag{2.61}$$

This means, optimization of the MILBO balances maximization of the MI $I_{\boldsymbol{\theta}}\left(\mathbf{s};\mathbf{y}\right)$ w.r.t. $\boldsymbol{\theta}$ and minimization of the KL divergence $D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) \parallel q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})\right)$ w.r.t. both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$. The latter criterion can be seen as a regularization term that favors encoders with high MI for which decoders can be learned that are close to the true posterior.

Examples: Lastly, for illustration, we show what (2.60) looks like for the example of a Gaussian variational posterior $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) = \mathcal{N}(\mu_{\boldsymbol{\varphi}}(\mathbf{y}), \sigma^2 \cdot \mathbf{I})$ with mean parametrized by a function $\mu_{\boldsymbol{\varphi}}(\mathbf{y})$, e.g., a DNN, conditioned on observation $\mathbf{y}$ and with parameters $\boldsymbol{\varphi}$:

$$\mathcal{L}^{\text{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} = \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}\left[\frac{1}{2\sigma^2}(\mathbf{s}-\mu_{\boldsymbol{\varphi}}(\mathbf{y}))^2 + \frac{1}{2}\ln 2\pi\sigma^2\right]. \tag{2.62}$$

This result shows that, if the noise variance σ^2 is not parametrized by $\boldsymbol{\varphi}$, then minimizing the amortized cross-entropy between true posterior $p(\mathbf{s}|\mathbf{y})$ and approximating Gaussian variational posterior $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ is equal to minimization of the MSE loss between true signal $\mathbf{s}$ and its prediction $\mu_{\boldsymbol{\varphi}}(\mathbf{y})$. With a discrete categorical pmf

$$q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) = \prod_{k=1}^{M} q_{\boldsymbol{\varphi}}(\mathbf{s}=\mathbf{m}_k|\mathbf{y})^{[\mathbf{s}=\mathbf{m}_k]} \quad \text{with} \quad \sum_{k=1}^{M} q_{\boldsymbol{\varphi}}(\mathbf{s}=\mathbf{m}_k|\mathbf{y}) = 1 \tag{2.63}$$

as described in (3.32) in Sec. 3.4.2 with M possible values $\mathbf{m}_k$ and $[\mathbf{s}=\mathbf{m}_k]$ being the Iverson bracket, we recover the cross-entropy loss oftentimes used in classification problems in ML:

$$\mathcal{L}^{\text{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} = \mathrm{E}_{\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}\left[-\sum_{k=1}^{M}[\mathbf{s}=\mathbf{m}_k]\cdot\ln q_{\boldsymbol{\varphi}}(\mathbf{s}=\mathbf{m}_k|\mathbf{y})\right] \tag{2.64}$$

$$\approx -\frac{1}{N}\sum_{i=1}^{N}\ln q_{\boldsymbol{\varphi}}(\mathbf{s}=\mathbf{s}_i|\mathbf{y}=\mathbf{y}_i)\,. \tag{2.65}$$

Monte Carlo Optimization: We note that computation of the MILBO leads to some problems: If calculating the expected value in (2.60) cannot be solved analytically or is computationally intractable, we can use MC sampling techniques as for example in (2.65). For Stochastic Gradient Descent (SGD)-based optimization (see Sec. 2.7.3), the gradient w.r.t. $\boldsymbol{\varphi}$ can then be computed easily. Calculation of the gradient w.r.t. $\boldsymbol{\theta}$ turns out to be more problematic since we sample w.r.t. the pdf $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ dependent on $\boldsymbol{\theta}$. Solution techniques to overcome this problem include the reparametrization trick that leads to the AE approach, and Stochastic Policy Gradient (SPG). Both are described and exploited in Chapter 4 and Chapter 5, specifically Sec. 4.5.6/Sec. 5.4.1 and Sec. 5.4.2, respectively.

2.6.2 Relation between M-Projection in Supervised Learning and InfoMax Principle

As an important remark, we arrive at a special case of the InfoMax principle if we fix the encoder with $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) = p(\mathbf{y}|\mathbf{s})$ and hence the transmitter. Then, only the receiver approximate posterior $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ needs to be optimized in (2.59). Comparing cross-entropies (2.20) and (2.60) for this case, maximization of the MILBO is equivalent to a *supervised learning problem* and amortized minimization of the KL divergence (2.19) between true and approximate posterior, i.e., the M-projection (see Sec. 2.3.3, Sec. 2.4.2, and Chapter 3).

This means the *M-projection is well-justified from a theoretical perspective for communications* since it maximizes a lower bound on the mutual information between transmitted data and received signal. We recall from Sec. 2.4.3 that we learn approximations of the IO posteriors for detection that minimize the SER using the M-projection with a MFVI assumption, as outlined for MIMO detection in Sec. 2.4.4.

Fixing the transmitter can have several benefits: In practice, we avoid the *Reinforce gradient* (see Chapter 5), and especially we do not need any (ideal) connection between transmitter and receiver for optimization like the raw AE approach. Furthermore, even today in 5G, we can apply a ML receiver design to standardized systems with fixed transmitter capabilities to possibly achieve performance gains. We will investigate a ML-based receiver design given a fixed transmitter in Chapter 3, Appendix B.4, and [BBD21], [BSW+23], respectively. Finally, we note that the SotA transmitters aim for digital bit-perfect transmission, which may be unnecessary from a semantic perspective and could lead to a waste of resources. Hence, it is also worth considering the adaptation of the transmitter to achieve a more efficient use of bandwidth. We will elaborate on this point in Chapter 4.

2.6.3 Comparison to Generative Models and ELBO

To conclude this section, we now compare *discriminative* and *directed generative modeling* in unsupervised learning, since the respective approximative optimization criteria of maximizing MILBO and Evidence Lower BOund (ELBO) are important concepts in ML. Note that *"Learning a "useful" representation of data in an unsupervised way is one of the "holy grails" of current ML research"* [APF+18]. First, we transfer typical communications modeling assumptions to *generative* modeling to demonstrate how it can be used in communications:

- Using a directed generative model $p_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{y}) = p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y}) \cdot p_{\boldsymbol{\theta}}(\mathbf{y})$

parametrized by $\boldsymbol{\theta}$ means that we assume that a latent representation $\mathbf{y}$ generates the observation $\mathbf{s}$ through forward model $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})$.

- Since in communications this observation $\mathbf{s}$ is our input signal, applying generative modeling with the assumption on the role of the variables from the InfoMax scenario is not useful.

- If we *switch the roles of observation* $\mathbf{s} \rightsquigarrow \mathbf{y}$ *and latent representation* $\mathbf{y} \rightsquigarrow \mathbf{s}$, we arrive at a scenario where we aim to learn which mechanism, e.g., including transmitter and channel, generated our observations $\mathbf{y}$ at the receiver. Then, the generative model $p_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{y}) = p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) \cdot p_{\boldsymbol{\theta}}(\mathbf{s})$ consists of the forward model $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ — including channel and transmitter — and the prior on the input signal $p_{\boldsymbol{\theta}}(\mathbf{s})$.

- With $\mathbf{x} = \mathbf{s}$, we split between channel $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ and prior on the transmit symbols $p_{\boldsymbol{\theta}}(\mathbf{x})$. This example resembles *blind channel equalization*.

To learn generative models, typically the KL divergence between the true data/observation $p(\mathbf{y})$ and its approximative distribution $p_{\boldsymbol{\theta}}(\mathbf{y})$ is minimized [Bis06; Sim18a]:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} D_{\mathrm{KL}}\left(p(\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{y})\right) \tag{2.66}$$

$$= \arg\min_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[-\ln p_{\boldsymbol{\theta}}(\mathbf{y})] - \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[-\ln p(\mathbf{y})] \tag{2.67}$$

$$= \arg\min_{\boldsymbol{\theta}} \mathcal{H}\left(p(\mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{y})\right) . \tag{2.68}$$

The problem (2.66) is also known as the M-projection of the data distribution $p(\mathbf{y})$ into the model $p_{\boldsymbol{\theta}}(\mathbf{y})$. Since the KL divergence in (2.66) converges to infinity when $p_{\boldsymbol{\theta}}(\mathbf{y})$ approaches zero, the M-projection tends to overestimate the support and can provide blurry estimates. Alternatively to the M-projection $D_{\mathrm{KL}}\left(p(\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{y})\right)$ also used in data-driven supervised learning, we can use other f-divergences $D_f\left(p(\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{y})\right)$ such as the reverse KL divergence $D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{y}) \parallel p(\mathbf{y})\right)$ (I-projection) or GANs to alleviate this problem [Sim18a].

In fact, compared to the InfoMax principle (2.44) used for discriminative model learning, (2.66) and other f-divergences do not include any measure of the quality of the latent representation variable $\mathbf{s}$ and thus do not necessarily lead to good representations and models $p_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{y})$, as shown in [APF+18].

To optimize (2.66), marginalization of the generative model $p_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{y})$ is required to compute the evidence $p_{\boldsymbol{\theta}}(\mathbf{y})$ which may be computationally

intractable. To avoid marginalization, we can introduce a variational approximation $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ in

$$\ln p_{\boldsymbol{\theta}}(\mathbf{y}) = \ln \sum_{\mathbf{s}\in\mathcal{M}_s^{N_s\times 1}} p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y}) \tag{2.69}$$

$$= \ln \sum_{\mathbf{s}\in\mathcal{M}_s^{N_s\times 1}} q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \cdot \frac{p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}{q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})} \tag{2.70}$$

$$\geq \sum_{\mathbf{s}\in\mathcal{M}_s^{N_s\times 1}} q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \cdot \ln\left(\frac{p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})}{q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})}\right) \tag{2.71}$$

$$= -D_{\mathrm{KL}}\left(q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})\right) \tag{2.72}$$

$$= -\mathrm{E}_{\mathbf{s}\sim q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})}[-\ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})] - D_{\mathrm{KL}}\left(q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{s})\right) \tag{2.73}$$

$$= \ln p_{\boldsymbol{\theta}}(\mathbf{y}) - D_{\mathrm{KL}}\left(q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})\right) . \tag{2.74}$$

to arrive at the Evidence Lower BOund (ELBO) in (2.72) via Jensen's inequality in (2.71). The surrogate objective function of the amortized minimization problem in (2.66) now reads:

$$D_{\mathrm{KL}}\left(p(\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{y})\right) \leq \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[D_{\mathrm{KL}}\left(q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})\right)] = -\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\varphi}}^{\mathrm{ELBO}} . \tag{2.75}$$

One typical approach to maximize $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\varphi}}^{\mathrm{ELBO}}$ is the Expectation Maximization (EM) algorithm where first the variational posterior $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ and then the model $p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{y})$ is optimized [Bis06; Sim18a]. As for the MILBO, we can also optimize w.r.t. both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ and use MC approximations for SGD-based optimization. This leads to the same problem already observed for the MILBO that we need to compute the gradient of the cross-entropy term in (2.73) w.r.t. parameters $\boldsymbol{\varphi}$ that parametrize the pdf $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ we sample from.

Variational AutoEncoder (VAE): We can solve this problem in (2.73) by means of the reparametrization trick. This approach is known as the Variational AutoEncoder (VAE) consisting of encoder $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ and decoder $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ [Sim18a; BB23]. The VAE is applicable if the *variational regularization* term $D_{\mathrm{KL}}\left(q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y}) \parallel p_{\boldsymbol{\theta}}(\mathbf{s})\right)$ in (2.73) can be computed and differentiated w.r.t. $\boldsymbol{\varphi}$. This reparametrization requirement restricts the form of the pdfs $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ and $p_{\boldsymbol{\theta}}(\mathbf{s})$ even if expressive DNNs, e.g., deep DNNs that have a high capacity to model complex functions and capture intricate patterns in data, are used for parametrization. For example, with Gaussian pdf assumptions on $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ and $p_{\boldsymbol{\theta}}(\mathbf{s})$ [Sim18a], the performance of this unsupervised learning approach can be affected.

Example – Blind Channel Equalization: In the example of a communication system, unsupervised learning of a generative model could mean that we learn transmitter, channel and receiver by $p_{\boldsymbol{\theta}}(\mathbf{s})$, $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$, and $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$, respectively. If we assume the transmitter technology, e.g., modulation, to be known without knowledge of the exact transmit symbols, this is known as blind channel equalization. This task was tackled in [CB20] by means of the VAE. Since the authors assume a typical wireless communication channel, i.e., linear Gaussian channel model and a Bernoulli prior, both being from the exponential family, analytical computations of the expected values in the ELBO are possible. Otherwise, for example, application of the Gumbel-softmax trick (see Chapter 3) becomes necessary due to the discrete nature of the latent RVs $\mathbf{s}$. The authors demonstrate significant and consistent improvements in the SER compared to SotA approaches while being computational efficient.

Relation between I-Projection in Supervised Learning and ELBO: If we assume the generative model pdf in (2.72) to be known $p_{\boldsymbol{\theta}}(\mathbf{y}, \mathbf{s}) = p(\mathbf{y}, \mathbf{s})$, we arrive at a supervised learning problem. Then, maximization of the ELBO (2.72) and the I-projection (2.18) from Sec. 2.4.1 coincide. Thus, the *I-projection has its theoretical justification in minimizing an upper bound on the KL divergence between true observation pdf and its approximation.* This means we learn representations $\mathbf{s}$ that explain the observed data well, but that do not necessarily contain much information about the observation $\mathbf{y}$. This can lead to bad detection performance: We recall that the performance of the I-projection using a MFVI assumption in the MIMO detection example from Sec. 2.4.4 is inferior to that of the M-projection. The purpose of the I-projection is to make inference computationally tractable and efficient.

2.7 Probabilistic Models for Learning

In this section, we elaborate on the most important learning models used in this thesis.

2.7.1 Exponential Family Models

As already mentioned in Sec. 2.4.1, variational posteriors or more generally pdfs $q_{\boldsymbol{\varphi}}(\mathbf{x}) = q(\mathbf{x}|\boldsymbol{\varphi})$ from the exponential family exhibit many desirable mathematical properties [Sim18a]. These pdfs are log-linear and characterized by a set of sufficient statistics — functions of the data that contain all information relevant for determination of the pdfs' natural parameters $\boldsymbol{\varphi} = \boldsymbol{\varphi}_{\text{nat}}$ [Bis06; Sim18a]. Their structure gives rise to three key properties:

1. The feasible natural parameters $\boldsymbol{\varphi}_{\text{nat}}$ are from a convex set.

2. The mapping between natural parameters $\boldsymbol{\varphi}_{\text{nat}}$ and mean parameters — which are defined as the expectation of the vector of sufficient statistics and can be used as an alternative parametrization — is invertible.

3. The log-likelihood $\ln q(\mathbf{x}|\boldsymbol{\varphi}_{\text{nat}})$ is a concave function of the natural parameters $\boldsymbol{\varphi}_{\text{nat}}$. Thus, minimizing the cross-entropy for a member of the exponential family corresponds to a convex optimization problem.

These properties make the exponential family pdfs well-suited for tractable computation in, e.g., message passing or gradient descent-based optimization. For example, all models in the exponential family admit conjugate priors, which ease posterior computation in Bayesian inference (see Sec. 2.5.4). The exponential family includes Gaussian, Laplace, Gamma, Beta and Dirichlet pdfs, as well as Bernoulli, Categorical, Multinomial and Poisson pmfs. Furthermore, exponential family models have a strong theoretical justification being solutions of the maximum-entropy problem: These models retain the maximum uncertainty about the RVs given known constraints on the RVs' moments, i.e., they make the fewest assumptions about the data which improves generalization (see Sec. 2.5.5). For a deeper and mathematical description of the exponential family, see [Bis06; Sim18a].

One popular extension of the exponential family to conditional pdfs $q_{\boldsymbol{\varphi}}(\mathbf{x}|\mathbf{y})$ or $q_{\boldsymbol{\varphi}}(\mathbf{s}|\mathbf{y})$ is the Generalized Linear Model (GLM). Here, the natural parameters $\boldsymbol{\varphi}_{\text{nat}}$ are a linear function $\boldsymbol{\varphi}_{\text{nat}} = \mathbf{W} \cdot \phi(\mathbf{y})$ of a vector of features $\phi(\mathbf{y})$ of the inputs $\mathbf{y}$ with weights $\mathbf{W}$ such that $q(\mathbf{x}|\boldsymbol{\varphi}_{\text{nat}} = \mathbf{W}\phi(\mathbf{y})) = q(\mathbf{x}|\mathbf{y}, \mathbf{W}) = q_{\boldsymbol{\varphi}}(\mathbf{x}|\mathbf{y})$ with $\boldsymbol{\varphi} = \{\mathbf{W}\}$. Moving beyond GLMs with, e.g., DNNs, enables to also learn the input features $\phi(\mathbf{y})$ themselves [Sim18a].

2.7.2 Artificial Neural Networks

At the core of almost all recent ML breakthroughs such as Chat Generative Pre-Trained Transformer (ChatGPT) [VSP+17] and AlphaGo [SHM+16] lie artificial Neural Networks (NNs) inspired by the working principle of the human brain. One of the reasons is that, compared to GLMs fixed basis functions $\phi(\mathbf{y})$, NNs enable arbitrarily accurate approximations of non-linear functions according to the universal approximation theorem [HSW89; Bis06; OH17]. The training effort is higher, as the basis functions, i.e., features $\phi(\mathbf{y})$, of the model are not predefined, but can be learned.

Model: The functionality of an artificial neuron is based on a linear combination of all inputs, which are mapped to one output via a non-linear

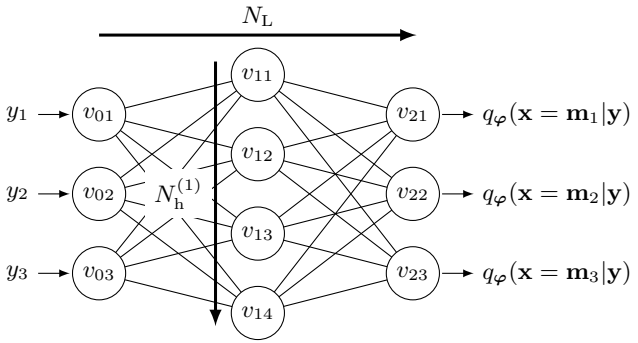

Figure 2.3: DNN with $N_\mathrm{L}=2$ two layers of width $N_\mathrm{h}^{(1)}=4$ and $N_\mathrm{h}^{(2)}=3$. In this example, the final softmax layer computes the class probabilities of the variational posterior pmf $q_{\boldsymbol{\varphi}}(\mathbf{x}|\mathbf{y})$ based on observation $\mathbf{y}$.

activation function $\rho(\cdot)$ such as linear, Rectified Linear Unit (ReLU), sigmoid, softmax, and tanh. For a short summary of these activation functions, see Appendix C. The performance of the entire NN is determined by the number of neurons in parallel (layer width) and in series (depth) and the choice of activation functions. The typical multi-layer structure with depth N_L and (hidden) layer width $N_\mathrm{h}^{(l)}$ of the l-th layer is shown in Fig. 2.3. A NN with high N_L is considered as a Deep Neural Network (DNN) and able to learn abstract features present in the data [Sim18a].

In total, the DNN consists of multiple dense, fully-connected layers with parameters, i.e., additive biases $\mathbf{b}_l$ and weights $\mathbf{W}_l$:

$$\mathbf{v}_0 = \mathbf{y} \tag{2.76a}$$

$$\mathbf{v}_l = \rho_l\left(\mathbf{W}_l \cdot \mathbf{v}_{l-1} + \mathbf{b}_l\right) \quad \text{with} \quad l = 1, \ldots, N_\mathrm{L}-1 \tag{2.76b}$$

$$\mathbf{v}_{N_\mathrm{L}} = \rho_{N_\mathrm{L}}\left(\mathbf{W}_{N_\mathrm{L}} \cdot \mathbf{v}_{N_\mathrm{L}-1} + \mathbf{b}_{N_\mathrm{L}}\right). \tag{2.76c}$$

The parameters $\boldsymbol{\varphi}$ have the following dimensions:

$$\boldsymbol{\varphi} = \Big\{\mathbf{W}_1 \in \mathbb{R}^{N_\mathrm{h}^{(1)} \times N_y}, \mathbf{W}_2 \in \mathbb{R}^{N_\mathrm{h}^{(2)} \times N_\mathrm{h}^{(1)}}, \ldots, \mathbf{W}_{N_\mathrm{L}} \in \mathbb{R}^{N_\mathrm{h}^{(N_\mathrm{L})} \times N_\mathrm{h}^{(N_\mathrm{L}-1)}},$$
$$\mathbf{b}_1 \in \mathbb{R}^{N_\mathrm{h}^{(1)} \times 1}, \mathbf{b}_2 \in \mathbb{R}^{N_\mathrm{h}^{(2)} \times 1}, \ldots, \mathbf{b}_{N_\mathrm{L}} \in \mathbb{R}^{N_\mathrm{h}^{(N_\mathrm{L})} \times 1}\Big\}. \tag{2.77}$$

A promising aspect for the application of DNNs in communication systems is that the *complexity* of the resulting algorithms can be analyzed more easily due to the regular structure (2.76) of the DNNs. In addition, the requirement of computational resources and the *latency* can be easily determined or even specified or taken into account in the optimization.

Reparametrizing Variational Distributions by DNNs: Note that if the final layer of a DNN is a softmax activation $\rho_{N_\mathrm{L}}(\cdot) = \sigma(\cdot)$, it outputs values in the interval $[0, 1]$ summing up to 1: This means the DNN represents a valid conditional categorical pmf $q(\mathbf{x}|\boldsymbol{\varphi}_\mathrm{nat}) \triangleq \sigma(\boldsymbol{\varphi}_\mathrm{nat}) = \mathbf{v}_{N_\mathrm{L}} \in [0, 1]^{M\times 1}$ with $M = \left|\mathcal{M}_x^{N_x}\right|$ classes, whose natural parameters are a linear function $\boldsymbol{\varphi}_\mathrm{nat} = \mathbf{W}_{N_\mathrm{L}} \cdot \mathbf{v}_{N_\mathrm{L}-1} + \mathbf{b}_{N_\mathrm{L}}$ of preceding layer outputs and input $\mathbf{y}$. Each k-th entry in $\mathbf{v}_{N_\mathrm{L}}$ gives the respective probability $q_{\boldsymbol{\varphi}}(\mathbf{x} = \mathbf{m}_k|\mathbf{y})$ of class $\mathbf{m}_k$ for $\mathcal{M}_x^{N_x} = \{\mathbf{m}_k\}_{k=1}^{M}$. We can use this DNN with final softmax layer as a discriminative model or variational posterior $q_{\boldsymbol{\varphi}}(\mathbf{x}|\mathbf{y})$ with parameters $\boldsymbol{\varphi}$ from (2.77) for classification. Other members of the exponential family can also be parametrized by DNNs. For example, the mean parameters of a Gaussian $q_{\boldsymbol{\varphi}}(\mathbf{x}|\mathbf{y})$ can be computed by a DNN, i.e., the mean and variance by $\{\mu, \sigma^2\} = \mathbf{v}_{N_\mathrm{L}}$, as typical in VAEs.

DNNs and Overfitting: Building on the discussion in Sec. 2.5.5, DNNs particularly are prone to overfitting since DNNs are large-capacity models with millions or billions of parameters and many layers that can approximate arbitrarily well.

However, when the model class capacity exceeds the so-called *interpolation point*, where the DNN is effectively able to memorize the entire training set, the generalization error decreases again. This phenomenon — subject of current research — is called *double descent* and one of the reasons for the success of Large Language Models (LLMs) [BB23]. The current notion is that in a larger model space there may be more local parameter optima that generalize well such that a suboptimal training algorithm is able to find one of those. As a result, the general wisdom in the deep learning community is that large-capacity DNNs combined with suboptimal training algorithms generalize better, especially with limited data. Furthermore, surprisingly, performance can decrease if the dataset size increases, as a larger model is required to reach the interpolation point.

Since low-complexity, low-power, and low-latency components are crucial for efficient communication system design, we expect low-capacity DNNs to be of greater interest. In this regime — which is the focus of this thesis — the conventional notions of the bias-variance trade-off become particularly important to mitigate overfitting. Moreover, if a well-defined communications model is available, virtually unlimited training data can be generated, enabling high generalization performance.

Incorporating Domain Knowledge and Structural Priors: Regularization of the form of the DNN model $q_{\boldsymbol{\varphi}}(\mathbf{x}|\mathbf{y})$, e.g., by incorporating expert knowledge into the DNN architecture, or its parameters $\boldsymbol{\varphi}$, e.g., using the

MAP criterion from Sec. 2.5.3, constrains the hypothesis space of solutions. Therefore, it can help mitigate overfitting in case of small datasets and improve performance more generally. Additionally, it can reduce the number of parameters required to achieve a desired performance level.

Pushing the concept to its limits, the idea of *deep unfolding* leverages known iterative model-based algorithms such as AMP or Gradient Descent to create a DNN model: Trainable weights are added to each iteration of the algorithm and the number of iterations is fixed [BS19; MLE21]. We present unfolding of our own algorithm Concrete MAP Detection Network (CMDNet) in Chapter 3.

On an abstract level, domain knowledge can also be introduced to generic architectures: For example, convolutional layers are used in pattern recognition because these are robust to translation, rotation, scale and luminance variance of objects in the image [KSH12]. A DNN incorporating these layers is known as a convolutional DNN and has reduced complexity as convolution operators reduce the number of connections between the layers. This results in filters that are modeled on the visual cortex in animals.

2.7.3 DNN Optimization: Stochastic Gradient Descent

Typically, with the given DNN model and weights, the central optimization/learning criterion of this chapter, i.e., the minimization of cross-entropy (2.10) and (2.59), cannot be evaluated and solved analytically w.r.t. the weights. Therefore, approximate inference techniques including VI, MC samples, and amortized inference are exploited. By these means, we can derive common defined loss functions and the use of datasets describing the usual viewpoint on DNN training.

Since DNNs are non-linear functions due to the inherent non-linear activation functions $\rho(\cdot)$, methods based on gradient descent are used for the optimization of, e.g., the empirical cross-entropy, i.e,

$$\boldsymbol{\varphi}^* \approx \arg\min_{\boldsymbol{\varphi}} \; -\frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \ln q_{\boldsymbol{\varphi}}(\mathbf{x}_i|\mathbf{y}_i)\,. \tag{2.78}$$

Calculating the solution of (2.78) becomes computationally intractable for large datasets $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_{\text{train}}}$.

Stochastic Gradient Descent: To overcome this problem, the most basic ML optimizer Stochastic Gradient Descent (SGD) is usually used for DNN training. The idea of SGD is to split the dataset into a subset, i.e., a mini-batch $\mathcal{D}_{\text{Batch}}$, of samples from the training set and to evaluate the gradient descent step for each mini-batch. The procedure is as follows:

1. Initialize parameters $\boldsymbol{\varphi}^{(0)}$ and set a learning rate schedule $\epsilon^{(j)}$.

2. Split training set $\mathcal{D}$ into mini-batches $\mathcal{D}_{\text{Batch}}$ according to a predetermined or random order.

3. Parameters update with gradient descent for each mini-batch:

$$\boldsymbol{\varphi}^{(j+1)} = \boldsymbol{\varphi}^{(j)} - \frac{\epsilon^{(j)}}{|\mathcal{D}_{\text{Batch}}|} \sum_{\mathcal{D}_{\text{Batch}}} \frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{x}_i|\mathbf{y}_i)}{\partial \boldsymbol{\varphi}} \bigg|_{\boldsymbol{\varphi}=\boldsymbol{\varphi}^{(j)}} . \tag{2.79}$$

In addition to the DNN model, the optimization parameters batch size $|\mathcal{D}_{\text{Batch}}|$ and learning rate $\epsilon^{(j)}$ can be considered as hyperparameters to be optimized via validation [Sim18a]. If the learning rate schedule fulfills the Robbins-Monro conditions, e.g., $\epsilon^{(j)} = 1/j$, then SGD is known to converge to the global optimum for strictly convex functions and to stationary points for non-convex functions typical for DNNs [Sim18a].

In practice, choosing a large batch size $|\mathcal{D}_{\text{Batch}}|$ decreases the variance of the gradient estimate. However, a small batch size improves convergence speed if the current solution is far from the optimum and is known to improve generalization performance avoiding "sharp" minima of the objective function (2.78) [Sim18a]. An increasing batch size schedule suggests itself.

Furthermore, the gradients for DNN training are calculated by the *back-propagation algorithm*. It consists of the following steps:

1. The forward pass calculates $\ln q_{\boldsymbol{\varphi}}(\mathbf{x}_i|\mathbf{y}_i)$ and all intermediate layer responses for all samples.

2. The backward pass requires reiterated use of the chain rule of differentiation for all the weights in each layer for each training sample.

We refer the reader to [Bis06; Sim18a] for a more detailed description and other variations of SGD such as Adaptive Moment Estimation (Adam).

In practice, frameworks such as TensorFlow [AAB+15], PyTorch [PGM+19] and Keras [Cho+15] are used to create computation graphs and perform automatic differentiation, including optimized execution on Graphics Processing Units (GPUs). Exploiting the parallel structure of DNN optimization by parallel execution of training steps on GPUs enables fast training and is one of the main drivers for recent advancements in ML.

DNN Parameter Initialization: When optimization problems are solved iteratively, as in the case of SGD, an initial weight starting point $\boldsymbol{\varphi}^{(0)}$ is required, which can heavily influence the training performance and can be seen as another hyperparameter instance. For $l = 1, \ldots, N_{\text{L}}$, typical default

weights $\mathbf{W}_l \in \mathbb{R}^{N_\mathrm{h}^{(l+1)} \times N_\mathrm{h}^{(l)}}$ from (2.76) are sampled according to the Glorot Uniform Initialization from [GB10]

$$w_{nm}^{(l)} \sim \mathcal{U}\left(-\frac{\sqrt{6}}{\sqrt{N_\mathrm{h}^{(l+1)} + N_\mathrm{h}^{(l)}}}, \frac{\sqrt{6}}{\sqrt{N_\mathrm{h}^{(l+1)} + N_\mathrm{h}^{(l)}}} \right) \tag{2.80}$$

and default biases are $\mathbf{b}_l = \mathbf{0}$. This initialization is known to speed up training convergence.

To regularize the weights during training, we can introduce a non-uniform prior distribution in (2.35) w.r.t. the weights to define a MAP optimization problem (2.42) and optimize $q(\mathbf{x}, \boldsymbol{\varphi}|\mathbf{y})$, as outlined in Sec. 2.5.3. Then, for example, by assuming a Laplace distribution as the prior $q(\boldsymbol{\varphi})$, we can encourage solutions for $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\varphi})$ with sparse weights [Sim18a].

2.8 Chapter Summary

In this chapter, we summarized the fundamentals of ML crucial for its application to communications in this thesis:

- We introduced the most important measures and taxonomy.
- Our information-theoretic view closely connects ML concepts and communications design. We introduce ML techniques very communications-oriented from a unique view with a unique notation.
- Most notably, we reflect on the approaches used in this thesis in the more general context of ML theory — illuminating their background and interconnections within the web of ML concepts — thereby motivating both their choice and possible alternatives.
- We provide a learning definition for both receiver inference and transceiver design from information theory instead of common ML practices. Most notably, we propose the InfoMax principle for communications design.
- We derive the MaxL principle — a common optimization criterion in communications — as a special case of the KL divergence minimization between true and approximating pdf and the InfoMax principle.
- Since the optimization problems are difficult to solve in practice, we introduce useful approximate inference techniques to overcome these difficulties such as amortized VI and MC methods.

- We derive the MFVI solution for a MIMO system and provide numerical results revealing the suboptimality of the I-projection in VI. In contrast, the optima of the M-projection are the IO detectors and hence well-justified from a theoretical perspective.
- We show that overfitting is a result of a MC approximation with too few samples deteriorating generalization performance. Furthermore, we conclude that the recent research finding of double descent — challenging classical intuition by showing that large-capacity DNN models can generalize well despite overparametrization — is not relevant for low-complexity wireless communications design.
- We explain one of the most powerful probabilistic models, i.e., DNNs, requiring the use of SGD variants for solving the non-convex optimization problem.

Part I

Machine Learning for Digital Communications

Chapter 3

Publication 1 – CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection with Low Complexity

E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection With Low Complexity," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8214–8227, Dec. 2021. DOI: 10.1109/TCOMM.2021.3114682

It is an extended version of the paper [BBD20] that has been excluded to maintain focus. The simulation source code is available in [Bec23]. Further analyses and additional details for this publication are provided in Appendix A.

3.1 Abstract

Following the great success of Machine Learning (ML), especially Deep Neural Networks (DNNs), in many research domains in the 2010s, several ML-based approaches were proposed for detection in large inverse linear problems, e.g., massive MIMO systems. The main motivation behind is that the complexity of Maximum A Posteriori (MAP) detection grows exponentially with system dimensions. Instead of using DNNs, essentially being a black-box, we take a slightly different approach and introduce a probabilistic continuous relaxation of discrete variables to MAP detection. Enabling close approximation and continuous optimization, we derive an iterative detection algorithm: Concrete MAP Detection (CMD). Furthermore, extending CMD by the idea of deep unfolding into CMDNet, we allow for (online) optimization of a small number of parameters to different working points while limiting complexity. In contrast to recent DNN-based approaches, we select the optimization criterion and output of CMDNet based on information theory and are thus able to learn approximate probabilities of the individual optimal detector. This is crucial for soft decoding in today's communication systems. Numerical simulation results in MIMO systems reveal CMDNet to feature a promising accuracy complexity trade-off compared to state of the art. Notably, we demonstrate CMDNet's soft outputs to be reliable for decoders.

Index Terms

Maximum A Posteriori (MAP), individual optimal, massive MIMO, concrete distribution, Gumbel-softmax, machine learning, neural networks.

3.2 Introduction

Communications is a long-standing engineering discipline whose theoretical foundation was laid by Claude Shannon with his landmark paper "A Mathematical Theory of Communication" in 1948 [1]. Since then, the theory has evolved into an own field known as information theory today and found its way into many other research areas where data or information is processed including artificial intelligence and especially its subdomain Machine Learning (ML). Information theory relies heavily on description with probabilistic models playing a significant role for design of new generations of cellular communication systems from 2–6G with respective increases in data rate. Probabilistic models have shown to be advantageous also in the ML research

domain. Accordingly, both fields, communications and ML, have touched repeatedly in the past (see, e.g, [2]–[4]).

In the early 2010s, a special class of these models gave rise to several breakthroughs in data-driven ML research: Deep Neural Networks (DNNs). Inspired by the brain, several layers of artificial neurons are stacked on top of each other to create an expressive feed forward DNN able to approximate arbitrarily well [5] and thus to learn higher levels of abstraction, i.e., features, present in data [6]. This is of crucial importance for tasks where there are no well-established models but data to be collected. Previously considered intractable to optimize, dedicated hardware and software, i.e., Graphics Processing Units (GPUs) and automatic differentiation frameworks [7], innovation to DNN models [8], [9] and advancements in training [8] have made it possible to build algorithms that equal or even surpass human performance in specific tasks such as pattern recognition [10] and playing games [11]. The impact included all ML subdomains, e.g., classification [9], [10] in supervised learning, generative modeling in unsupervised learning [12] and Q-learning in reinforcement learning [11].

3.2.1 ML in Communications

The great success of DNNs in many domains has stimulated large amount of work in communications just in recent years [6]. Especially in problems with a model deficit, e.g., detection in molecular and fiber-optical channels [13], [14], or without any known analytical solution, e.g., finding codes for AWGN channels with feedback [15], DNNs have already proven to allow for promising application. Notably, the authors of the early work [16] demonstrate a complete communication system design by interpreting transmitter, channel and receiver as an autoencoder which is trained end-to-end similar to one DNN. The resulting encodings are shown to reach the Bit Error Rate (BER) performance of handcrafted systems in a simple AWGN scenario. A model-free approach based on reinforcement learning is proposed in [17]. Using advances in unsupervised learning, also blind channel equalization can be improved [18].

In contrast to typical ML research areas, a model deficit does not apply to wireless communications. The models, e.g., AWGN, describe reality well and enable development of optimized algorithms. However, those algorithms may be too complex to be implemented. This algorithm deficit applies to the core problem typical for communications: classification in large inverse problems. Therefore, it is crucial to find an approximate solution with an excellent trade-off between detection accuracy and complexity.

3.2.2 Related Work

A prominent example for large inverse problems under current deep investigation and a key enabler for better spectral efficiency in 5G/6G are massive Multiple Input Multiple Output (MIMO) systems [19]. In an uplink scenario, a Base Station (BS) is equipped with a very large number of antennas (around 64-256) and simultaneously serves multiple single-antenna User Equipments (UEs) on the same time-frequency resource. As a first step in receiver design, different tasks such as channel equalization/estimation and decoding are typically split to lower complexity. But still, an algorithm deficit applies to both MIMO detection and decoding of large block-length codes, e.g., LDPC and Polar codes, since Maximum A Posteriori (MAP) detection has high computational complexity growing exponentially with system or code dimensions. Even its efficient implementation, the Sphere Detector (SD), remains too complex in such a scenario [20].

Hence, in communications history, many suboptimal solutions have been proposed to overcome the complexity bottleneck of the optimal detectors. One key approach is to relax the discrete Random Variables (RVs) to be continuous: Remarkable examples include Matched Filter (MF), zero forcing and MMSE equalization. But linear equalization with subsequent detection leads to a strong performance degradation compared to SD in symmetric systems.

A heuristic based on the latter is the V-Blast algorithm which first equalizes and then detects one layer with largest Signal-to-Noise Ratio (SNR) successively to reduce interference iteratively. A more efficient and sophisticated implementation, MMSE Ordered Successive Interference Cancellation (MOSIC), is based on a sorted QR Decomposition of a MMSE extended system matrix with post sorting and offers a good trade-off between complexity and accuracy [21].

Pursuing another philosophy of mathematical optimization, the SemiDefinite Relaxation (SDR) technique [22] treats MIMO detection as a non-convex homogeneous quadratically constrained quadratic problem and relaxes it to be convex by dropping the only non-convex requirement. Proving to be a close approximation, SDR is more complex than MOSIC and solved by interior point methods from convex optimization.

Furthermore, also probabilistic model-based ML techniques were introduced to improve the trade-off and to integrate detection seamlessly with decoding: Mean-Field Variational Inference (MFVI) provides a theoretical derivation of soft Successive Interference Cancellation (SIC) and the Bethe approach lays the foundation for loopy belief propagation [23]. Simplifying the latter, Approximate Message Passing (AMP) is derived known to be optimal for large system dimensions in i.i.d. Gaussian channels and compu-

tational cheap [24]. As a further benefit, soft outputs are computed, today a strict requirement to account for subsequent soft decoding. But in practice, the performance of probabilistic approximations like MFVI and AMP suffers if the approximating conditions are not met, i.e., from the fully-connected graph structure and finite dimensions in MIMO systems, respectively.

More recent work considers DNNs for application in MIMO systems and focus on the idea of deep unfolding [25], [26]. In deep unfolding, the number of iterations of a model-based iterative algorithm is fixed and its parameters untied. Further, it is enriched with additional weights and non-linearities to create a computational efficient DNN being optimized for performance improvements in MIMO detection [27], [28], belief propagation decoding [29]–[31] and MMSE channel estimation [32]. The former approach DetNet, a generic DNN model with a large number of trainable parameters based on an unfolded projected gradient descent, proves DNNs to allow for a promising trade-off between accuracy and complexity. In [33], unfolding of an extension of AMP to unitarily invariant channels, the Orthogonal Approximate Message Passing (OAMP), into OAMPNet is proposed adding only 2 trainable parameters per layer. Offering promising performance, the complexity bottleneck of one matrix inversion per iteration makes this model-driven approach rather unattractive compared to DetNet. Another DNN-like network MMNet is inspired by iterative soft thresholding algorithms and AMP [34]: Striking the balance between expressiveness and complexity, and exploiting spectral and temporal locality, MMNet can be trained online for any realistic channel realization if coherence time is large enough. Since online training is in general wasteful, an efficient implementation non-trivial and requires particularly deep analysis, we focus in this work on offline learning. One major drawback of the latter approaches is that they focus on MIMO detection and do not provide soft outputs.

3.2.3 Main Contributions

The main contributions of this article are manifold: Inspired by recent ML research, we first introduce a CONtinuous relaxation of the probability mass function (pmf) of the disCRETE RVs by a probability density function (pdf) from [35], [36] to the MAP detection problem. The proposed CONCRETE relaxation offers many favorable properties: On the one hand, the pdf of continuous RVs converges to the exact pmf in the parameter limit. On the other hand, we notice good algorithmic properties like avoiding marginalization and allowing for differentiation instead. By this means, we replace exhaustive search by computationally cheaper continuous optimization to approximately solve the MAP problem in any probabilistic non-linear model.

We name our approach Concrete MAP Detection (CMD).

Second, following the idea of Deep Unfolding, we unfold the gradient descent algorithm into a DNN-like model CMDNet with a fixed number of iterations to allow for parameter optimization and to further improve detection accuracy while limiting complexity. By this means, we are able to combine the advantages of DNNs and model-based approaches. As the number of parameters is small, we are able to dynamically adapt them to easily adjust CMDNet to different working points. Further, the resulting structure potentially allows for fast online training of CMDNet.

Thirdly, we derive the optimization criterion from an information theoretic perspective and are hence able to provide probabilities of detection, i.e., reliable soft outputs. We show that optimization is then equivalent to learning an approximation of the Individual Optimal (IO) detector. This allows us to account for subsequent decoding, e.g., in MIMO systems, in contrast to literature [28], [34].

Finally, we provide numerical simulation results for use of CMD and CMDNet in MIMO systems including a variety of simulation setups, e.g., correlated channels, revealing CMDNet to be a generic and promising approach competitive to State of the Art (SotA). Notably, we show superiority to other recently proposed ML-based approaches and demonstrate with simulations in coded systems CMDNet's soft outputs to be reliable for decoders as opposed to [28]. Furthermore, by estimating the computational complexity, we prove CMD to feature a promising trade-off between detection accuracy and complexity. Notably, only the matched filter has lower complexity.

In the following, we first introduce the concrete relaxation to MAP detection in Section 3.3 using the example of an inverse linear problem. In Section 3.4, we follow a different route and explain how to learn the posterior, i.e., replacing it by some tractable approximation. To yield a suitable model for this approximation, we propose to unfold CMD into CMDNet which we are then able to train by variants of Stochastic Gradient Descent (SGD). Finally, in Section 3.5 and 3.6, we provide numerical results of the bit error performance in comparison to other SotA approaches using the example of uncoded and coded MIMO systems and summarize the main results, respectively.

3.3 Concrete Relaxation of MAP problem

3.3.1 System Model and Problem Statement

To motivate the concrete relaxation, we consider a probabilistic and (possibly) non-linear observation model described by a continuous and differentiable

pdf $p(\mathbf{y}|\mathbf{x})$. Based on this model, the task is to classify/detect the discrete multivariate RV $\mathbf{x}$, i.e., $\mathbf{x} = \{x_n\}_{n=1}^{N_\mathrm{T}}$ whose i.i.d. elements are from a set $\mathcal{M}$, given the observation $\mathbf{y} \in \mathbb{C}^{N_\mathrm{R}\times 1}$.

To illustrate our findings with an example typically encountered in communications, we focus on a linear complex-valued observation model, e.g., MIMO system, although the following derivations hold without loss of generality for general $p(\mathbf{y}|\mathbf{x})$. We first exclude coding from our model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{3.1a}$$

$$\text{with} \quad p(\mathbf{y}|\mathbf{x}, \mathbf{H}, \sigma_\mathrm{n}^2) = \frac{1}{\pi^{N_\mathrm{R}} \sigma_\mathrm{n}^{2N_\mathrm{R}}} e^{-\frac{1}{\sigma_\mathrm{n}^2}(\mathbf{y}-\mathbf{H}\mathbf{x})^H(\mathbf{y}-\mathbf{H}\mathbf{x})} . \tag{3.1b}$$

There, a linear channel $\mathbf{H} \in \mathbb{C}^{N_\mathrm{R}\times N_\mathrm{T}}$ with statistic $p(\mathbf{H})$, e.g., such that taps $h_{mn} \sim \mathcal{N}_\mathbb{C}(0, 1/N_\mathrm{R})$ are i.i.d. Gaussian distributed, introduces correlation between the elements x_n with $\mathrm{E}[|x_n|^2] = 1$ from typical modulation sets $\mathcal{M}$, e.g., BPSK, 8-PSK or 16-QAM. Then, Gaussian noise $\mathbf{n} \sim \mathcal{N}_\mathbb{C}(\mathbf{0}, \sigma_\mathrm{n}^2 \mathbf{I}_{N_\mathrm{R}})$ with variance σ_n^2 distributed according to $p(\sigma_\mathrm{n}^2)$ interferes. The matrix $\mathbf{I}_{N_\mathrm{R}}$ denotes the identity matrix of dimension $N_\mathrm{R} \times N_\mathrm{R}$. For the following derivations, note that we are able to replace $\mathbf{y}$ by one total observation $\tilde{\mathbf{y}}$ including RVs $\mathbf{H}$ and σ_n^2 without loss of generality since $\mathbf{x}$, $\mathbf{H}$ and σ_n^2 are statistically independent:

$$p(\tilde{\mathbf{y}}|\mathbf{x}) = p(\mathbf{y}, \mathbf{H}, \sigma_\mathrm{n}^2|\mathbf{x}) = p(\mathbf{y}|\mathbf{x}, \mathbf{H}, \sigma_\mathrm{n}^2) \cdot p(\mathbf{H}) \cdot p(\sigma_\mathrm{n}^2) . \tag{3.2}$$

In this detection problem, there exist two optimal detectors from a probabilistic Bayesian viewpoint: First, we have the likelihood function $p(\mathbf{y}|\mathbf{x})$ but would like to infer the most likely transmit signal $\mathbf{x}$ based on an a-posteriori pdf $p(\mathbf{x}|\mathbf{y})$. Using Bayes rule, we are able to reform the MAP problem w.r.t. the known likelihood into

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}\times 1}}{\arg\max} \; p(\mathbf{x}|\mathbf{y}) \tag{3.3a}$$

$$= \underset{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}\times 1}}{\arg\max} \; p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x}) \tag{3.3b}$$

$$= \underset{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}\times 1}}{\arg\min} \; -\ln p(\mathbf{y}|\mathbf{x}) - \ln p(\mathbf{x}) \tag{3.3c}$$

where $p(\mathbf{x})$ is the known a-priori pdf. Since the RV is discrete, i.e., $x_n \in \mathcal{M}$, an exhaustive search over all element combinations is required to solve the MAP problem becoming computationally intractable for large system dimensions. Note that the Sphere Detector (SD) provides an efficient implementation [20]. Second, we notice that the MAP detector only delivers the most likely received vector $\mathbf{x}$. Hence, it minimizes frame error rate and provides hard decisions.

In coded systems with soft decoders usually employed today, delivering soft information is a strict requirement. The Individual Optimal (IO) detector delivers such soft output as probabilities and is optimal in terms of minimizing the Symbol Error Rate (SER) per individual symbol without coding. It is obtained by evaluating the marginal posterior distribution w.r.t. every single x_n:

$$\hat{x}_n = \underset{x_n \in \mathcal{M}}{\arg\max}\; p(x_n|\mathbf{y}) = \underset{x_n \in \mathcal{M}}{\arg\max}\; \frac{\sum\limits_{\mathbf{x}\setminus x_n} p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})}{\sum\limits_{x_n} \sum\limits_{\mathbf{x}\setminus x_n} p(\mathbf{y}|\mathbf{x}) \cdot p(\mathbf{x})}\,. \tag{3.4}$$

However, it has higher complexity due to required marginalization w.r.t. $\mathbf{x}$. Since the MAP detector performance coincides with the IO detector in the high SNR regime and is of lower complexity, we restrict to the MAP detector as a benchmark in simulations without coding.

3.3.2 Concrete Distribution

We now focus on the following question to improve the performance complexity trade-off: How to model the prior information $p(\mathbf{x})$ accurately by some approximation $p(\tilde{\mathbf{x}})$? In [37], we proposed to use ML tricks from [35], [36] to achieve this and to make inference computationally tractable. The idea was recently discovered in the ML community in the context of unsupervised learning of generative models [35], [36]. There, marginalization to compute the objective function, the evidence, becomes intractable. Therefore, the Evidence is replaced by its Evidence Lower BOund (ELBO) by means of an auxiliary posterior function. But optimizing w.r.t. the ELBO results in high variance of the gradient estimators. For variance reduction, the so called reparametrization trick is used and leads to an optimization structure similar to an autoencoder known as the variational autoencoder [23]. There, the stochastic node is reparametrized by a continuous RV, e.g., a Gaussian, and its parameters, e.g., mean and variance. In contrast to continuous RVs, reparametrization of discrete RVs is not possible. Hence, a CONtinuous relaxation of disCRETE RVs, the CONCRETE distribution, was proposed in [35], [36] independently.

To explain the introduction of this relaxation to MAP detection, let us assume that we have the discrete binary RV $x \in \mathcal{M}$ with $\mathcal{M} = \{-1, +1\}$. Further, we define the discrete RV $\mathbf{z}$ as a one-hot vector where all elements are zero except for one element, i.e., $\mathbf{z} \in \{0, 1\}^{2\times 1}$ with two possible realizations $\mathbf{z}_1 = [1, 0]^T$, $\mathbf{z}_2 = [0, 1]^T$. In addition, we describe the values of $\mathcal{M}$ by the representing vector $\mathbf{m} = [-1, 1]^T$. That way, we can write $x = \mathbf{z}^T\mathbf{m}$, e.g.,

$x = [1,0] \cdot [-1,1]^T = -1$. Now, the one-hot vector $\mathbf{z} \in \{0,1\}^{M\times 1}$ represents a categorical RV with $M = |\mathcal{M}|$ classes. Connecting Monte Carlo methods to optimization [35], the Gumbel-Max trick states that we are able to generate samples, i.e., classes, of such a categorical RV or pmf $p(x)$ by sampling an index i^* from M continuous i.i.d. Gumbel RVs g_i known from extreme value theory:

$$i^* = \underset{i=1,\dots,M}{\arg\max} \ \ln p(x = m_i) + g_i \,. \tag{3.5}$$

Defining the function one-hot(i^*) which sets the i^*-th element in the one-hot vector $z_{i^*} = 1$ and $z_{l\neq i^*} = 0$, the Gumbel-Max trick hence allows sampling one-hot vectors $\mathbf{z}$. Thus, we are able to reparametrize $\mathbf{z}$ through a continuous multivariate Gumbel RV $\mathbf{g} \in \mathbb{R}^{M\times 1}$ and a vector $\boldsymbol{\alpha} \in [0,1]^{M\times 1}$ of class probabilities $p(x = m_k)$ with $\sum_{k=1}^{M} \alpha_k = 1$:

$$\mathbf{z} = \text{one-hot}\left(\underset{i=1,\dots,M}{\arg\max} \ [\ln(\boldsymbol{\alpha}) + \mathbf{g}]\right) . \tag{3.6}$$

Note that (3.6) and equally x are still discrete RVs, i.e., $p(\mathbf{z}) \triangleq p(x)$, but represented in probabilistic sense by continuous RVs $\mathbf{g}$. To arrive at a continuous RV, we now replace the one-hot and arg max computation in (3.6) by the softmax function [35], [36]:

$$\tilde{\mathbf{z}} = \sigma_\tau(\mathbf{g}) = \frac{e^{(\ln(\boldsymbol{\alpha})+\mathbf{g})/\tau}}{\sum_{i=1}^{M} e^{(\ln(\alpha_i)+g_i)/\tau}} \,. \tag{3.7}$$

The resulting RV $\tilde{\mathbf{z}} \in [0,1]^{M\times 1}$ is the so-called concrete or Gumbel-Softmax RV and now continuous, e.g., $\tilde{\mathbf{z}} = [0.2, 0.8]^T$. It is controlled by a parameter, the softmax temperature τ. The distribution of $\tilde{\mathbf{z}}$ in (3.7) was found to have a closed form density in [35], [36] which gives the definition of the concrete distribution:

$$p(\tilde{\mathbf{z}}|\boldsymbol{\alpha}, \tau) = (M-1)!\ \tau^{M-1} \prod_{k=1}^{M} \left(\frac{\alpha_k \tilde{z}_k^{-\tau-1}}{\sum_{i=1}^{M} \alpha_i \tilde{z}_i^{-\tau}}\right) . \tag{3.8}$$

With $\tilde{\mathbf{z}}$, we are finally able to relax the discrete RV x into a continuous RV $\tilde{x}$ by defining $\tilde{x} = \tilde{\mathbf{z}}^T \mathbf{m}$. Now, our derivation of the relaxation is complete. In Fig. 3.1, we illustrate the distribution $p(\tilde{x})$ for the special case $M = 2$ of binary RVs in comparison to the original categorical pmf $p(x)$, i.e., a Bernoulli pmf. It has the following properties [35]: First, we are able to reparametrize the concrete RV $\tilde{\mathbf{z}}$ and hence the RV $\tilde{x}$ by Gumbel variables $\mathbf{g}$, a direct result from the initial idea (3.7). Moreover, the smaller τ, the more $\tilde{\mathbf{z}}$ approaches a categorical RV and the approximation becomes more accurate. Thus, the statistics of x and $\tilde{x}$ remain the same for $\tau \to 0$.

3.3.3 Reparametrization

In [37], our idea is to use the concrete distribution in order to relax the MAP problem (3.3c) to

$$\hat{\mathbf{x}} = \underset{\tilde{\mathbf{x}} \in [\min(\mathcal{M}), \max(\mathcal{M})]^{N_\mathrm{T} \times 1}}{\arg\min} \; - \ln p(\mathbf{y}|\tilde{\mathbf{x}}) - \ln p(\tilde{\mathbf{x}}) \,. \tag{3.9}$$

Note that the original MAP problem is included or recovered in the zero temperature limit $\tau \to 0$. Moreover, the objective function in (3.9) may be non-convex as illustrated in Fig. 3.2 for $M = 2$. The conditional pdf $p(\mathbf{y}|\tilde{\mathbf{x}})$ is log-concave and the prior concrete pdf $p(\tilde{\mathbf{x}})$ log-convex for $\tau \leq (M-1)^{-1}$ [35], so the negative log joint distribution $p(\mathbf{y}, \tilde{\mathbf{x}})$ forms a non-convex objective function (3.9). The reparametrization of $\tilde{\mathbf{z}}$ by $\mathbf{g}$ helps to rewrite (3.9) by expressing each $\tilde{x}_n$ in $\tilde{\mathbf{x}}$ with (3.7) by the RV $\mathbf{g}_n$, $n = 1, \ldots, N_\mathrm{T}$, of i.i.d. Gumbel RVs g_{kn}:

$$\tilde{\mathbf{x}}(\mathbf{G}) = \begin{bmatrix} \tilde{x}_1 \\ \vdots \\ \tilde{x}_{N_\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{z}}_1^T \\ \vdots \\ \tilde{\mathbf{z}}_{N_\mathrm{T}}^T \end{bmatrix} \mathbf{m} = \begin{bmatrix} \sigma_\tau \left(\mathbf{g}_1\right)^T \\ \vdots \\ \sigma_\tau \left(\mathbf{g}_{N_\mathrm{T}}\right)^T \end{bmatrix} \mathbf{m} \tag{3.10}$$

$$\text{with} \quad \mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \cdots & \mathbf{g}_{N_\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{M \times N_\mathrm{T}} \,. \tag{3.11}$$

By doing so, we will obtain an unconstrained optimization problem w.r.t. matrix $\mathbf{G}$. Now, we reformulate the relaxed MAP problem (3.9): This means, we replace the likelihood $p(\mathbf{y}|\tilde{\mathbf{x}})$ by $p(\mathbf{y}|\mathbf{G})$ and introduce the Gumbel distribution $p(g_{kn}) = \exp\left(-g_{kn} - \exp\left(-g_{kn}\right)\right)$ as the new prior distribution:

$$\hat{\mathbf{G}} = \underset{\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}}{\arg\min} \; - \ln p(\mathbf{y}|\mathbf{G}) - \ln p(\mathbf{G}) \tag{3.12a}$$

$$= \underset{\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}}{\arg\min} \; - \ln p(\mathbf{y}|\mathbf{G}) - \sum_{n=1}^{N_\mathrm{T}} \sum_{k=1}^{M} \ln p(g_{kn}) \tag{3.12b}$$

$$= \underset{\mathbf{G} \in \mathbb{R}^{M \times N_\mathrm{T}}}{\arg\min} \; \underbrace{- \ln p(\mathbf{y}|\mathbf{G}) + \mathbf{1}^T \mathbf{G} \mathbf{1} + \mathbf{1}^T e^{-\mathbf{G}} \mathbf{1}}_{L(\mathbf{G}, \tau)} \,. \tag{3.12c}$$

However, due to the softmax and exponential terms in $L(\mathbf{G}, \tau)$, (3.12c) has no analytical solution. Furthermore, $L(\mathbf{G}, \tau)$ may be non-convex: For real-valued model (3.1) with $N_\mathrm{T} = 1$, $\mathbf{H} = 1$ and $M = 2$, the first term is a vertically shifted, squared and scaled two-dimensional non-convex sigmoid function w.r.t. g_1 and g_2. The operations applied to the sigmoid do not change non-convexity. Also, the sum of this non-convex term and convex functions, i.e., linear and exponential functions, remains non-convex.

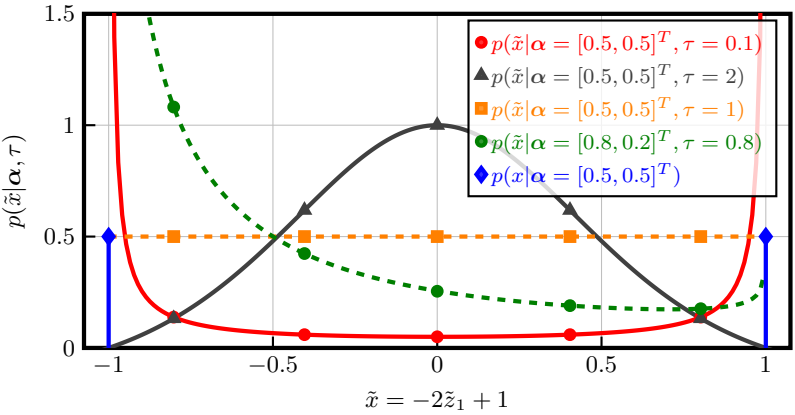

Figure 3.1: The concrete pdf $p(\tilde{x}|\boldsymbol{\alpha}, \tau)$ shown for different parameter sets and $M = 2$. It relaxes the Bernoulli pmf $p(x|\boldsymbol{\alpha})$ into the interior. Notably, for $\tau \leq (M-1)^{-1}$, it is log-convex and log-concave otherwise. Symmetry results if $\alpha_1 = \ldots = \alpha_M$.

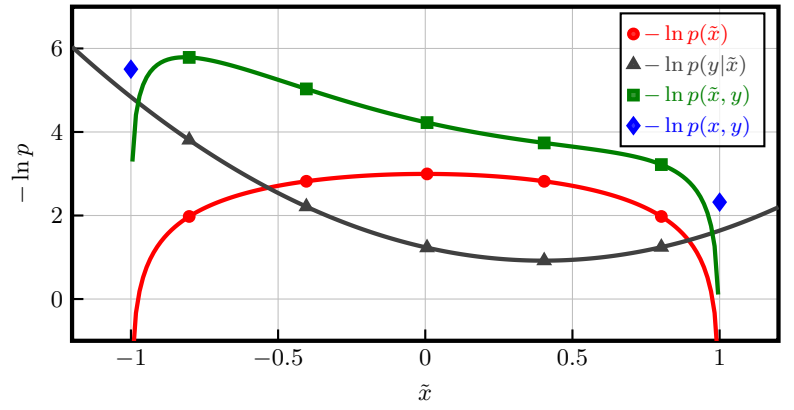

Figure 3.2: Exemplary plot of the concrete binary MAP cost function (green) for model (3.1) (with $N_\mathrm{T} = 1$, $\mathbf{H} = 1$, $y = 0.4$, $\sigma_\mathrm{n}^2 = 4$, $\alpha_1 = 0.5$, $\alpha_2 = 0.5$ and $\tau = 0.1$) and the contribution of conditional (black) and prior pdf (red) to it. The original binary MAP cost function (blue) is shown for comparison.

3.3.4 Gradient Descent Optimization

One common strategy for solving the non-linear and non-analytical problem (3.12c) is to use a variant of gradient descent based approaches. Since we aim to reduce complexity, we choose the most basic form the steepest descent. The minimum is approached iteratively by taking gradient descent steps until the necessary condition

$$\frac{\partial L(\mathbf{G}, \tau)}{\partial \mathbf{G}} = \mathbf{0} \tag{3.13}$$

is fulfilled. We point out that convergence to the global solution depends heavily on the starting point initialization since the objective function may be non-convex. A reasonable choice of starting point value is $\tilde{\mathbf{x}}^{(0)} = \mathrm{E}[\mathbf{x}] = \boldsymbol{\alpha}^T \cdot \mathbf{m}$, i.e., the expected value of the true discrete RV $\mathbf{x}$. We achieve this by setting $\mathbf{G}^{(0)} = \mathbf{0}$ and $\tau = 1$. After some tensor/matrix calculus and by noting that every $\tilde{x}_n$ only depends on one $\mathbf{g}_n$, the gradient descent step for (3.12c) in iteration j is:

$$\mathbf{G}^{(j+1)} = \mathbf{G}^{(j)} - \delta^{(j)} \cdot \left.\frac{\partial L(\mathbf{G}, \tau)}{\partial \mathbf{G}}\right|_{\mathbf{G}=\mathbf{G}^{(j)}} \tag{3.14a}$$

$$\begin{aligned}\frac{\partial L(\mathbf{G}, \tau)}{\partial \mathbf{G}} = &- \begin{bmatrix} \frac{\partial \tilde{x}_1(\mathbf{g}_1)}{\partial \mathbf{g}_1} & \cdots & \frac{\partial \tilde{x}_{N_\mathrm{T}}(\mathbf{g}_{N_\mathrm{T}})}{\partial \mathbf{g}_{N_\mathrm{T}}} \end{bmatrix} \\ &\cdot \operatorname{diag}\left\{ \frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \tilde{\mathbf{x}}} \right\} + 1 - e^{-\mathbf{G}}\end{aligned} \tag{3.14b}$$

$$\frac{\partial \tilde{x}_n(\mathbf{g}_n)}{\partial \mathbf{g}_n} = \frac{1}{\tau^{(j)}} \cdot \left[\operatorname{diag}\left\{\sigma_\tau\left(\mathbf{g}_n\right)\right\} \cdot \mathbf{m} - \sigma_\tau\left(\mathbf{g}_n\right) \cdot \tilde{x}_n(\mathbf{g}_n)\right] . \tag{3.14c}$$

The operator $\operatorname{diag}\{\mathbf{a}\}$ creates a diagonal matrix with the vector $\mathbf{a}$ on its main diagonal. The step-size $\delta^{(j)}$ can be chosen adaptively in every iteration j just as the parameter $\tau^{(j)}$. For example, we can follow a heuristic schedule like in simulated annealing: We start with a large $\tau^{(j)}$ and decrease until we approach the true prior pdf for $\tau^{(j)} \to 0$. Finally, after the last iteration N_it, we get as a result the continuous estimate $\mathbf{G}^{(N_\mathrm{it})}$. For approximate detection of $\mathbf{x}$ in (3.3c), the estimate has to be transformed back to the discrete domain by quantizing $\tilde{\mathbf{x}}$ onto the discrete set $\mathcal{M}$:

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{M}^{N_\mathrm{T} \times 1}}{\arg\min} \left\| \mathbf{x} - \tilde{\mathbf{x}}\left(\mathbf{G}^{(N_\mathrm{it})}\right) \right\|_2 . \tag{3.15}$$

In the following, we name this detection approach Concrete MAP Detection (CMD). It is generic and applicable in any differentiable probabilistic non-linear model. For our guiding example of a linear Gaussian model (3.1), we

are able to give the explicit expression of

$$\frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \tilde{\mathbf{x}}} = -\frac{2}{\sigma_{\mathrm{n}}^2} \cdot \left[\mathbf{H}^H \mathbf{H} \tilde{\mathbf{x}}(\mathbf{G}) - \mathbf{H}^H \mathbf{y}\right] \tag{3.16}$$

in (3.14b). This means that further only element-wise non-linearities and matrix vector multiplications are present in this example. As a final remark, we note that our implementation of Section 3.5 relies on scaling of the objective function by the noise variance parameter, i.e., $\sigma_{\mathrm{n}}^2 \cdot L(\mathbf{G}, \tau)$. Although scaling does not change the optimization problem, we observed that this slightly modified version of (3.14) is numerically more stable.

3.3.5 Special Case: Binary Random Variables

Noting that the softmax function (3.7) is normalized, we are able to eliminate one degree of freedom in matrix $\mathbf{G} \in \mathbb{R}^{M \times N_{\mathrm{T}}}$ along dimension M. For the special case of binary RVs or $M = 2$ classes, this means that the matrix $\mathbf{G}$ can be reduced to a vector $\mathbf{s} \in \mathbb{R}^{N_{\mathrm{T}} \times 1}$ of logistic RVs to derive a different algorithm of low complexity. Here, we only briefly summarize the result of binary CMD in a real-valued system model and refer the reader to [37] for the complete derivation:

$$\mathbf{s}^{(j+1)} = \mathbf{s}^{(j)} - \delta^{(j)} \cdot \left.\frac{\partial L(\mathbf{s}, \tau)}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{s}^{(j)}} \tag{3.17a}$$

$$\frac{\partial L(\mathbf{s}, \tau)}{\partial \mathbf{s}} = -\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \frac{\partial \ln p(\mathbf{y}|\mathbf{s})}{\partial \tilde{\mathbf{x}}} + \tanh\left(\frac{\mathbf{s}}{2}\right) \tag{3.17b}$$

$$\overset{(3.1)}{=} \frac{1}{\sigma_{\mathrm{n}}^2} \cdot \frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \left[\mathbf{H}^T \mathbf{H} \tilde{\mathbf{x}}(\mathbf{s}) - \mathbf{H}^T \mathbf{y}\right] + \tanh\left(\frac{\mathbf{s}}{2}\right) \tag{3.17c}$$

$$\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} = \frac{1}{2\tau^{(j)}} \cdot \mathrm{diag}\left\{1 - \tilde{\mathbf{x}}^2(\mathbf{s})\right\} \tag{3.17d}$$

$$\tilde{\mathbf{x}}(\mathbf{s}) = \tanh\left(\frac{\ln\left(1/\alpha - 1\right) + \mathbf{s}}{2\tau^{(j)}}\right). \tag{3.17e}$$

The final step consists again of quantization — in this case it simplifies to the sign function: $\hat{\mathbf{x}} = \mathrm{sign}(\tilde{\mathbf{x}}(\mathbf{s}^{(N_{\mathrm{it}})}))$.

3.4 Learning to Relax

Although being simple and computational efficient, using a gradient descent approach like (3.14) and (3.17) leads to several inconveniences. Regarding theoretical properties, a major drawback becomes apparent: Convergence

of the gradient descent steps to an optimum is slow since consecutive gradients are perpendicular. Also, practical questions arise: How to choose the parameters $\tau^{(j)}$ and $\delta^{(j)}$ and the number of iterations N_{it} for a good complexity–performance trade-off? And how are we able to deliver soft information, e.g., probabilities, to a soft decoder which is standard in today's communication systems?

Our idea is to improve CMD by learning and in particular the idea of deep unfolding to address these questions. This means we have to deal with

A. how learning is defined

B. the application of deep unfolding to CMD.

3.4.1 Basic Problem of Learning

To introduce our notation of learning, we revisit our basic task of MAP detection. Ideally, we would like to infer the most likely transmit signal $\mathbf{x}$ based on an a-posteriori pdf $p(\mathbf{x}|\mathbf{y})$. But as pointed out earlier, evaluation of $p(\mathbf{x}|\mathbf{y})$ has intractable complexity. For this reason, we propose to relax the MAP problem and CMD, respectively.

Another idea to tackle this problem is to approximate this pdf $p(\mathbf{x}|\mathbf{y})$ by another computationally tractable pdf $q(\mathbf{x}|\mathbf{y})$, e.g., by calculation of $q(\mathbf{x}|\mathbf{y})$ using few samples/observations $\mathbf{x}$, and to use this pdf for inference. Note that this approach includes cases where we do not know the pdf $p(\mathbf{x}|\mathbf{y})$ completely. The quality of the approximation can be quantified by the information theoretic measure of Kullback–Leibler (KL) divergence:

$$D_{\mathrm{KL}}\left(p \parallel q\right) = \sum_{\mathbf{x}\in\mathcal{M}^{N_{\mathrm{T}}\times 1}} p(\mathbf{x}|\mathbf{y}) \ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x}|\mathbf{y})} \tag{3.18}$$

$$= \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}\left[\ln \frac{p(\mathbf{x}|\mathbf{y})}{q(\mathbf{x}|\mathbf{y})}\right] . \tag{3.19}$$

Just as the Mean Square Error (MSE), the measure of KL divergence can be used to define an optimization problem targeting at a tight $q(\mathbf{x}|\mathbf{y})$ as a solution. This brings me to a crucial viewpoint of this article: **Learning is defined to be the optimization process aiming to derive a good approximation $q(\mathbf{x}|\mathbf{y})$ of $p(\mathbf{x}|\mathbf{y})$, i.e.,**

$$q^*(\mathbf{x}|\mathbf{y}) = \arg\min_{q} \; D_{\mathrm{KL}}\left(p \parallel q\right) . \tag{3.20}$$

This kind of problem is also referred to as Variational Inference (VI). We can rewrite the KL divergence into a sum of cross-entropy $\mathcal{H}\left(p, q\right)$ and

entropy $\mathcal{H}(p)$:

$$D_{\mathrm{KL}}(p \parallel q) = \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y})] - \mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln p(\mathbf{x}|\mathbf{y})] \tag{3.21}$$

$$= \mathcal{H}(p,q) - \mathcal{H}(p)\ . \tag{3.22}$$

Since we defined the basic learning problem (3.20) w.r.t. approximation q, we can neglect the entropy term $\mathcal{H}(p)$ independent of q and use cross-entropy as the learning criterion. If we further restrict q to a model $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$, the optimization problem now reads:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{H}(p,q)\ . \tag{3.23}$$

We note that problem (3.23) is solved separately for each $\mathbf{y}$ and thus parameters $\boldsymbol{\theta}$ need to be continuously updated in an online learning procedure. Since this procedure is not computationally efficient, we follow an offline learning strategy known as Amortized Inference [23] and define one inference distribution $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ for any value $\mathbf{y}$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}}\ \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}(p(\mathbf{x}|\mathbf{y}), q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}))] \tag{3.24}$$

$$= \arg\min_{\boldsymbol{\theta}}\ \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}\left[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\mathbf{y})}[-\ln q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})]\right] \tag{3.25}$$

$$\approx \arg\min_{\boldsymbol{\theta}}\ -\frac{1}{N}\sum_{i=1}^{N} \ln q(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\theta}) \quad , N\to\infty\,. \tag{3.26}$$

Rewriting the optimization criterion of (3.24) into

$$\mathrm{E}_{\tilde{\mathbf{y}}\sim p(\tilde{\mathbf{y}})}\left[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\tilde{\mathbf{y}})}[-\ln q(\mathbf{x}|\tilde{\mathbf{y}},\boldsymbol{\theta})]\right] \tag{3.27}$$

$$= \mathrm{E}_{\sigma_\mathrm{n}^2\sim p(\sigma_\mathrm{n}^2)}\left[\mathrm{E}_{\mathbf{H}\sim p(\mathbf{H})}\left[\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{H},\sigma_\mathrm{n}^2)}\left[\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x}|\tilde{\mathbf{y}})}[-\ln q(\mathbf{x}|\tilde{\mathbf{y}},\boldsymbol{\theta})]\right]\right]\right]$$

for our guiding example (3.1), we note that we are able to amortize across all observations $\tilde{\mathbf{y}}$ from (3.2) and hence to obviate the need for online training also for each channel $\mathbf{H}$ and noise variance σ_n^2 at the potential cost of accuracy.

The final result (3.26) equals the maximum likelihood problem in supervised learning. We make use of it in the following since it allows for numerical optimization based on N data points $\{\mathbf{x}_i, \mathbf{y}_i\}$. Furthermore, it proves to be a Monte Carlo approximation of (3.24) and is hence well motivated from information theory [23].

3.4.2 Idea of Unfolding and Application to CMD

Learning gives us the ability to obtain a tractable approximation $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$. But it remains one question: How to choose a suitable functional form of

$q(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ of low complexity and for good performance? We follow the idea of deep unfolding from [25], [26] and apply it to our model-based approach CMD with parameters $\boldsymbol{\theta} = \left\{\tau^{(0)}, \ldots, \tau^{(N_{\mathrm{it}})}, \delta^{(0)}, \ldots, \delta^{(N_{\mathrm{it}}-1)}\right\} \in \mathbb{R}^{(2N_{\mathrm{it}}+1)\times 1}$ able to relax tightly. Thereby, we combine strengths of DNNs and the latter: DNNs are known to be universal approximators [5] and their fixed structure of parallel computations layer per layer allows to define a good performance complexity trade off at run time. But if the model is dynamic and changes, e.g., the channel or noise over time, reiterated optimization of (3.23), i.e., possibly wasteful online training, is required and the benefit disappears. Fortunately, we know our model (3.1), a MIMO channel, well and are able to use generative model-based approaches which mostly rely on a suitable approximation of (3.20) for computational tractability. For example, MFVI and AMP belong to this algorithm family. By model-based DNN design, we introduce varying model parameters like channel or noise explicitly and in a more sophisticated way into the DNN design and thus make efficient offline learning from (3.26) at only a small cost of accuracy possible. Indeed, training of a DNN for our guiding example (3.1) simply fed with inputs $\mathbf{y}$ and $\mathbf{H}$, reshaped as a vector, does not converge/lead to satisfactory results if trained offline [28].

This means we unfold the iterations (3.14) of CMD into a DNN by untying the parameters $\tau^{(j)}$ and $\delta^{(j)}$. Furthermore, we fix the complexity by setting the number of iterations N_{it}. The resulting graph illustrated in Fig. 3.3 for binary CMD and (3.1) has a DNN-like structure which should be able to generalize and approximate well at the same time. Owing to the skip connection from $\mathbf{s}^{(j)}$ to $\mathbf{s}^{(j+1)}$ on the right-hand side, the structure resembles a Residual Network (ResNet) layer which is SotA in image processing [9]. It is a result of the gradient descent approach which allows interpreting optimization of ResNets as learning gradient descent steps. The reason for the success of ResNet lies in the skip connection: The training error is able to backpropagate through it to early layers which allows for fast adaptation of early weights and hence fast training of DNNs. This makes CMD especially suitable for online training proposed in [34] and allows for refinement in application.

As before, we have to define a final layer which is now also used for optimization. Usually, its output is chosen to be a continuous estimate of $\mathbf{x}$ and optimized w.r.t. the MSE criterion, see [28], [34]. This viewpoint relaxes the estimate $\hat{\mathbf{x}}$ into $\mathbb{R}^{N_{\mathrm{T}}\times 1}$ and assumes a Gaussian distribution for errors at the output. In our case, the output would correspond to $\tilde{\mathbf{x}}(\mathbf{G}^{(N_{\mathrm{it}})})$ from (3.15). But this is in contrast to our information theoretic viewpoint on learning which states that we want to approximate an output of the true pmf $p(\mathbf{x}|\mathbf{y})$. Like in MFVI, we assume a factorization of the

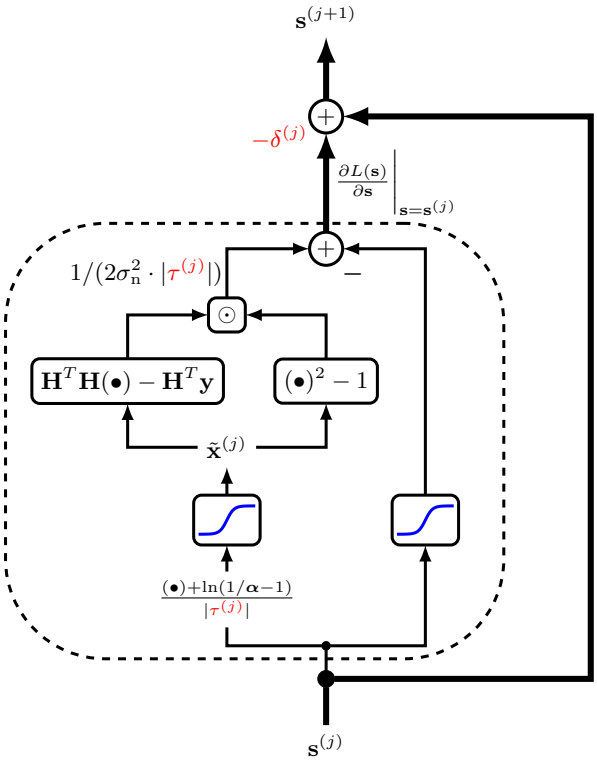

Figure 3.3: One layer of the unfolded binary CMD algorithm CMDNet when applied to MIMO systems. In red: trainable parameters.

approximating posterior to make it computationally tractable and derive our learning criterion:

$$\mathcal{H}(p,q) = -\sum_{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}}} p(\mathbf{x}|\mathbf{y}) \cdot \ln q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) \tag{3.28}$$

$$\overset{\text{MFVI}}{=} -\sum_{\mathbf{x}\in\mathcal{M}^{N_\mathrm{T}}} p(\mathbf{x}|\mathbf{y}) \cdot \ln \prod_{n=1}^{N_\mathrm{T}} q_n(x_n|\mathbf{y},\boldsymbol{\theta}) \tag{3.29}$$

$$= -\sum_{n} \sum_{x_n\in\mathcal{M}} \sum_{\mathbf{x}\setminus x_n\in\mathcal{M}^{N_\mathrm{T}-1}} p(\mathbf{x}|\mathbf{y}) \cdot \ln q_n(x_n|\mathbf{y},\boldsymbol{\theta})$$

$$\mathcal{H}\left(p,q\right) = -\sum_{n}\sum_{x_n\in\mathcal{M}} p(x_n|\mathbf{y})\cdot\ln q_n(x_n|\mathbf{y},\boldsymbol{\theta}) \tag{3.30}$$

$$= \sum_{n=1}^{N_\mathrm{T}} \mathcal{H}\left(p(x_n|\mathbf{y}), q_n(x_n|\mathbf{y},\boldsymbol{\theta})\right) . \tag{3.31}$$

This interesting result shows that assuming MFVI factorization leads to an optimization criterion w.r.t. the soft output $p(x_n|\mathbf{y})$ of the IO detector (3.4). This soft output is required for subsequent decoding and thus exactly what we need.

The last step of our idea consists of inserting our unfolded CMD structure into $q_n(x_n|\mathbf{y},\boldsymbol{\theta})$. Hence, we propose to use a softmax function for the last layer being a typical choice for classification in discriminative probabilistic models. Fortunately, CMD already includes this softmax function as part of its structure, so we rewrite

$$q_n(x_n|\mathbf{y},\boldsymbol{\theta}) = \prod_{k=1}^{M} q_{n,k}(x_n|\mathbf{y},\boldsymbol{\theta})^{(x_n=m_k)} = \prod_{k=1}^{M} \tilde{z}_{n,k}^{(x_n=m_k)} \tag{3.32}$$

with $\tilde{\mathbf{z}}_n = \sigma_{\tau^{(N_\mathrm{it})}}(\mathbf{g}_n^{(N_\mathrm{it})})$ from the last iteration N_it of (3.14). To summarize, we optimize the parameter set $\boldsymbol{\theta}$ of our approximating pdf $q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ based on CMD:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \ \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}\left(p(\mathbf{x}|\mathbf{y}), q(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})\right)] \tag{3.33}$$

$$\approx \arg\min_{\boldsymbol{\theta}} \ -\frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{N_\mathrm{T}} \begin{bmatrix} x_{i,n}=m_1 \\ \vdots \\ x_{i,n}=m_M \end{bmatrix}^T \ln\left(\sigma_{\tau^{(N_\mathrm{it})}}(\mathbf{g}_n^{(N_\mathrm{it})})\right) . \tag{3.34}$$

As a side effect, we also learn to relax with CMD by $\tau^{(j)}$. We call this approach based on unfolding of CMD CMDNet. The optimization problem (3.34) can be efficiently solved by variants of SGD. Thanks to having a model, we are able to create infinite training and test data for reasonable approximation of (3.33) by (3.34) in every iteration of SGD. We notice that this is in contrast to classic data sets from the machine learning community.

Table 3.1: Simulation Scenarios

Scenario	Sys. Dim.	Mod.	Corr.	Coding
Large MIMO	32×32	QPSK	no	no
MIMO	8×8	QPSK	no	no
Multi-class	32×32	16-QAM	no	no
Massive MIMO One-Ring	64×32	QPSK	$20°$	no
Soft Output	32×32	QPSK	no	LDPC

3.5 Numerical Results

3.5.1 Implementation Details / Settings

In order to evaluate the performance of the proposed approaches CMD and CMDNet, we present numerical simulation results of application in our guiding example for different MIMO systems with N_{T} transmit and N_{R} receive antennas given in Tab. 3.1. We assume an uplink scenario with multiple UEs, each transmitting one symbol x_n with equal a-priori probabilities $\alpha_1 = \ldots = \alpha_M$ to one BS. As an example, we assume the number of iterations or layers to be $N_{\mathrm{it}} = N_{\mathrm{L}} = 2N_{\mathrm{T}}$. For numerical optimization of the parameters $\delta^{(j)}$ and $\tau^{(j)}$ of CMDNet according to (3.34), we employ the TensorFlow framework in Python [7]. Here, we use Adaptive Moment Estimation (Adam) as a popular variant of SGD with a default batch size of $N_{\mathrm{b}} = 500$ and $N_{\mathrm{e}} = 10^5$ training iterations. Although providing fast convergence and requiring little hyperparameter tuning, it is known to generalize poorly [38]. Since we are able to generate a sufficient amount of training data, i.e., $N = N_{\mathrm{b}} \cdot N_{\mathrm{e}} = 5 \cdot 10^7$ to fulfill (3.33) by (3.34) approximately, we make sure that generalization to unseen data points is possible. As TensorFlow does not natively support computation with complex numbers, we transform the complex-valued system model (3.1) into its real-valued equivalent to allow for training and comparison to DNN-based approaches. This means, we restrict to QAM constellations with Gray encoding so that we have $\mathbf{x} \in \mathcal{M}^{2N_{\mathrm{T}} \times 1}$. As a training default, we choose the noise variance statistics $p(\sigma_{\mathrm{n}}^2)$ such that $E_{\mathrm{b}}/N_0 = 10\log_{10}(1/\sigma_{\mathrm{n}}^2) - 10\log_{10}(\log_2(M))$ is uniformly distributed between $[4, 27]$ dB. We set the default parameter starting point to $\boldsymbol{\theta}_0$ with constant $\delta_0^{(j)} = 1$ and heuristically

Table 3.2: Selected detection algorithms

Abbreviation	Complexity	Literature
MAP/SD	$\mathcal{O}(M^{\gamma N_\mathrm{T}})$, $\gamma \in (0, 1]$	[20]
SDR	$\mathcal{O}(\max(N_\mathrm{R}, N_\mathrm{T})^3 N_\mathrm{T}^{1/2} \log(1/\epsilon))$	[22]
OAMPNet	$\mathcal{O}(N_\mathrm{L} N_\mathrm{T}^3)$	[33]
MMSE/MOSIC	$\mathcal{O}(N_\mathrm{T}^3)$	[21], [34]
DetNet	$\mathcal{O}(N_\mathrm{L}(N_\mathrm{T} N_\mathrm{R} + N_\mathrm{T}^2 M))$	[27], [28]
MMNet (iid)	$\mathcal{O}(N_\mathrm{L} N_\mathrm{T}(N_\mathrm{T} + N_\mathrm{R} + M))$	[34]
AMP	$\mathcal{O}(N_\mathrm{it} N_\mathrm{T}(N_\mathrm{R} + M))$	[24]
CMD/CMDNet	$\mathcal{O}(N_\mathrm{L} N_\mathrm{T}(N_\mathrm{R} + M))$	[37]
MF	$\mathcal{O}(N_\mathrm{T} N_\mathrm{R})$	

motivated and linear decreasing

$$\tau_0^{(j)} = \tau_\mathrm{max} - (\tau_\mathrm{max} - 0.1)/N_\mathrm{it} \cdot j \tag{3.35}$$

with $\tau_\mathrm{max} = 1/(M-1)$, $j \in [0, N_\mathrm{it}]$. With this choice, $p(\tilde{x})$ is always log-convex and hence reasonably approximating $p(x)$ (see Fig. 3.1). For training of DNN-based approaches DetNet and MMNet, we used the original implementations uploaded to GitHub (see [28], [34]) with only minor modifications to parametrization if beneficial. Consequently, we trained MMNet with CMDNet training SNR and layer number. Since we focus on offline derived or trained algorithms which are used for inference at run time, we used its i.i.d. variant. We always used the soft output version of DetNet with output normalization to 1 since we noted that performance is close to or better than the hard decision version. Furthermore, we compare CMD and CMDNet to several SotA approaches for MIMO detection (see Tab. 3.2) choosing the number of Monte Carlo runs with data batches of size 10000 so that always 1000 errors are detected (100 for SD and SDR).

3.5.2 Symmetric MIMO System

First, we test application of CMDNet in a large symmetric 32×32 / 64×64 MIMO system with i.i.d. Gaussian channel statistics $p(\mathbf{H})$ and QPSK/BPSK modulation. Fig. 3.4 shows the results in terms of BER as a function of E_b/N_0. Owing to near-optimal performance, the SD is always provided as

a benchmark in the following. In addition, we give the AWGN curve as a reference since it shows the maximum accuracy if $N_{\mathrm{T}} = N_{\mathrm{R}} \to \infty$ [24].

Linear detectors perform bad in this setup: Since the curve of the MF remains almost constant at BER $\approx 20\%$ and the zero forcing performs even worse, both are not shown in the following. At least, MMSE equalization leads to an acceptable BER, but the curve is still separated by a 7 dB gap at $E_{\mathrm{b}}/N_0 = 13$ dB from SD's. In contrast, nonlinear SotA detectors like MOSIC, AMP and SDR technique (see Sec. 3.2 for algorithm details) have a strikingly better accuracy. Whereas AMP runs into an error floor for high SNR since then the message statistics are not Gaussian anymore in finite small-dimensional MIMO systems [24], SDR proves to be a close relaxation by only dropping the non-convex requirement of rank $\left(\mathbf{x}\mathbf{x}^T\right) = 1$ [22].

Notably, our approach CMDNet in its binary version $\mathrm{CMDNet}_{\mathrm{bin}}$ from (3.17) performs even better than the latter, comparable to the best suboptimal approaches in this setup DetNet and OAMPNet. Further, $\mathrm{CMDNet}_{\mathrm{bin}}$ does not run into an error floor in the simulated SNR range like AMP and DetNet. Setting the accuracy in context to complexity (see Tab. 3.2), this is impressive: Note that our approach is similar in asymptotic complexity to the light-weight algorithm AMP with $\mathcal{O}(N_{\mathrm{L}} N_{\mathrm{T}}(N_{\mathrm{R}} + M))$ at inference run time after offline training whereas DetNet and OAMPNet are very complex DNN architectures. In particular, OAMPNet requires one costly matrix inversion per iteration resulting in high $\mathcal{O}(N_{\mathrm{L}} N_{\mathrm{T}}^3)$. In Sec. 3.5.7 and Fig. 3.12, we give a more detailed complexity analysis and comparison illustrating CMD's promising accuracy complexity trade-off more clearly. In contrast, the other DNN-based approach $\mathrm{MMNet}_{\mathrm{iid}}$ with comparable low complexity fails to beat $\mathrm{CMDNet}_{\mathrm{bin}}$ and runs into an early error floor. Since we observed this behavior similar to AMP in all settings and MMNet is actually designed to perform well with fast online training, we omit further results. We conjecture that the denoising layers are insufficient expressive in the interference limited high SNR region with offline training.

Results in a smaller 8×8 MIMO system plotted in Fig. 3.5, show that all soft non-linear approaches except for SDR and MOSIC run into an error floor at lower SNR. Thus, we conjecture that they share the same suboptimality at finite system dimensions. They may rely on the statistics of the interference terms to be Gaussian like AMP which is only approximately true for large system dimensions. Apart from SDR and MOSIC, $\mathrm{CMDNet}_{\mathrm{bin}}$ manages to beat the more expressive and complex DNN models, i.e., DetNet and OAMPNet, and is close in accuracy to SDR for $E_{\mathrm{b}}/N_0 < 10$ dB.

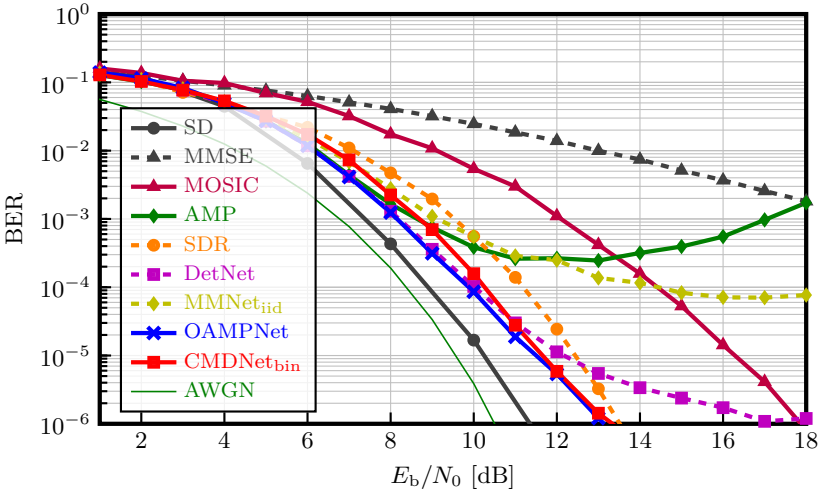

Figure 3.4: BER curves of several detection methods in a 32×32 MIMO system with QPSK modulation. Effective system dimension is 64×64 and for iterative algorithms $N_{it} = N_L = 64$.

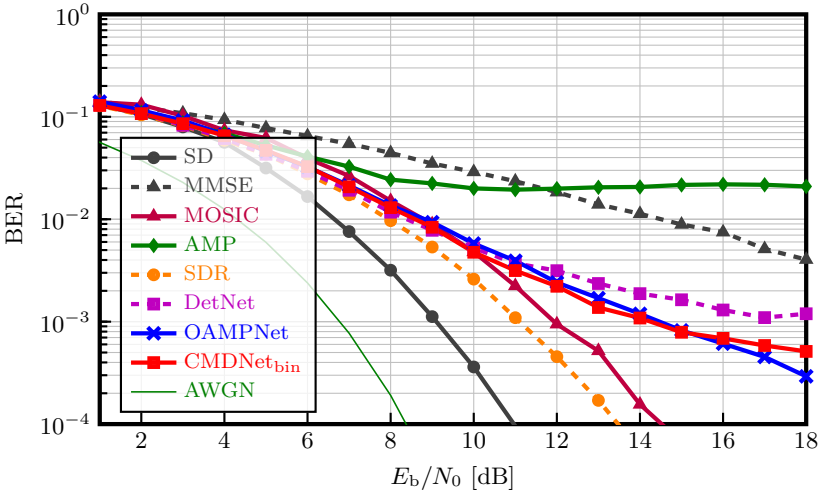

Figure 3.5: BER curves of several detection methods in a 8×8 MIMO system with QPSK modulation. Effective system dimension is 16×16 and for iterative algorithms $N_{it} = N_L = 16$.

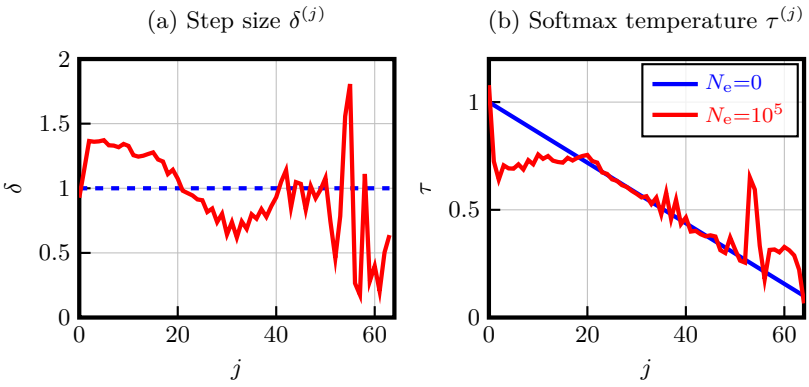

Figure 3.6: Parameters $\boldsymbol{\theta}$ of CMDNet$_{\text{bin}}$ in a 32×32 MIMO system with QPSK modulation. Effective system dimension is 64×64.

3.5.3 Algorithm and Parametrization

To investigate the influence of learning on CMDNet$_{\text{bin}}$ and the values of its parameters $\boldsymbol{\theta}$, we visualize them per layer j in Fig. 3.6 for the 32×32 MIMO system considered before. Basically, we cannot observe any pattern after parameter optimization and interpretation seems very difficult.

Furthermore, we notice from Fig. 3.7 that starting point initialization $\boldsymbol{\theta}_0$ has a large impact on the optimum $\boldsymbol{\theta}_{10^5}$ found by SGD (after $N_{\text{e}} = 10^5$ iterations). If we use a starting point $\boldsymbol{\theta}_{0,\text{splin}}$ with linear decreasing

$$\tau_{0,\text{splin}}^{(j)} = \delta_{0,\text{splin}}^{(j)} = 1 - (1 - 0.01)/N_{\text{it}} \cdot j \tag{3.36}$$

for $j \in [0, N_{\text{it}}]$, a solution $\boldsymbol{\theta}_{10^5,\text{splin}}$ is learned allowing CMDNet to perform better in the low E_{b}/N_0 region from 6 to 10 dB. Notably, CMDNet even reaches the performance of the best suboptimal algorithm considered in this setup OAMPNet. To explain the error floor in the interference limited higher E_{b}/N_0 region in contrast to CMDNet with default training, we conjecture that a higher starting and correlating end step size (see Fig. 3.6) allows CMDNet to leave a local optimum with higher probability and to find a better one. On the contrary, a small step size enforces convergence to a local solution. In the noise limited E_{b}/N_0 region, noise removal is crucial and hence convergence. This means CMDNet can be optimized to different working points and is sensitive to starting point initialization. The result supports our view of a promising accuracy complexity trade-off: Since CMDNet only has a small parameter set, we are able to load the $\boldsymbol{\theta}$ dynamically for each

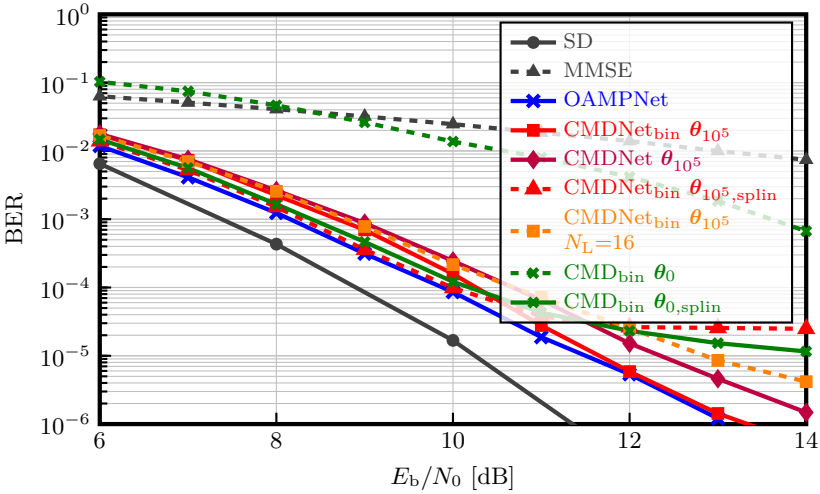

Figure 3.7: BER curves of CMD and CMDNet with different parametrization or algorithmic in a 32×32 MIMO system with QPSK modulation. Effective system dimension is 64×64. Default number of iterations or layers is $N_{it} = N_L = 64$.

E_b/N_0 to achieve the performance of the best suboptimal algorithm in all E_b/N_0 regions.

In particular, we are able to further decrease the number of parameters with negligible performance loss: CMDNet with only $N_L = 16$ layers performs equally well compared to default CMDNet with $N_L = 64$ at low E_b/N_0 and slightly worse at $E_b/N_0 = 12$ dB by 1 dB.

Without unfolding, heuristics for parameter selection are required similar to starting point initialization. The detection accuracy of CMD with such heuristic parameters $\boldsymbol{\theta}_{0,\text{splin}}$ is quite impressive since the BER curve is close to that of learned CMDNet with $\boldsymbol{\theta}_{10^5,\text{splin}}$. Therefore, we are able to use the plain algorithm CMD for detection. We note that this is not true with default parameters $\boldsymbol{\theta}_0$ and that performance can be quite different after optimization ($\boldsymbol{\theta}_{10^5}$).

Finally, we compare the accuracy of algorithm CMDNet$_{bin}$ for the special case of binary RVs from (3.17) with that of the generic multi-class algorithm CMDNet from (3.14) since both are different. From Fig. 3.7, we observe that the performance is very similar and conjecture that CMDNet is capable of achieving the same accuracy if training is parameterized correctly.

3.5.4 Multi-class Detection

So far, only BPSK modulation and hence two classes have been considered. To test multi-class detection with $M = 4$ classes, we show numerical results in a 32×32 MIMO system with 16-QAM modulation being equivalent to a 64×64 4-ASK MIMO system after transformation into the equivalent real-valued problem. Owing to now 3 degrees of freedom in the softmax function and denser symbol packing, we changed our batch size to $N_\mathrm{b} = 1500$ and training SNR to higher $E_\mathrm{b}/N_0 \in [10, 33]$, respectively. Setting the default starting point with $\tau_\mathrm{max} = 2/(M-1) = 2/3$ so that the MAP criterion $\ln p(\tilde{\mathbf{x}}, \mathbf{y})$ becomes convex for a couple of iterations proves to be crucial for successful training of CMDNet with multiple classes. Without training parameter tuning, CMDNet performs even worse than the MMSE detector.

Fig. 3.8 shows BER curves in this system. Clearly, we can now observe a large gap between the BER curve of SD and that of all other suboptimal approaches. Comparing the latter, OAMPNet is superior over the whole SNR region. Observing a maximum 2 dB curve shift, we note that CMDNet is competitive to OAMPNet and SDR at $E_\mathrm{b}/N_0 \in [10, 17]$ and when BER$= [10^{-2}, 10^{-3}]$ which is a typical working point of decoders whereas being much less complex. At higher SNR, an error floor follows. Although using a more expressive DNN model, DetNet now trained for $E_\mathrm{b}/N_0 \in [9, 16]$ fails to beat CMDNet especially in this region.

3.5.5 Massive MIMO System

Investigation in large symmetric MIMO systems reveals the potential and shortcomings of the algorithms. Rather in 5G, massive MIMO systems with $N_\mathrm{R} > N_\mathrm{T}$ are employed [19]. Assuming i.i.d. Gaussian channels, we shortly report the results of a 64×32 MIMO system with QPSK modulation: The BER curves of learning based approaches and SDR almost follow that of SD and thus suggest that they fit perfectly for application in massive MIMO.

However, in practice, channels are spatially correlated at the receiver side due to good spatial resolution of BSs' large arrays compared to the number of scattering clusters [19]. Hence, the results for i.i.d. Gaussian channel statistics $p(\mathbf{H})$ are less meaningful as noted in [34]. As a first and quick attempt towards a realistic channel model which captures its key characteristics, we test performance in the so-called One-ring model $p(\mathbf{H})$ assuming a BS equipped with a uniform linear antenna array [19], [28]. We parameterize the correlation matrices of every column in $\mathbf{H}$ with reasonable values: Assuming an urban cellular network, we set the angular spread to 20° and sample the nominal angle uniformly from $[-60^\circ, 60^\circ]$, i.e., 120° cell sector. Further, we place the antennas at half a wavelength distance.

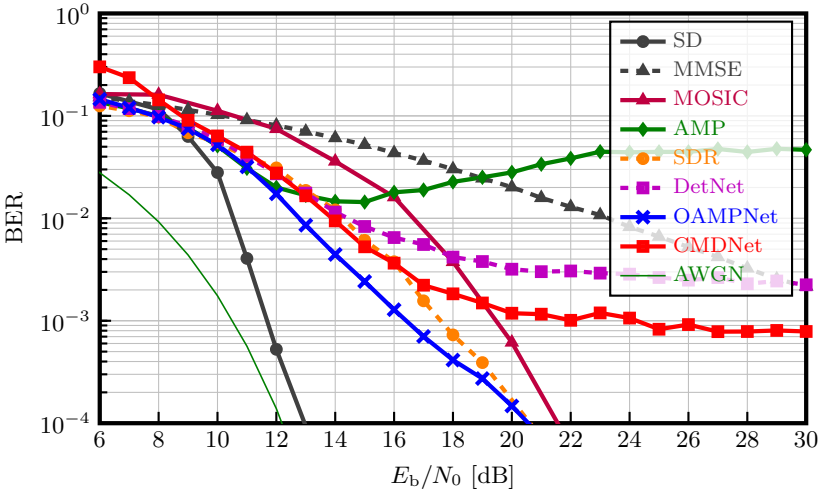

Figure 3.8: BER curves of several detection methods in a 32×32 MIMO system with 16-QAM modulation. Effective system has dimension 64×64 and 4-ASK modulation and for iterative algorithms $N_{\text{it}} = N_{\text{L}} = 64$.

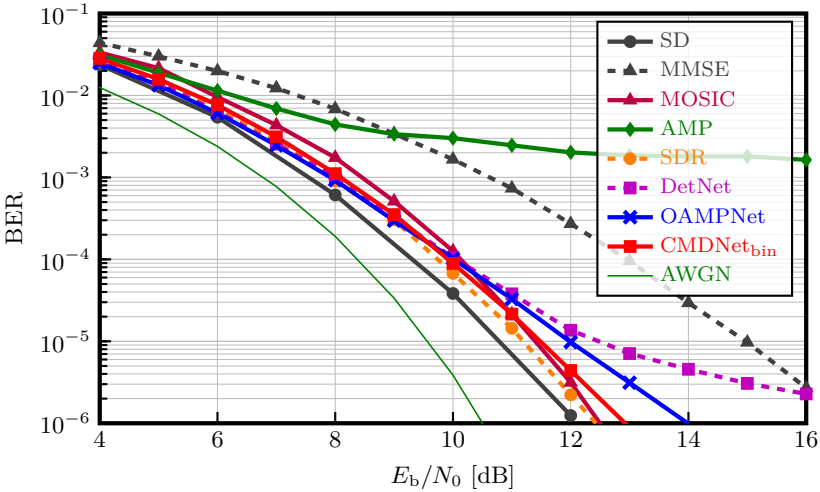

Figure 3.9: BER curves of several detection methods in a correlated 64×32 MIMO system with QPSK modulation. The correlation matrices were generated according to a One-Ring model with 20° angular spread and 120° cell sector. Effective system dimension is 128×64 and for iterative algorithms $N_{\text{it}} = N_{\text{L}} = 64$.

From Fig. 3.9, it becomes evident that the performance loss of learning based approaches compared to SD in such a One-Ring model of dimension 64×32 is similar to the symmetric setting 32×32 in Fig. 3.4. Surprisingly, MOSIC and SDR now prove to be comparable whereas the BER of AMP degrades since the i.i.d. Gaussian channel assumption is not fulfilled anymore. Again, CMDNet outperforms other learning-based approaches DetNet and OAMPNet and performs very close to the best suboptimal algorithm SDR whereas being much less complex (see Tab. 3.2 and Fig. 3.12).

Considering the low complexity, we finally conclude that CMDNet performs surprisingly well in all previous settings. Hence, it proves to be a generic and hence promising detection approach.

3.5.6 Soft Output (Coded MIMO System)

After investigation of detection performance in uncoded systems, we turn to an interleaved and horizontally coded 32×32 MIMO system with Rayleigh block fading reflecting our uplink model. We aim to verify whether not only hard decisions but also soft outputs generated by CMDNet and the soft output version of DetNet have high quality. This is especially important in practice since coding is an essential component besides equalization in today's communication systems. Therefore, we use a 128×64 LDPC code with rate $R_{\mathrm{C}} = 1/2$ from [39] and at receiver side a belief propagation decoder with 10 iterations. The results in terms of Coded Frame Error Rate (CFER) as a function of $E_{\mathrm{b}}/N_0/R_{\mathrm{C}}$ are shown in Fig. 3.10. Owing to overwhelming computational complexity, we refrained from using the MAP solution with coding as a benchmark and instead show uncoded CMDNet and SD curves for reference. Strikingly, CMDNet with coding beats the latter and allows for a coding gain. In contrast, AMP with coding runs into an error floor after 9 dB: The output statistics become unreliable for high SNR in finite dimensional systems [24]. Surprisingly, although being one of the best detection methods in the uncoded setting, DetNet with coding performs close to MMSE equalization with soft outputs and thus worse than expected. Actually, the soft output version of DetNet should deliver accurate probabilities or Log-Likelihood Ratios (LLRs) according to [28] after optimization.

Indeed, we visualize with an exemplary histogram of LLRs that this is not the case. In Fig. 3.11, we show the relative frequencies of LLRs of one symbol x_n in one random channel realization $\mathbf{H}$ for $E_{\mathrm{b}}/N_0 = 10$ dB. First, we note the histograms for $x_n = -1$ and $x_n = 1$ to be symmetric meaning that both algorithms fulfill a basic quality criterion. Furthermore, it can be clearly seen that DetNet mostly provides hard decisions with $\approx 97\%$

LLRs being $-\infty$ and ∞, respectively. Only a few values are close to 0. In contrast, CMDNet provides meaningful soft information resembling a mixture of Gaussians as expected from literature [40] ranging from -30 to 30. These results strongly indicate that the difference of soft output quality originates from different underlying optimization strategies: As pointed out in Section 3.4.2, CMDNet relies on minimization of KL divergence between IO a-posteriori and approximating softmax pmf whereas the one-hot representation in DetNet is optimized w.r.t. MSE. We conclude that our approach yields a better optimization strategy.

3.5.7 Complexity Analysis

Since complexity is the main driver for development of suboptimal algorithms like CMD instead of relying on MAP detection, we complete our numerical study by relating detection accuracy to results on the computational complexity given in Tab. 3.2. With regard to CMD and CMDNet applied in our guiding example (3.1), the iterative asymptotic complexity of $\mathcal{O}(N_\mathrm{T}(2N_\mathrm{R}+4M))$ or $\mathcal{O}(2N_\mathrm{T}N_\mathrm{R})$ for binary RVs is dominated by the matrix vector multiplications in $\mathbf{H}^T\mathbf{H}\tilde{\mathbf{x}}$, i.e., CMD scales linearly with the input and output dimension as well as the number of classes. Clearly, CMD and CMDNet have very low complexity comparable to AMP and MMNet but with remarkable higher detection rate (see, e.g., Fig. 3.4). In most analyzed scenarios, the accuracy is even higher than DetNet's as well as OAMPNet's and on par with SDR's.

Besides qualitative $\mathcal{O}(\cdot)$ analysis, we capture complexity quantitatively by counting the number of Multiplicative OPerations (MOPs) for one iteration and channel realization being the most common and costly floating point operations. In Fig. 3.12, we show the respective bar chart assuming a realistic low-complexity implementation in a 32×32 with QPSK ($M = 2$) and $N_\mathrm{L} = 16$ and worst-case complexity implementation with 16-QAM modulation ($M = 4$) and $N_\mathrm{L} = 64$, respectively. For BPSK and the lower bar of MMSE equalization, we assumed Gaussian elimination to solve the linear equation system and, for higher order QAM and the higher bar, LU decomposition. We estimate the upper bound on SDR MOP count by unadapted $\mathcal{O}(\max(N_\mathrm{R}, N_\mathrm{T})^4 N_\mathrm{T}^{1/2} \log(1/\epsilon))$ and the lower bound on MOPs to account for half of the FLOPS from [28] with inaccurate $\epsilon = 0.1$. The expected number of visiting nodes $\mathcal{O}(M^{\gamma N_\mathrm{T}})$ of the SD is SNR dependent with $\gamma \in (0, 1]$ and was extracted from [20] for $E_\mathrm{b}/N_0 = 10$ dB.

Apparently, only the very basic MF beats CMD and CMDNet in complexity at considerably worse detection accuracy. Approaches with comparable accuracy like DetNet, OAMPNet and SDR are 10-100 times more complex

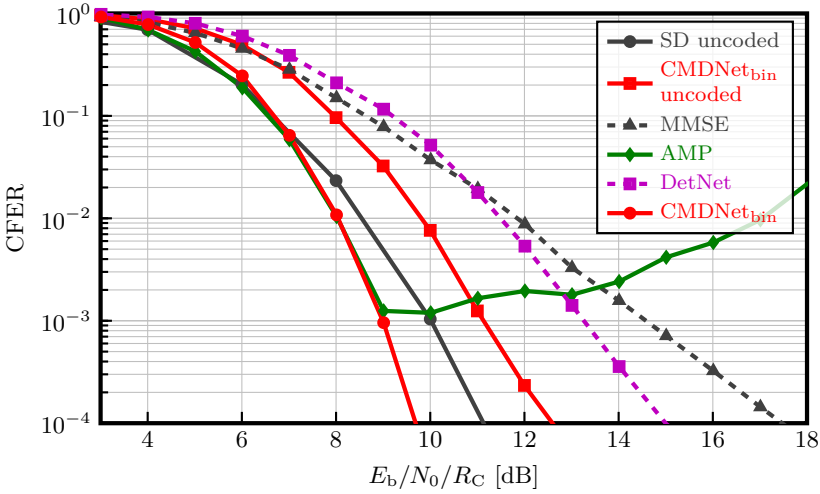

Figure 3.10: CFER curves of a horizontally coded 32×32 MIMO system with QPSK modulation. A 128×64 LDPC code with belief propagation decoder was used. Effective system dimension is 64×64 and for iterative soft detectors $N_{\text{it}} = N_{\text{L}} = 64$.

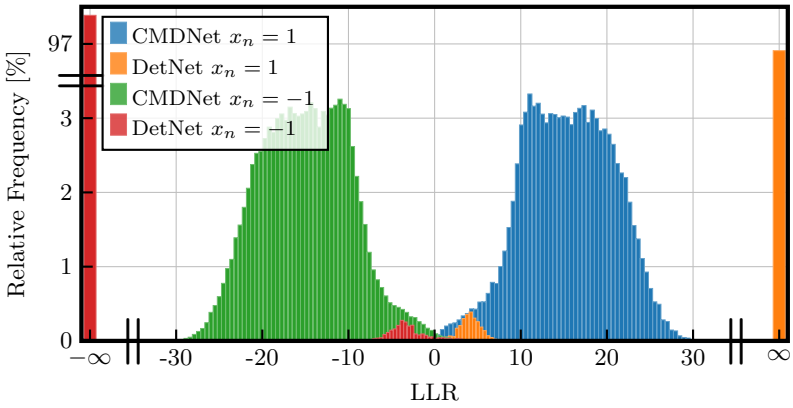

Figure 3.11: Exemplary histogram showing the relative frequencies of LLRs of one symbol x_n in one random channel realization $\mathbf{H}$ at $E_{\text{b}}/N_0 = 10$ dB.

w.r.t. MOPs. We conclude that CMDNet offers an excellent accuracy complexity trade-off and note that AMP, MMNet, DetNet and CMDNet further come with the benefit of already delivering soft outputs.

As a final remark, note that complexity analysis depends on the assumptions made: If we, e.g., assume long channel coherence time intervals, MMSE and MOSIC are able to reuse its computations with only one matrix vector multiplication remaining for any further detection inside the interval effectively decreasing complexity. For the same reason, online learning approaches do not require further training inside the interval and could be feasible. Comparing training cost of all unfolding algorithms in Tab. 3.3, we note that N_b and N_e lie in the same range. Hence, the forward pass of backpropagation in SGD and respectively run time complexity from Fig. 3.12 as well as the number of parameters $|\boldsymbol{\theta}|$ to be optimized dominate training complexity. OAMPNet fails in the former and DetNet in the latter category with $|\boldsymbol{\theta}| \in [10^5, 10^7]$ assuming $N_\mathrm{L} = \{16, 64\}$ and {QPSK, 16-QAM}. In contrast, CMDNet with low runtime complexity and only $|\boldsymbol{\theta}| = \{33, 129\}$ may be a promising online training approach similar to MMNet [34].

3.6 Conclusion

In this article, we introduced the so-called continuous relaxation of discrete RVs to the MAP detection problem. Allowing to replace exhaustive search by continuous optimization, we defined our classification approach Concrete MAP Detection (CMD), e.g., based on gradient descent. By unfolding CMD into a DNN CMDNet, we further were able to optimize its low number of parameters and hence to improve detection accuracy while limiting it to low complexity. As a side effect, the resulting structure has the potential to allow for fast online training. Using the example of MIMO detection, simulations reveal CMDNet to be a generic detection method competitive to SotA outperforming it in terms of complexity and other recently proposed ML-based approaches DetNet and MMNet in every considered scenario. Notably, we selected an optimization criterion grounded in information theory, i.e., cross-entropy, and showed that it aims at learning an approximation of the individual optimal detector. By simulations in coded systems, we demonstrated its ability to provide reliable soft outputs as opposed to [28], being a requirement for soft decoding, a crucial component in today's communication systems.

All these findings prove CMDNet to be a promising detection approach for application in future massive MIMO systems. Further research is required to evaluate its potential for fast online learning and to demonstrate its applicability to non-linear scenarios of other research domains.

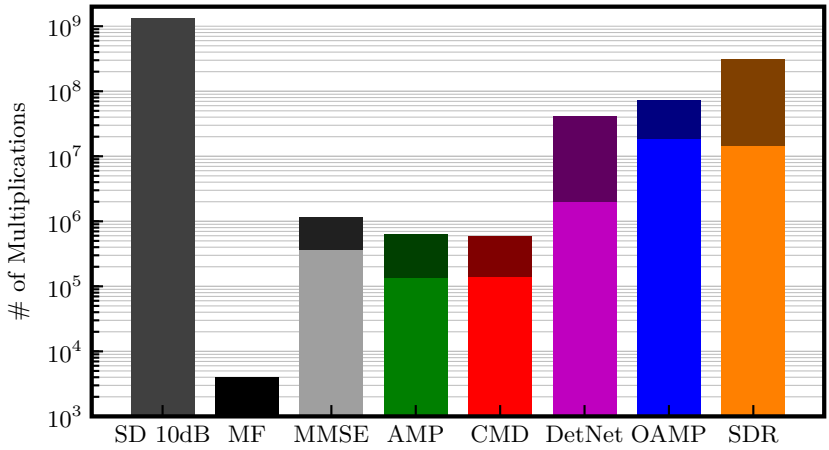

Figure 3.12: Complexity of detection algorithms in terms of number of multiplicative operations in a 32×32 / 64×64 MIMO system: Light colored bars indicate a realistic low-complexity implementation with BPSK and dark-colored bars the worst-case complexity with 16-QAM modulation.

Table 3.3: Training complexity

Algorithm	$\approx N_{\mathrm{b}}$	$\approx N_{\mathrm{e}}$	$\lvert\boldsymbol{\theta}\rvert$
DetNet	2000	10^5	$N_{\mathrm{L}}[(\{2,4\}M + \{6,20\})N_{\mathrm{T}}^2$
{QPSK, 16-QAM}	-5000		$+(M + \{3,6\})N_{\mathrm{T}} + 2]$
OAMPNet	1000	10^4-10^5	$2N_{\mathrm{L}}$
MMNet {iid, full}	500	10^4-10^5	$\{2N_{\mathrm{L}}, N_{\mathrm{L}}N_{\mathrm{T}}(N_{\mathrm{R}}+1)\}$
CMDNet	500	10^4-10^5	$2N_{\mathrm{L}} + 1$

3.7 References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[2] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967. DOI: 10.1109/TIT.1967.1054010.

[3] D. D. Lin and T. J. Lim, "A Variational Inference Framework for Soft-In Soft-Out Detection in Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2345–2364, May 2009. DOI: 10.1109/TIT.2009.2016054.

[4] E. Riegler, G. E. Kirkelund, C. N. Manchon, M. Badiu, and B. H. Fleury, "Merging Belief Propagation and the Mean Field Approximation: A Free Energy Approach," *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 588–602, Jan. 2013. DOI: 10.1109/TIT.2012.2218573.

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989. DOI: 10.1016/0893-6080(89)90020-8.

[6] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018. DOI: 10.1109/TCCN.2018.2881442.

[7] M. Abadi *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, 2015.

[8] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, Ft. Lauderdale, FL, USA: JMLR Workshop and Conference Proceedings, Jun. 2011, pp. 315–323.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[10] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in *25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, Providence, RI, USA, Jun. 2012, pp. 3642–3649. DOI: 10.1109/CVPR.2012.6248110.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. DOI: 10.1038/nature16961.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *27th Conference on Advances in Neural Information Processing Systems (NIPS 2014)*, Montreal, Canada, 2014, pp. 2672–2680.

[13] N. Farsad and A. Goldsmith, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018. DOI: 10.1109/TSP.2018.2868322.

[14] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-End Deep Learning of Optical Fiber Communications," *IEEE/OSA Journal of Lightwave Technology*, vol. 36, no. 20, pp. 4843–4855, Oct. 2018. DOI: 10.1109/JLT.2018.2865109.

[15] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback Codes via Deep Learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 194–206, May 2020. DOI: 10.1109/JSAIT.2020.2986752.

[16] 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, Dec. 2017. DOI: 10.1109/TCCN.2017.2758370.

[17] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019. DOI: 10.1109/JSAC.2019.2933891.

[18] A. Caciularu and D. Burshtein, "Unsupervised Linear and Nonlinear Channel Equalization and Decoding Using Variational Autoencoders," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 1003–1018, Sep. 2020. DOI: 10.1109/TCCN.2020.2990773.

[19] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, Nov. 2017. DOI: 10.1561/2000000093.

[20] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005. DOI: 10.1109/TSP.2005.843746.

[21] D. Wübben, R. Böhnke, V. Kühn, and K.-D. Kammeyer, "MMSE Extension of V-BLAST based on Sorted QR Decomposition," in *58th IEEE Vehicular Technology Conference (VTC 2003-Fall)*, Orlando, USA, Oct. 2003, pp. 508–512. DOI: 10.1109/VETECF.2003.1285069.

[22] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite Relaxation of Quadratic Optimization Problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, May 2010. DOI: 10.1109/MSP.2010.936019.

[23] O. Simeone, "A Brief Introduction to Machine Learning for Engineers," *Foundations and Trends® in Signal Processing*, vol. 12, no. 3-4, pp. 200–431, Aug. 2018. DOI: 10.1561/2000000102.

[24] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of Large MIMO Detection via Approximate Message Passing," in *IEEE International Symposium on Information Theory (ISIT 2015)*, Hong Kong, Jun. 2015, pp. 1227–1231. DOI: 10.1109/ISIT.2015.7282651.

[25] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, Mar. 2021. DOI: 10.1109/MSP.2020.3016905.

[26] A. Balatsoukas-Stimming and C. Studer, "Deep Unfolding for Communications Systems: A Survey and Some New Directions," in *IEEE International Workshop on Signal Processing Systems (SiPS 2019)*, Nanjing, China, Oct. 2019, pp. 266–271. DOI: 10.1109/SiPS47522.2019.9020494.

[27] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO Detection," in *18th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2017)*, Sapporo, Japan, Jul. 2017, pp. 1–5. DOI: 10.1109/SPAWC.2017.8227772.

[28] N. Samuel, T. Diskin, and A. Wiesel, "Learning to Detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, May 2019. DOI: 10.1109/TSP.2019.2899805.

[29] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Annual Allerton Conference on Communication, Control, and Computing (Allerton 2016)*, vol. 54, Monticello, IL, USA, Sep. 2016, pp. 341–346. DOI: 10.1109/ALLERTON.2016.7852251.

[30] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep Learning Methods for Improved Decoding of Linear Codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, Feb. 2018. DOI: 10.1109/JSTSP.2017.2788405.

[31] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, "On deep learning-based channel decoding," in *51st Annual Conference on Information Sciences and Systems (CISS 2017)*, Baltimore, MD, USA, Mar. 2017, pp. 1–6. DOI: 10.1109/CISS.2017.7926071.

[32] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE Channel Estimator," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2905–2917, Jun. 2018. DOI: 10.1109/TSP.2018.2799164.

[33] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A Model-Driven Deep Learning Network for MIMO Detection," in *6th IEEE Global Conference on Signal and Information Processing (GlobalSIP 2018)*, Anaheim, CA, USA, Nov. 2018, pp. 584–588. DOI: 10.1109/GlobalSIP.2018.8646357.

[34] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive Neural Signal Detection for Massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020. DOI: 10.1109/TWC.2020.2996144.

[35] C. J. Maddison, A. Mnih, and Y. W. Teh, "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables," in *5th International Conference on Learning Representations (ICLR 2017)*, Toulan, France, Apr. 2017, pp. 1–20. DOI: 10.48550/arXiv.1611.00712.

[36] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," in *5th International Conference on Learning Representations (ICLR 2017)*, Toulan, France, Apr. 2017, pp. 1–13. DOI: 10.48550/arXiv.1611.01144.

[37] E. Beck, C. Bockelmann, and A. Dekorsy, "Concrete MAP Detection: A Machine Learning Inspired Relaxation," in *24th International ITG Workshop on Smart Antennas (WSA 2020)*, Hamburg, Germany: VDE VERLAG, Feb. 2020, pp. 1–5.

[38] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," in *31st Conference on Advances in Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, Dec. 2017, pp. 4148–4158.

[39] M. Helmling, S. Scholl, F. Gensheimer, T. Dietz, K. Kraft, S. Ruzika, and N. Wehn. "Database of Channel Codes and ML Simulation Results." (2019), [Online]. Available: https:/www.uni-kl.de/channel-codes.

[40] M. Cirkic, D. Persson, J.-Å. Larsson, and E. G. Larsson, "Approximating the LLR Distribution for a Class of Soft-Output MIMO Detectors," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6421–6434, Dec. 2012. DOI: 10.1109/TSP.2012.2217336.

Part II

Semantic Communications

Chapter 4

Publication 2 – Semantic Information Recovery in Wireless Networks

E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic Information Recovery in Wireless Networks," *Sensors*, vol. 23, no. 14, p. 6347, Jul. 2023. DOI: 10.3390/s23146347

The simulation source code is available in [Bec24]. Further analyses and additional details for this publication are provided in Appendix B.

4.1 Abstract

Motivated by the recent success of Machine Learning (ML) tools in wireless communications, the idea of semantic communication by Weaver from 1949 has gained attention. It breaks with Shannon's classic design paradigm by aiming to transmit the meaning of a message, i.e., semantics, rather than its exact version and, thus, enables savings in information rate. In this work, we extend the fundamental approach from Basu et al. for modeling semantics to the complete communications Markov chain. Thus, we model semantics by means of hidden random variables and define the semantic communication task as the data-reduced and reliable transmission of messages over a communication channel such that semantics is best preserved. We consider this task as an end-to-end information bottleneck problem, enabling compression

while preserving relevant information. As a solution approach, we propose the ML-based semantic communication system SINFONY and use it for a distributed multipoint scenario; SINFONY communicates the meaning behind multiple messages that are observed at different senders to a single receiver for semantic recovery. We analyze SINFONY by processing images as message examples. Numerical results reveal a tremendous rate-normalized SNR shift up to 20 dB compared to classically designed communication systems.

Keywords

Semantic communication; wireless communications; wireless networks; InfoMax; information bottleneck; machine learning; task-oriented communication; goal-oriented communication

4.2 Introduction

When Shannon laid the theoretical foundation of the research area of communications engineering back in 1948, he deliberately excluded semantic aspects from the system design [1], [2]. In fact, the idea of addressing semantics in communications arose shortly after Shannon's work in [2], but it remained largely unexplored. Since then, the design focus of communication systems has been on digital error-free point-to-point symbol transmission.

Today, the systems already operate close to the Shannon limit calling for a paradigm shift towards including semantic content of messages in the system design. For example, the data traffic growth still continues with the emergence of the Internet-of-Everything including, e.g., autonomous driving and virtual reality, and cannot be managed by semantics-agnostic communication as it limits the achievable efficiency in terms of bandwidth, power, latency, and complexity trade-offs [3]. Other notable examples include wireless sensor networks, broadcast scenarios, and non-ergodic channels where separation of source and channel coding according to Shannon's digital design paradigm is generally suboptimal [4], [5].

Owing to the great success of Artificial Intelligence (AI) and, in particular, its subdomain Machine Learning (ML), ML tools have been recently investigated for wireless communications and has shown promising application for improving the performance complexity trade-off [6]–[8]. Now, ML with its ability to extract features appears to be a proper means to realize a semantic design. Further, we note that the latter design is supported and

possibly enabled by the 6G vision of integrating AI and ML on all layers of the communications system design, i.e., by a ML-native air interface.

Motivated by these new ML tools, and driven by the unprecedented needs of the next wireless communication standard, 6G, in terms of data rate, latency, and power, the idea of semantic communication has received considerable attention [2], [9]–[13]. It breaks with the existing classic design paradigms by including semantics in the design of the wireless transmission. The goal of such a transmission is, therefore, to deliver the required data from which the highest levels of quality of information may be derived, as perceived by the application and/or the user. More precisely, semantic communication aims to transmit the meaning of a message rather than its exact version and hence enables compression and coding to the actual semantic content. Thus, savings in bandwidth, power, and complexity are expected.

In the following, we first summarize in Section 4.3 related work on semantic communication and justify our main contributions in Section 4.4. In Section 4.5.1, we reinterpret Weaver's philosophical considerations paving the way for our proposed theoretical framework in Section 4.5. Finally, in Sections 4.6 and 4.7, we provide one numerical example of semantic communication, i.e., SINFONY, and summarize the main results, respectively.

4.3 Related Work

The notion of semantic communication traces back to Weaver [2] who reviewed Shannon's information theory [1] in 1949 and amended considerations with regard to semantic content of messages. Often quoted is his statement that *"there **seem** to be [communication] problems at three levels"* [2]:

A. How accurately can the symbols of communication be transmitted? (The technical problem.)

B. How precisely do the transmitted symbols convey the desired meaning? (The semantic problem.)

C. How effectively does the received meaning affect conduct in the desired way? (The effectiveness problem.)

Since then semantic communication was mainly investigated from a philosophical point of view, see, e.g., [14], [15].

The generic model of Weaver was revisited by Bao, Basu et al. in [16], [17] where the authors define semantic information source and semantic channel. In particular, the authors consider a semantic source that *"observes the world*

and generates meaningful messages characterizing these observations" [17]. The source is equivalent to conclusions, i.e., "models" of the world, that are unequivocally drawn following a set of known inference rules based on observation of messages. In [16], the authors consider joint semantic compression and channel coding at Level B with the classic transmission system, i.e., Level A, as the (semantic) channel. In contrast, [17] only deals with semantic compression and uses a different definition of the semantic channel (which we will make use of in this article): It is equal to the entailment relations between "models" and "messages". By this means, the authors are able to derive semantic counterparts of the source and channel coding theorems. However, as the authors admit, these theorems do not tell how to develop optimal coding algorithms and the assumption of a logic-based model-theoretical description leads to *"many non-trivial simplifications"* [16].

In [18], the authors follow a different approach in the context of Natural Language Processing (NLP). They define semantic similarity as a semantic error measure using taxonomies, i.e., human knowledge graphs, to quantify the distance between the meanings of two words. Based on this metric, communication of a finite set of words is modeled as a Bayesian game from game theory and optimized for improved semantic transmission over a binary symmetric channel.

Recently, drawing inspiration from Weaver, Bao, Basu et al. [2], [16], [17] and enabled by the rise of ML in communications research, Deep Neural Network (DNN)-based NLP techniques, i.e., transformer networks, were introduced in AutoEncoders (AEs) for the task of text transmission [19]–[21]. The aim of these techniques is to learn compressed hidden representations of the semantic content of sentences to improve communication efficiency, but the exact recovery of the source (text) is the main objective. The approach improves performance in semantic metrics, especially at low SNR compared to classical digital transmissions. It has been adapted to numerous other problems, e.g., speech transmission [22], [23] and multi-user transmission with multi-modal data [24]. Even knowledge graphs, i.e., a prior knowledge base, were incorporated into the transformer-based AE design to improve inference at the receiver side and, thus, text recovery [25].

Not considering Weaver's idea of semantic communication in particular, the authors in [26] show, for the first time, that task-oriented communications (Level C) for edge cloud transmission can be mathematically formulated as an Information Bottleneck (IB) optimization problem. Moreover, for solving the IB problem, they introduce a DNN-based approximation and show its applicability for the specific task of edge cloud transmission. The terminus *"semantic information"* is only mentioned once in [26] referring

to Joint Source-Channel Coding (JSCC) of text from [19] using recurrent neural networks. In [19], the authors observe that sentences that express the same idea have embeddings that are close together in Hamming distance. But they use cross-entropy between words and estimated words as the loss function and use the word error rate as the performance measure, which both do not reflect if two sentences have the same meaning but rather that both are exactly the same.

As a result, semantic communication is still a nascent field; it still remains unclear what this term exactly means [27] and, in particular, its distinction from JSCC [19], [28]. As a result, many survey papers aim to provide an interpretation, see, e.g., [9]–[13]. We will revisit this issue in Section 4.5.

4.4 Main Contributions

The main contributions of this article are:

- Motivated by the approach of Bao, Basu et al. [16], [17], we adopt the terminus of a semantic source. Inspired by Weaver's notion, we bring it to the context of communications by considering the complete Markov chain, including semantic source, communications source, transmit signal, communication channel, and received signal in contrast to both [16], [17]. Further, we also extend beyond the example of deterministic entailment relations between "models" and "messages" based on propositional logic in [16], [17] to probabilistic semantic channels.

- We define the task of semantic communication in the sense that we perform data compression, coding, and transmission of messages observed such that the semantic Random Variable (RV) at a recipient is best preserved. Basically, we implement joint source-channel coding of messages conveying the semantic RV, but not differentiating between Levels A and B. We formulate the semantic communication design either as an information maximization or as an Information Bottleneck (IB) optimization problem [29]–[31].

  - Although the approach pursued here again leads to an IB problem as in [26], our article introduces a new classification and perspective of semantic communication and different ML-based solution approaches. Different from [26], we solve the IB problem maximizing the mutual information for a fixed encoder output dimension that bounds the information rate.

  - The publication presented here differs also both in the interpretation of what is meant by semantic information and in the objective of recovering this semantic information from approaches to semantic communication presented in the literature like, e.g., [21], [32].

- Finally, we propose the ML-based semantic communication system SINFONY for a distributed multipoint scenario in contrast to [26]: SINFONY communicates the meaning behind multiple messages that are observed at different senders to a single receiver for semantic recovery. Compared to the distributed scenario in [33], [34], we include the communication channel.

- We analyze SINFONY by processing images as an example of messages. Notably, numerical results reveal a tremendous rate-normalized SNR shift up to 20 dB compared to classically designed communication systems.

4.5 A Framework for Semantics

4.5.1 Philosophical Considerations

Despite the much-renewed interest, research on semantic communication is still in its infancy and recent work reveals a differing understanding of the word *semantics*. In this work, we contribute our interpretation. To motivate it, we shortly revisit the research birth hour of communications from a philosophical point of view; its theoretical foundation was laid by Shannon in his landmark paper [1] in 1948.

He stated that *"Frequently the messages have meaning; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic aspects of communication are irrelevant to the engineering problem".* In fact, this viewpoint abstracts all kinds of information one may transmit, e.g., oral and written speech, sensor data, etc., and also lays the foundation for the research area of Shannon information theory. Thus, it found its way into many other research areas where data or information are processed, including Artificial Intelligence (AI) and especially its subdomain Machine Learning (ML).

Weaver saw this broad applicability of Shannon's theory back in 1949. In their comprehensive review of [1], he first states that *"there **seem** to be [communication] problems at three levels"* [2] already mentioned in Section 4.3. These three levels are quoted in recent works, where Level C is oftentimes referred to as goal-oriented communication instead [10].

But we note that, in his concluding section, he then questions this segmentation. He argues for the generality of the theory at Level A for all levels and ***"that the interrelation of the three levels is so considerable that one's final conclusion may be that the separation into the three levels is really artificial and undesirable"***.

It is important to emphasize that the separation is rather arbitrary. We agree with Weaver's statement because the most important point that is also the focus herein is the definition of the term semantics, e.g., by Basu et al. [16], [17]. Note that the entropy of the semantics is less than or equal to the entropy of the messages. Consequently, we can save information rate by introducing meaning or context. In fact, we are able to add arbitrarily many levels of semantic details to the communication problem and optimize communications for a specific semantic background, e.g., an application or a human.

4.5.2 Semantic System Model

Semantic Source and Channel

Now, we will define our information-theoretic system model of semantic communication. Figure 4.1 shows the schematic of our model. We assume the existence of a semantic source, described as a hidden target multivariate Random Variable (RV) $\mathbf{z} \in \mathcal{M}_z^{N_z \times 1}$ from a domain $\mathcal{M}_z$ of dimension N_z distributed according to a probability density function (pdf) or probability mass function (pmf) $p(\mathbf{z})$. To simplify the discussion, we assume it to be discrete and memoryless. For the remainder of the article, note that the domain of all RVs $\mathcal{M}$ may be either discrete or continuous. Further, we note that the definition of entropy for discrete and continuous RVs differs. For example, the differential entropy of continuous RVs may be negative whereas the entropy of discrete RVs is always positive [35]. Without loss of generality, we will thus assume all RVs either to be discrete or to be continuous. In this work, we avoid notational clutter by using the expected value operator; replacing the integral by summation over discrete RVs, the equations are also valid for discrete RVs and vice versa.

Our approach is similar to that of [16], [17]. In [16], [17], the semantic source is described by "models of the world". (Note that, in [17], the semantic information source is defined as a tuple $(\mathbf{z}, \mathbf{s}, p(\mathbf{z}, \mathbf{s}), L)$. In this original notation, $\mathbf{z}$ is the model, $\mathbf{s}$ the message, $p(\mathbf{z}, \mathbf{s})$ the joint distribution of $\mathbf{z}$ and $\mathbf{s}$, and L is the deterministic formal language.) In [17], a semantic channel then generates messages through entailment relations between "models" and "messages". We will call these "messages" source signal and define it to be a RV $\mathbf{s} \in \mathcal{M}_s^{N_s \times 1}$ as it is usually observed and enters the communication

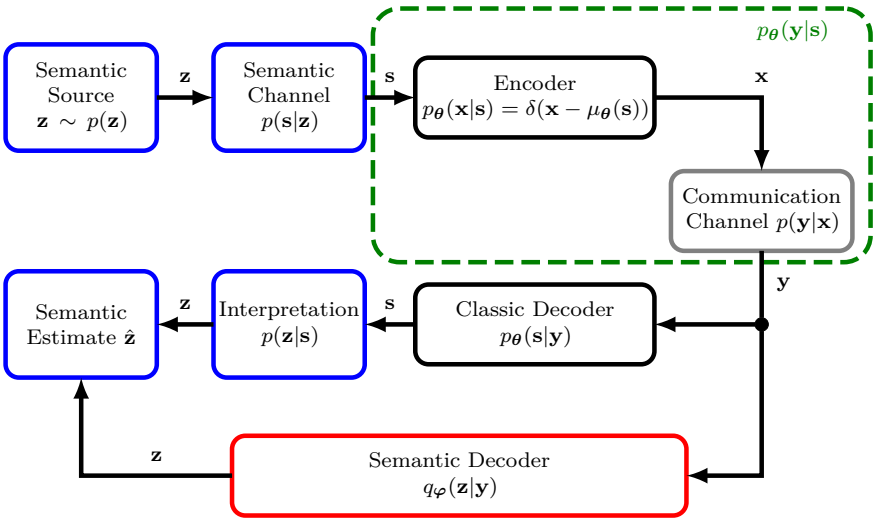

Figure 4.1: Block diagram of the considered semantic system model.

system. In the classic Shannon design, the aim is to reconstruct the source $\mathbf{s}$ as accurately as possible at the receiver side. Further, we note that the authors in [17] considered the example of a semantic channel with deterministic entailment relations between $\mathbf{z}$ and $\mathbf{s}$ based on propositional logic. In this article, we go beyond this assumption and consider probabilistic semantic channels modeled by distribution $p(\mathbf{s}|\mathbf{z})$ that include the entailment in [17] as special cases, i.e., $p(\mathbf{s}|\mathbf{z}) = \delta(\mathbf{s} - f(\mathbf{z}))$ where $\delta(\cdot)$ is the Dirac delta function and $f(\cdot)$ is any generic function. Our viewpoint is motivated by the recent success of pattern recognition tools that advanced the field of AI in the 2010s and may be used to extract semantics [7].

Our approach also extends models as in [21]. There, the authors design a semantic communication system for the transmission of written language/text similar to [19] using transformer networks. In contrast to our work, [21] does not define meaning as RV $\mathbf{z}$. The objective in [21] is to reconstruct $\mathbf{s}$ (sentences) as well as possible, rather than the meaning (RV $\mathbf{z}$) conveyed in $\mathbf{s}$. Optimization is completed with regard to a loss function consisting of two parts, cross-entropy between language input $\mathbf{s}$ and output estimate $\hat{\mathbf{s}}$, as well as a scaled mutual information term between transmit signal $\mathbf{x}$ and receive signal $\mathbf{y}$. After optimization, the authors measure semantic performance by some semantic metric $\mathcal{L}(\mathbf{s}, \hat{\mathbf{s}})$.

We now provide an example to explain what we understand under a semantic source $\mathbf{z}$ and channel $p(\mathbf{s}|\mathbf{z})$. Let us imagine a biologist who has

an image of a tree. The biologist wants to know what kind of tree it is by interpreting the observed data (image). In this case, the semantic source $\mathbf{z}$ is a multivariate RV composed of a categorical RV with M tree classes. For any realization (sample value) $\mathbf{z}_i$ of the semantic source, the semantic channel $p(\mathbf{s}|\mathbf{z})$ then outputs with some probability one image $\mathbf{s}_i$ of a tree conveying characteristics of $\mathbf{z}$, i.e., its meaning. Note that the underlying meaning of the same sensed data (message) can be different for other recipients, e.g., humans or tasks/applications, i.e., in other semantic contexts. Imagine a child, i.e., a person with different characteristics (personality, expertise, knowledge, goals, and intentions) than the biologist, who is only interested if he/she can climb up this tree or whether the tree provides shade. Thus, we include the characteristics of the sender and receiver in the RV $\mathbf{z}$ and consider it directly in compression and encoding.

Compared to [16], we therefore argue that we also include level C by semantic source and channel since context can be included on increasing layers of complexity. First, a RV $\mathbf{z}_1$ might capture the interpretation, like the classification of images or sensor data. Moving beyond the first semantic layer, then a RV $\mathbf{z}_2$ might expand this towards a more general goal, like keeping a constant temperature in power plant control. In fact, we can add or remove context, i.e., semantics and goals, arbitrarily often according to the human or application behind, and we can optimize the overall (communication) system with regard to $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_i$, respectively.

As a last remark, we note that we basically defined probabilistic semantic relationships, and it remains the question of how exactly they might look. In our example, the meaning of the images needs to be labeled into real-world data pairs $\{\mathbf{s}_i, \mathbf{z}_i\}$ by experts/humans, since image recognition lacks precise mathematical models. This is also true for NLP [21]; how can we measure if two sentences have the same meaning, i.e., how does the semantic space look like? In contrast, in [17], the authors are able to solve their well-defined technical problem (motion detection) by a model-driven approach. We can thus distinguish between model- and data-driven semantics, which both can be handled within Shannon's information theory.

Semantic Channel Encoding

After the semantic source and channel in Figure 4.1, we extend upon [16] by differentiating between "message"/source signal $\mathbf{s}$ and transmit signal $\mathbf{x} \in \mathcal{M}_x^{N_{\mathrm{Tx}} \times 1}$. Our challenge is to encode the source signal $\mathbf{s}$ onto the transmit signal vector $\mathbf{x}$ for reliable semantic communication through the physical communication channel $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y} \in \mathcal{M}_y^{N_{\mathrm{Rx}} \times 1}$ is the received signal vector. We assume the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ to be parametrized by a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}} \times 1}$. Note that $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ is probabilistic here, but

assumed to be deterministic in communications with $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) = \delta(\mathbf{x} - \mu_{\boldsymbol{\theta}}(\mathbf{s}))$ and encoder function $\mu_{\boldsymbol{\theta}}(\mathbf{s})$.

In summary, in contrast to both [16], [17], we consider the complete Markov chain $\mathbf{z} \leftrightarrow \mathbf{s} \leftrightarrow \mathbf{x} \leftrightarrow \mathbf{y}$ including semantic source $\mathbf{z}$, communications source $\mathbf{s}$, transmit signal $\mathbf{x}$ and receive signal $\mathbf{y}$. By this means, we distinguish from [17] which only deals with semantic compression, and [16] which is about joint semantic compression and channel coding (Level B). In [16], the authors consider the classic transmission system (Level A) as the (semantic) channel (not to be confused with the definition of the semantic channel in [17] which we make use of in this publication).

At the receiver side, one approach is maximum a posteriori decoding with regard to RV $\mathbf{s}$ that uses the posterior $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})$, being deduced from prior $p(\mathbf{s})$ and likelihood $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ by application of Bayes law. Based on the estimate of $\mathbf{s}$, then the receiver interprets the actual semantic content $\mathbf{z}$ by $p(\mathbf{z}|\mathbf{s})$.

Another approach we propose is to include the semantic hidden target RV $\mathbf{z}$ into the design by processing $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$. If the calculation of the posterior is intractable, we can replace $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$ by the approximation $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$, i.e., the semantic decoder, with parameters $\boldsymbol{\varphi} \in \mathbb{R}^{N_{\boldsymbol{\varphi}} \times 1}$. We expect the following benefit: We assume the entropy $\mathcal{H}(\mathbf{z}) = \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})}[-\ln p(\mathbf{z})]$ of the semantic RV $\mathbf{z}$, i.e., the actual semantic uncertainty or information content, to be less or equal to the entropy $\mathcal{H}(\mathbf{s})$ of the source $\mathbf{s}$, i.e., $\mathcal{H}(\mathbf{z}) \leq \mathcal{H}(\mathbf{s})$. There, $\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x})}[f(\mathbf{x})]$ denotes the expected value of $f(\mathbf{x})$ with regard to both discrete or continuous RVs $\mathbf{x}$. Consequently, since we would like to preserve the relevant, i.e., semantic, RV $\mathbf{z}$ rather than $\mathbf{s}$, we can compress more s.t. preserving $\mathbf{z}$ conveyed in $\mathbf{s}$. Note that in semantic communication the relevant variable is $\mathbf{z}$, not $\mathbf{s}$. Thus, processing $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})$ without taking $\mathbf{z}$ into consideration resembles the classical approach. Instead of using (and transmitting) $\mathbf{s}$ for inference of $\mathbf{z}$, we now want to find a compressed representation $\mathbf{y}$ of $\mathbf{s}$ containing the relevant information about $\mathbf{z}$.

4.5.3 Semantic Communication Design via InfoMax Principle

After explaining the system model and the basic components, we are able to approach a semantic communication system design. We first define an optimization problem to obtain the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ following the Information Maximization (InfoMax) principle from an information theoretic perspective [35]. Thus, we like to find the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ that maps $\mathbf{s}$ to a representation $\mathbf{x}$ such that most information of the relevant RV $\mathbf{z}$ is included in $\mathbf{y}$, i.e., we maximize the Mutual Information (MI) $I(\mathbf{z};\mathbf{y})$ with

regard to $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ [36]:

$$\begin{aligned}
&\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\ I_{\boldsymbol{\theta}}\left(\mathbf{z};\mathbf{y}\right) && (4.1)\\
=&\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\ \mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}\left[\ln\frac{p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}{p(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{y})}\right] && (4.2)\\
=&\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\ \mathcal{H}\left(\mathbf{z}\right)-\mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y}),p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})\right) && (4.3)\\
=&\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\ \mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})]\ . && (4.4)
\end{aligned}$$

There, $\mathcal{H}\left(p(\mathbf{x}),q(\mathbf{x})\right)=\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\ln q(\mathbf{x})]$ is the cross-entropy between two pdfs/pmfs $p(\mathbf{x})$ and $q(\mathbf{x})$. Note the independence from $\boldsymbol{\theta}$ in $\mathcal{H}\left(\mathbf{z}\right)$ and dependence in $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$ and $p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})$ through the Markov chain $\mathbf{z}\to\mathbf{s}\to\mathbf{y}$. Problem (4.1) is convex with regard to the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ for fixed $p(\mathbf{s})$ [37], but not necessarily convex with regard to the encoder parameters $\boldsymbol{\theta}$. For example, it is non-convex if the encoder function is non-convex with regard to its parameters being typically the case with DNN encoders. It is worth mentioning that we so far have not set any constraint on the variables we deal with. Hence, the form of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ has to be constrained to avoid learning a trivial identity mapping $\mathbf{y}=\mathbf{s}$. We indeed constrain the optimization by our communication channel $p(\mathbf{y}|\mathbf{x})$ we assume to be given.

If the calculation of the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$ in (4.4) is intractable, we are able to replace it by a variational distribution $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ with parameters $\boldsymbol{\varphi}$. Similar to the transmitter, DNNs are usually proposed [21], [38] for the design of the approximate posterior $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ at the receiver. To improve the performance complexity trade-off, the application of *deep unfolding* can be considered, a model-driven learning approach that introduces model knowledge of $p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{z})$ to create $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ [8], [39]. With $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$, we are able to define a MI Lower BOund (MILBO) [36] similar to the well-known Evidence Lower BOund (ELBO) [7]:

$$\begin{aligned}
I_{\boldsymbol{\theta}}\left(\mathbf{z};\mathbf{y}\right)&\geq\mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] && (4.5)\\
&=\mathrm{E}_{\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y})}\left[\mathrm{E}_{\mathbf{z}\sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})]\right] && (4.6)\\
&=-\mathrm{E}_{\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y})}[\mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}),q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right)] && (4.7)\\
&=-\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}\ . && (4.8)
\end{aligned}$$

The lower bound holds since $-\mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y}),p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})\right)$ itself is a lower bound of the expression in (4.3) and $\mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})/q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})]\geq 0$. Now, we can calculate optimal values of $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ of our semantic communication design by minimizing the amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ in (4.7), i.e., marginalized across observations $\mathbf{y}$ [8].

Thus, the idea is to learn parametrizations of the transmitter discriminative model and of the variational receiver posterior, e.g., by AEs or reinforcement learning. Note that, in our semantic problem (4.1), we do not auto-encode the hidden $\mathbf{z}$ itself, but encode $\mathbf{s}$ to obtain $\mathbf{z}$ by decoding. This can be seen from Figure 4.1 and by rewriting the amortized cross-entropy (4.7) and (4.8):

$$\begin{aligned}\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} &= \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}\left(p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}), q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right)] && (4.9)\\ &= \mathrm{E}_{\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{z}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{z})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] && (4.10)\\ &= \mathrm{E}_{\mathbf{s},\mathbf{z}\sim p(\mathbf{s},\mathbf{z})}\left[\mathrm{E}_{\mathbf{x}\sim p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}\left[\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})]\right]\right] .\end{aligned}$$

We can further prove the amortized cross-entropy to be decomposable into

$$\begin{aligned}\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} &= \mathrm{E}_{\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y})}\left[\,\mathrm{E}_{\mathbf{z}\sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y}) + \ln p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}) - \ln p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})]\right] && (4.11)\\ &= \mathrm{E}_{\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y})}[D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}) \parallel q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right)] + \underbrace{\mathcal{H}\left(\mathbf{z}|\mathbf{y}\right)}_{=-I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})+\mathcal{H}(\mathbf{z})} && (4.12)\\ &= \mathcal{H}\left(\mathbf{z}\right) - \underbrace{I_{\boldsymbol{\theta}}\left(\mathbf{z};\mathbf{y}\right)}_{\text{enc. objective}} + \underbrace{\mathrm{E}_{\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y})}[D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}) \parallel q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right)]}_{\text{dec. objective}} . && (4.13)\end{aligned}$$

In the end, maximization of the MILBO with regard to $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ balances maximization of the mutual information $I_{\boldsymbol{\theta}}\left(\mathbf{z};\mathbf{y}\right)$ and minimization of the Kullback–Leibler (KL) divergence $D_{\mathrm{KL}}\left(p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}) \parallel q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right)$. The former objective can be seen as a regularization term that favors encoders with high mutual information, for which decoders can be learned that are close to the true posterior.

4.5.4 Classical Design Approach

If we consider classical communication design approaches, we would solve the problem

$$\arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} I\left(\mathbf{s};\mathbf{y}\right) \tag{4.14}$$

which relates to Joint Source-Channel Coding (JSCC). There, the aim is to find a representation $\mathbf{x}$ that retains a significant amount of information about the source signal $\mathbf{s}$ in $\mathbf{y}$. Again, we can apply the lower bound (4.8). In fact, bounding (4.14) by (4.8) shows that approximate maximization of the mutual information justifies the minimization of the cross-entropy in the AEs approach [6], often seen in recent wireless communication literature [6], [19], [28].

4.5.5 Information Bottleneck View

It should be stressed that we have not set any constraints on the variables in the InfoMax problem so far. However, in many applications, compression is needed because of the limited information rate. Therefore, we can formulate an optimization problem where we like to maximize the relevant information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$ subject to the constraint to limit the compression rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ to a maximum information rate I_{C}:

$$\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\ I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y}) \quad \text{s.t.} \quad I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq I_{\mathrm{C}}\,. \tag{4.15}$$

Problem (4.15) is an important variation of the InfoMax principle and called the Information Bottleneck (IB) problem [10], [29], [40], [41]. The IB method introduced by Tishby et al. [29] has been the subject of intensive research for years and has proven to be a suitable mathematical/information-theoretical framework for solving numerous problems—as well as in wireless communications [30], [31], [42], [43]. Note that we aim for an encoder that compresses $\mathbf{s}$ into a compact representation $\mathbf{x}$ for discrete RVs by clustering and for continuous RVs by dimensionality reduction.

To solve the constrained optimization problem (4.15), we can use Lagrangian optimization and obtain

$$\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\ I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y}) - \beta I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \tag{4.16}$$

with Lagrange multiplier $\beta \geq 0$. The Lagrange multiplier β allows the defining of a trade-off between the relevant information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$ and compression rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$, which indicates the relation to rate distortion theory [30]. With $\beta = 0$, we have the InfoMax problem (4.1) whereas for $\beta \to \infty$ we minimize compression rate. Calculation of the mutual information terms may be computationally intractable, as in the InfoMax problem (4.1). Approximation approaches can be found in [44], [45]. Notable exceptions include if the RVs are all discrete or Gaussian distributed.

We note that in [10], [26] the authors already introduced the IB problem to task-oriented communications. But [10], [26] do not address our viewpoint or classification. We compress and channel encode the messages/communications source $\mathbf{s}$ for given entailment $p(\mathbf{s}|\mathbf{z})$, in the sense of a data-reduced and reliable communication of the semantic RV $\mathbf{z}$. Basically, we implement joint source-channel coding of $\mathbf{s}$ s.t. preserving the semantic RV $\mathbf{z}$, and we do not differentiate between Levels A and B, as indicated by Weaver's notion outlined in Section 4.3. Indeed, we draw a direct connection to IB compared to related semantic communication literature [19], [21], [38] that, so far, only included optimization with terms reminiscent of the IB problem.

Semantic Information Bottleneck

This article does not only distinct itself on a conceptual, but also on a technical level from [26], [34]. We follow a different strategy to solve (4.15).

First, using the data processing inequality [46], we see that the compression rate is upper bounded by the mutual information of the encoder $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x})$ and that of the channel $I(\mathbf{x};\mathbf{y})$:

$$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq \min\left\{I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}), I(\mathbf{x};\mathbf{y})\right\} . \tag{4.17}$$

In case of negligible encoder compression $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}) > I(\mathbf{x};\mathbf{y})$, the channel becomes the limiting factor of information rate. For example, with a deterministic continuous mapping $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$, this is true since $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}) \to \infty$. Using the chain rule of mutual information [46], we see that this upper bound on compression rate grows with the dimension of $\mathbf{x}$, i.e., the number of channel uses N_{Tx}:

$$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq I(\mathbf{x};\mathbf{y}) = \sum_{n=1}^{N_{\mathrm{Tx}}} \underbrace{I(x_n;\mathbf{y}|x_{n-1},\ldots,x_1)}_{\geq 0} . \tag{4.18}$$

Assuming $\mathbf{y}$ to be conditional dependent on x_n given $x_{n-1},\ldots,x_1$, i.e., $p(\mathbf{y}|x_n,\ldots,x_1) \neq p(\mathbf{y}|x_{n-1},\ldots,x_1)$ being, e.g., true for an Additive White Gaussian Noise (AWGN) channel, it is $I(x_n;\mathbf{y}|x_{n-1},\ldots,x_1) > 0$ [46] and the sum in (4.18) indeed strictly increases. Replacing $\mathbf{y}$ in $I(\mathbf{x};\mathbf{y})$ of (4.18) by $\mathbf{s}$, the result also holds for encoder compression $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x})$, respectively. Hence, increasing the encoder output dimension N_{Tx}, we can increase the possible compression rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$. Interchanging $\mathbf{x}$ and $\mathbf{y}$ in (4.18), we see that the same holds for the receiver input dimension N_{Rx}.

Furthermore, the mutual information of the channel and, thus, the compression rate are upper bounded by channel capacity:

$$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq I(\mathbf{x};\mathbf{y}) \leq \max_{p(\mathbf{x});\mathrm{E}\left[|x_n|^2\right]\leq 1} I(\mathbf{x};\mathbf{y}) = C . \tag{4.19}$$

For example, with an AWGN channel with noise standard deviation σ_{n}, we have $C = N_{\mathrm{Tx}}/2 \cdot \ln\left(1 + 1/\sigma_{\mathrm{n}}^2\right)$ again increasing with N_{Tx}.

Now, let us assume the RVs to be discrete so that $\mathcal{H}(\mathbf{x}|\mathbf{s}) \geq 0$. Indeed, this is true if the RVs are processed discretely with finite resolution on digital signal processors, as in the numerical example of Section 4.6. As long as $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}) < C$, all information of the discrete RVs can be transmitted through the channel with arbitrary low error probability according to Shannon's channel coding theorem [1]. Then, we can upper bound encoder compression

$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x})$ and thus compression rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ by the sum of entropies of any output x_n [46] of the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$—each with cardinality $|\mathcal{M}_x|$:

$$I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \underbrace{\mathcal{H}(\mathbf{x}|\mathbf{s})}_{\geq 0} \leq \mathcal{H}(\mathbf{x}) \leq \sum_{n=1}^{N_{\mathrm{Tx}}} \mathcal{H}(x_n) \leq N_{\mathrm{Tx}} \cdot \log_2(|\mathcal{M}_x|)\,. \tag{4.20}$$

Note that the entropy sum in (4.20) grows again with N_{Tx} for discrete RVs since $0 \leq \mathcal{H}(x_n) \leq \log_2(|\mathcal{M}_x|)$. Moreover, we can define an encoder capacity $C_{\boldsymbol{\theta}}$ analogous to channel capacity C in (4.19) that upper bounds encoder compression $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x})$. It may be restricted by the chosen (DNN) model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ and optimization procedure with regard to $\boldsymbol{\theta}$, i.e., the hypothesis class [7].

In summary, we have proven by (4.19) and (4.20) that there is an information bottleneck when maximizing the relevant information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$ either due to the channel distortion $I(\mathbf{x};\mathbf{y})$ or encoder compression $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{x})$.

To fully exploit the available resources, we set constraint I_{C} to be equal to the upper bound, i.e., channel capacity C or the upper bound on encoder compression rate $N_{\mathrm{Tx}} \cdot \log_2(|\mathcal{M}_x|)$. In both cases, the upper bound grows (linearly) with the encoder output dimension N_{Tx}, and, thus, we can set the constraint I_{C} higher or lower by choosing N_{Tx}.

With fixed constraint I_{C}, we maximize the relevant information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$. By doing so, we derive an exact solution to (4.15) that maximizes $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$ for a fixed encoder output dimension that bounds the compression rate. As in the InfoMax problem, we can exploit the MILBO to use the amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ in (4.9) as the optimization criterion.

Variational Information Bottleneck

In [26], however, the authors solve the variational IB problem of (4.16) and require tuning of β. Albeit also using the MILBO as a variational approximation to the first term in (4.16), they introduce a KL divergence term as an upper bound to compression rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ derived by $D_{\mathrm{KL}}(p_{\boldsymbol{\theta}}(\mathbf{y}) \parallel q_{\boldsymbol{\vartheta}}(\mathbf{y})) \geq 0$ with some variational distribution $q_{\boldsymbol{\vartheta}}(\mathbf{y})$ with parameters $\boldsymbol{\vartheta}$ [44]. Then, the variational IB objective function reads [44]:

$$\begin{aligned} I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y}) - \beta I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \geq\; & \mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \\ & - \beta \cdot \mathrm{E}_{\mathbf{s}\sim p(\mathbf{s})}[D_{\mathrm{KL}}(p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) \parallel q_{\boldsymbol{\vartheta}}(\mathbf{y}))]\,. \end{aligned} \tag{4.21}$$

Moreover, the authors use a log-uniform distribution as the variational prior $q_{\boldsymbol{\vartheta}}(\mathbf{y})$ in [26] to induce sparsity on $\mathbf{y}$ so that the number of outputs

is dynamically determined based on the channel condition or SNR, i.e., $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}, \sigma_{\mathrm{n}}^2)$. The approach additionally necessitates approximation of the KL divergence term in (4.21) and estimation of the noise variance σ_{n}^2.

With our approach we avoid the additional approximations and tuning of the hyperparameter β in (4.21) possibly enabling better semantic performance as well as reduced inference and training complexity at the cost of full usage of N_{Tx} channels even when the channel capacity C enables its reduction. We leave a numerical comparison to [26] for future research as this is out of the scope of this paper.

4.5.6 Implementation Considerations

Now, we will provide important implementation considerations for optimization of (4.8), (4.10) and (4.15). We note that computation of the MILBO leads to similar problems as for the ELBO [35]; if calculating the expected value in (4.10) cannot be solved analytically or is computationally intractable—as typically the case with DNNs—we can approximate it using Monte Carlo sampling techniques with N samples $\{(\mathbf{z}_i, \mathbf{s}_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$.

For Stochastic Gradient Descent (SGD)-based optimization like, e.g., in the AE approach, the gradient with regard to $\boldsymbol{\varphi}$ can then be calculated by

$$\frac{\partial \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\varphi}}^{\mathrm{CE}}}{\partial \boldsymbol{\varphi}} = \frac{\partial}{\partial \boldsymbol{\varphi}} \mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) p(\mathbf{s}|\mathbf{z}) p(\mathbf{z})} [-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \tag{4.22}$$

$$= - \mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) p(\mathbf{s}|\mathbf{z}) p(\mathbf{z})} \left[\frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})}{\partial \boldsymbol{\varphi}} \right] \tag{4.23}$$

$$\approx - \frac{1}{N} \sum_{i=1}^{N} \frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)}{\partial \boldsymbol{\varphi}} \tag{4.24}$$

with N being equal to the batch size N_{b} and by application of the backpropagation algorithm to $\frac{\partial}{\partial \boldsymbol{\varphi}} \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i) = \frac{\partial}{\partial \boldsymbol{\varphi}} q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i) / q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)$ in Automatic Differentiation Frameworks (ADFs), e.g., TensorFlow and PyTorch. Computation of the so-called Reinforce gradient with regard to $\boldsymbol{\theta}$ leads to a high variance of the gradient estimate since we sample with regard to the distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ dependent on $\boldsymbol{\theta}$ [35].

Reparametrization Trick

Leveraging the direct relationship between $\boldsymbol{\theta}$ and $\mathbf{y}$ in $\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ can help reduce the estimator's high variance. Typically, e.g., in the Variational AutoEncoder (VAE) approach, the *reparametrization trick* is used to achieve this [35]. Here, we can apply it if we can decompose the latent variable

$\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ into a differentiable function $\mathbf{y} = f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{n})$ and a RV $\mathbf{n} \sim p(\mathbf{n})$ independent of $\boldsymbol{\theta}$. Fortunately, the typical forward model of a communication system $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ fulfills this criterion. Assuming a deterministic DNN encoder $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$ and additive noise $\mathbf{n}$ with covariance $\boldsymbol{\Sigma}$, we can thus rewrite $\mathbf{y}$ into $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{n}) = \mu_{\boldsymbol{\theta}}(\mathbf{s}) + \boldsymbol{\Sigma}^{1/2} \cdot \mathbf{n}$ and, accordingly, the amortized cross-entropy gradient into:

$$\frac{\partial \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}}{\partial \boldsymbol{\theta}} = -\frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s}) p(\mathbf{s},\mathbf{z})} [\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \tag{4.25}$$

$$= -\mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{n} \sim p(\mathbf{n}) p(\mathbf{s}|\mathbf{z}) p(\mathbf{z})} \left[\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{n})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})}{\partial \mathbf{y}} \right] \tag{4.26}$$

$$\approx -\frac{1}{N} \sum_{i=1}^{N} \frac{\partial f_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{n}_i)}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)}{\partial \mathbf{y}} \bigg|_{\mathbf{y} = f_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{n}_i)} . \tag{4.27}$$

The reparametrization trick can be easily implemented in ADFs by adding a noise layer—typically used for regularization in ML literature— after (DNN) function $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$. Then, our loss function (4.10) amounts to

$$\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} \approx -\frac{1}{N} \sum_{i=1}^{N} \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i | \mathbf{y}_i = f_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{n}_i)) . \tag{4.28}$$

This enables the joint optimization of both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$, as demonstrated in recent works [6], treating unsupervised optimization of AEs as a supervised learning problem.

4.6 Example of Semantic Information Recovery

In this section, we provide one numerical example of data-driven semantics to explain what we understand under a semantic communication design and to show its benefits: It is the task of image classification. In fact, we consider our example of the biologist from Section 4.5.2 who wants to know which type the tree is.

For the remainder of this article, we will thus assume the hidden semantic RV to be a one-hot vector $\mathbf{z} \in \{0, 1\}^{M \times 1}$ where all elements are zero except for one element representing one of the M image classes. Then, the semantic channel $p(\mathbf{s}|\mathbf{z})$ (see Figure 4.1) generates images belonging to this class, i.e., the source signal $\mathbf{s}$.

Note that for point-to-point transmission, as in [26], we could first classify the image based on the posterior $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{s})$, as shown in Figure 4.2 and transmit

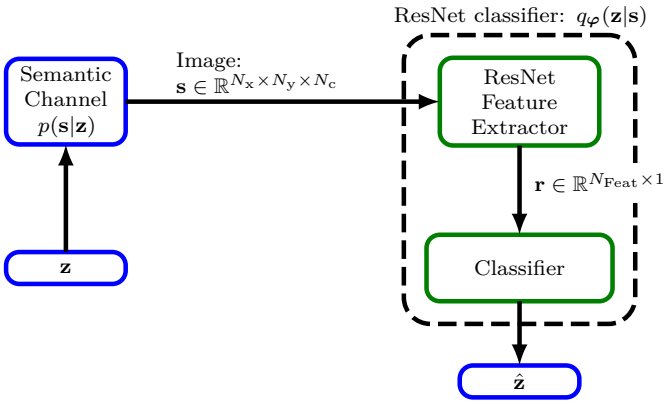

Figure 4.2: Central image processing: Based on the images, ResNet extracts semantics by classification.

the estimate $\hat{\mathbf{z}}$ (encoded into $\mathbf{x}$) through the physical channel since this would be most rate or bandwidth efficient.

But if the image information is distributed across multiple agents, all (sub) images may contribute useful information for classification. We could thus lose information when making hard decisions on each transmitter's side. In the distributed setting, transmission and combination of features, i.e., soft information, is crucial to obtain high classification accuracy.

Further, we note that transmission of full information, i.e., raw image data $\mathbf{s}$, through a wireless channel from each agent to a central unit for full image classification would consume a lot of bandwidth. This case is also shown in Figure 4.2 assuming perfect communication links between the output of the semantic channel and the input of the ResNet feature extractor.

Therefore, we investigate a distributed setting shown in Figure 4.3. There, each of four agents sees its own image $\mathbf{s}_1, \ldots, \mathbf{s}_4 \sim p(\mathbf{s}_i|\mathbf{z})$ being generated by the same semantic RV $\mathbf{z}$. Based on these images, a central unit shall extract semantics, i.e., perform classification. We propose to optimize the four encoders $p_{\boldsymbol{\theta}_i}(\mathbf{x}_i|\mathbf{s}_i)$ with $i = 1, \ldots, 4$, each consisting of a bandwidth efficient feature extractor (ResNet Feature Extractor i) and transmitter (Tx i) **jointly** with a decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4]^T)$, consisting of a Receiver (Rx) and concluding classifier (Classifier), with regard to cross-entropy (4.10) of the semantic labels (see Figure 4.3). Hence, we maximize the system's overall semantic measure, i.e., classification accuracy. Note that this scenario is different from both [33], [34]; we include a physical communication channel (Comm. Channel i) since we aim to transmit and not only compress. For the

sake of simplicity, we assume orthogonal channel access. The IB is addressed by limiting the number of channel uses, which defines the constraint I_{C} in (4.15).

As a first demonstration example, we use the grayscale MNIST and colored CIFAR10 datasets with $M = 10$ image classes [47]. We assume that the semantic channel generates an image that we divide into four equally sized quadrants and each agent observes one quadrant $\mathbf{s}_1, \ldots, \mathbf{s}_4 \in \mathbb{R}^{N_{\mathrm{x}} \times N_{\mathrm{y}} \times N_{\mathrm{c}}}$, where N_{x} and N_{y} is the number of image pixels in the x- and y-dimension, respectively, and N_{c} is the number of color channels. Albeit this does not resemble a realistic scenario, note that we can still show the basic working principle and ease implementation.

4.6.1 ResNet

For the design of the overall system, we rely on a famous DNN approach for feature extraction, breaking records at the time of invention: ResNet [47], [48]. The key idea of ResNet is that it consists of multiple residual units. Each unit's input is fed directly to its output and if the dimensions do not match, a convolutional layer is used. This structure enables fast training and convergence of DNNs since the training error can be backpropagated to early layers through these skip connections. From a mathematical point of view, usual DNNs have the design flaw that using a larger function class, i.e., more DNN layers, does not necessarily increase the expressive power. However, this holds for nested functions like ResNet which contain the smaller classes of early layers.

Each residual unit itself consists of two Convolutional NNs (CNNs) with subsequent batch normalization and ReLU activation function, i.e., $\rho_{\mathrm{relu}}(\cdot) = \max(\cdot, 0)$, to extract translation invariant and local features across two spatial dimensions N_{x} and N_{y}. Color channels, like in CIFAR10, add a third dimension $N_{\mathrm{c}} = 3$ and additional information. The idea behind stacking multiple layers of CNNs is that features tend to become more abstract from early layers (e.g., edges and circles) to final layers (e.g., beaks or tires).

In this work, we use the pre-activation version of ResNet without bottlenecks from [47], [48] implemented for classification on the dataset CIFAR10. In Table 4.1, we show its structure for the distributed scenario from Figure 4.3. There, ResNetBlock is the basic building block of the ResNet architecture. Each block consists of multiple residual unit (res. un.) and we use 2 for the MNIST dataset and 3 for the CIFAR10 dataset, which means we use ResNet14 and ResNet20, respectively. We arrive at the architecture of central image processing from Figure 4.2 by removing the components Tx, (physical) Channel, and Rx and increasing each spatial dimension by 2

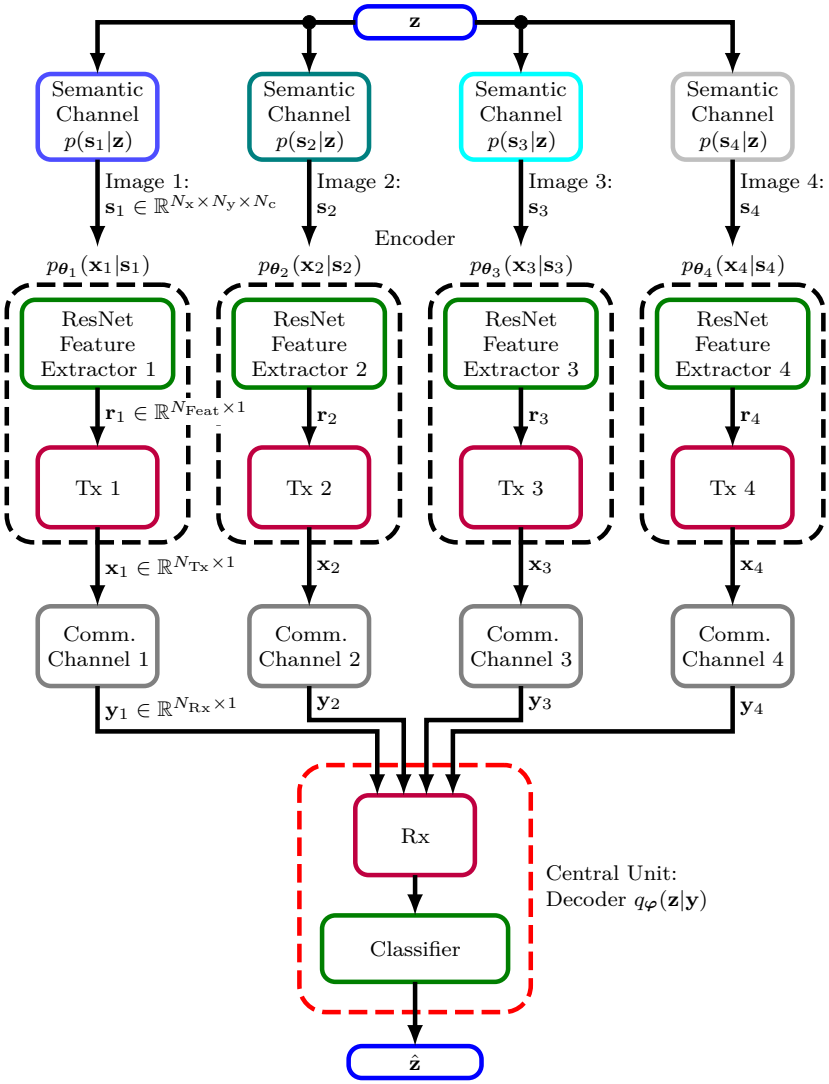

Figure 4.3: Semantic INFOrmation TraNsmission and RecoverY (SINFONY) for distributed agents. Each agent extracts features for bandwidth-efficient transmission. Based on the received signal, the central unit extracts semantics by classification.

Table 4.1: Semantic INFOrmation TraNsmission and RecoverY (SINFONY)–DNN architecture for distributed image classification.

Component	Layer	Dimension
Input	Image (MNIST, CIFAR10)	$(14, 14, 1)$, $(16, 16, 3)$
$4\times$	Conv2D	$(14, 14, 14)$, $(16, 16, 16)$
Feature	ResNetBlock (2/3 res. un.)	$(14, 14, 14)$, $(16, 16, 16)$
Extractor	ResNetBlock (2/3 res. un.)	$(7, 7, 28)$, $(8, 8, 32)$
	ResNetBlock (2/3 res. un.)	$(4, 4, 56)$, $(4, 4, 64)$
	Batch Normalization	$(4, 4, 56)$, $(4, 4, 64)$
	ReLU activation	$(4, 4, 56)$, $(4, 4, 64)$
	GlobalAvgPool2D	(56), (64)
$4\times$ Tx	ReLU	N_{Tx}
	Linear	N_{Tx}
	Normalization (dim.)	N_{Tx}
$4\times$ Channel	AWGN	N_{Tx}
Rx	ReLU ($4\times$ shared)	$(2, 2, N_{\text{w}})$
	GlobalAvgPool2D	N_{w}
Classifier	Softmax	$M = 10$

to contain all quadrants of the original image. For further implementation details, we refer the reader to the original work [48].

4.6.2 Distributed Semantic Communication Design Approach

Our key idea here is to modify ResNet with regard to the communication task by splitting it at a suitable point where a representation $\mathbf{r} \in \mathbb{R}^{N_{\text{Feat}} \times 1}$ of semantic information with low-bandwidth is present (see Figures 4.2 and 4.3). ResNet and CNNs in general can be interpreted to extract features; with full images, we obtain a feature map of size $8 \times 8 \times N_{\text{Feat}}$ after the last ReLU activation (see Table 4.1). These local features are aggregated by the global average pooling layers across the 2 spatial dimensions into $\mathbf{r}$. Based on these

N_{Feat} global features in $\mathbf{r}$, the softmax layer finally classifies the image. We note that the features contain the relevant information with regard to the semantic RV $\mathbf{z}$ and are of low dimension compared to the original image or even its sub-images, i.e., 64 compared to $16 \times 16 \times 3 = 768$ for CIFAR10.

Therefore, we aim to transmit each agent's local features $\mathbf{r}_i \in \mathbb{R}^{N_{\text{Feat}} \times 1}$ $(i = 1, \ldots, 4)$ instead of all sub-images $\mathbf{s}_i$ and add the component Tx in Table 4.1 to encode the features $\mathbf{r}_i$ into $\mathbf{x}_i \in \mathbb{R}^{N_{\text{Tx}} \times 1}$ for transmission through the wireless channel (see Figure 4.3). We note that $\mathbf{x}_i \in \mathbb{R}^{N_{\text{Tx}} \times 1}$ is analog and that the output dimension N_{Tx} of $\mathbf{x}_i$ defines the number of channel uses per agent/image. Note that the less often we use the wireless channel (N_{Tx}), the less information we transmit but the less bandwidth we consume, and vice versa. Hence, the number of channel uses defines the IB in (4.15). We implement the Tx module by DNN layers. To limit the transmission power to one, we constrain the Tx output by the norm along the training batch or the encoding vector dimension (dim.), i.e., $x_n = \tilde{x}_n / \sqrt{\mathrm{E}\left[\tilde{x}_n^2\right]}$ or $\mathbf{x}_i = \sqrt{N_{\text{Tx}}} \cdot \tilde{\mathbf{x}}_i / \left\|\tilde{\mathbf{x}}_i\right\|_2$, where $\tilde{\mathbf{x}}_i \in \mathbb{R}^{N_{\text{Tx}} \times 1}$ is the output of the layer Linear from Table 4.1. For numerical simulations, we choose all Tx layers to have width N_{Tx}.

At the receiver side, we use a single Rx module only with shared DNN layers and parameters $\boldsymbol{\varphi}_{\text{Rx}}$ for all inputs $\mathbf{y}_i$. This setting would be optimal if any feature is reflected in any sub-image and if the statistics of the physical channels are the same. Exploiting the prior knowledge of location-invariant features and assuming AWGN channels, this design choice seems reasonable. In our experiments, all layers of the Rx module have width N_{w}. A larger layer width N_{w} is equivalent to more computing power.

The output of the Rx module can be interpreted as a representation of the image features $\mathbf{r}_i$ with index i indicating the spatial location. Thus, we have a representation of a feature map of size $(2, 2, N_{\text{w}})$ that we aggregate across the spatial dimension according to the ResNet structure. Based on this semantic representation, a softmax layer with 10 units finally computes class probabilities $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ whose maximum is the maximum a posteriori estimate $\hat{z}$. In the following, we name our proposed approach Semantic INFOrmation TraNsmission and RecoverY (SINFONY).

4.6.3 Optimization Details

We evaluate SINFONY in TensorFlow 2 [49] on the MNIST and CIFAR10 datasets. The source code is available in [50] and the default simulation and training parameters are summarized in Table 4.2. We split the dataset into $N_{\text{train}} = 60$ k or 50 k training data and 10 k validation data samples, respectively. For preprocessing, we normalize the pixel inputs to range $[0, 1]$,

Table 4.2: Default simulation and training parameters.

Parameter Name	Variable	Value (MNIST, CIFAR10)
Batch size	N_b	64
Epoch number	N_e	20, 200
Learning rate	ϵ	Schedule: $\epsilon = \{0.1, 0.01, 0.001\}$ with $N_\mathrm{e} = \{3, 6\}$, $\{100, 150\}$
Optimizer		SGD with momentum= 0.9
Preprocessing		Input normalization to $[0, 1]$
Training SNR range	$\mathrm{SNR}_\mathrm{train}$	$[-4, 6]$ dB
Training dataset size	N_train	60 k, 50 k
Validation dataset size		10 k
Weight decay		0.0001
Weight initialization		Glorot uniform, ReLU: He uniform
Encoder normalization	dim.	Batch dimension
Rx layer width	N_w	56, 64

but we do not use data augmentation, in contrast to [47], [48], yielding slightly worse accuracy. The ReLU layers are initialized with uniform distribution according to He and all other layers according to Glorot [51].

In the case of CIFAR10 classification with central image processing and original ResNet, we need to train $N_{\boldsymbol{\theta}} + N_{\boldsymbol{\varphi}} = 273,066$ parameters. We like to stress that although we divided the image input into four smaller pieces, this number grows more than four times to $4N_{\boldsymbol{\theta}} + N_{\boldsymbol{\varphi}} = 1,127,754$ with $N_\mathrm{Tx} = N_\mathrm{Feat} = 64$ for SINFONY. The reason lies in the ResNet structure with minor dependence on the input image size and that we process at four agents with an additional Tx module. Only $N_{\boldsymbol{\varphi}} = 4810$ parameters amount to the Rx module and classification, i.e., the central unit. We note that the number of added Tx and Rx parameters of $33,560$ and 3192 is relatively small. Since the number of parameters only weakly grows with Rx layer width N_w in our design, we choose $N_\mathrm{w} = N_\mathrm{Feat}$ as the default.

For optimization of the cross-entropy (4.10) or the loss function (4.28), we use the reparametrization trick from Section 4.5.6 and SGD with a momentum of 0.9 and a batch size of $N_\mathrm{b} = 64$. We add l_2-regularization with a weight decay of 0.0001 as in [47], [48]. The learning rate of $\epsilon = 0.1$ is reduced to 0.01 and 0.001 after $N_\mathrm{e} = 100$ and 150 epochs for CIFAR10 and

after 3 and 6 epochs for MNIST. In total, we train for $N_\mathrm{e} = 200$ epochs with CIFAR10 and for 20 with MNIST. In order to optimize the transceiver for a wider SNR range, we choose the training SNR to be uniformly distributed within $\mathrm{SNR}_\mathrm{train} \in [-4, 6]$ dB where $\mathrm{SNR} = 1/\sigma_\mathrm{n}^2$ with noise variance σ_n^2.

4.6.4 Numerical Results and Discussion

In the following, we will investigate the influence of specific design choices on our semantic approach SINFONY. Then, we compare a semantic transmission approach with a classical Shannon-based transmission approach. The design choices are as follows:

- **Central:** Central and joint processing of full image information by the ResNet classifier, see Figure 4.2. It indicates the maximum achievable accuracy.
- **SINFONY – Perfect comm.:** The proposed distributed design SINFONY trained with perfect communication links and without channel encoding, i.e., Tx and Rx module, but with Tx normalization layer. Thus, the plain and power-constrained features are transmitted with $N_\mathrm{Tx} = N_\mathrm{Feat}$ channel uses. It serves as the benchmark since it indicates the maximum performance of the distributed design.
- **SINFONY – AWGN:** SINFONY – Perfect comm. evaluated with AWGN channel.
- **SINFONY – AWGN + training:** SINFONY – Perfect comm. trained with AWGN channel.
- **SINFONY – Tx/Rx ($\boldsymbol{N_\mathrm{Tx} = N_\mathrm{Feat}}$):** SINFONY trained with channel encoding, i.e., Tx and Rx module, and $N_\mathrm{Tx} = N_\mathrm{Feat}$ channel uses.
- **SINFONY – Tx/Rx ($\boldsymbol{N_\mathrm{Tx} < N_\mathrm{Feat}}$):** SINFONY trained with channel encoding and $N_\mathrm{Tx} < N_\mathrm{Feat}$ channel uses for feature compression.
- **SINFONY – Classic digital comm.:** SINFONY – Perfect comm. with classic digital communications (Huffman coding, LDPC coding with belief propagation decoding, and digital modulation) as additional Tx and Rx modules. For details, see Section 4.6.4.
- **SINFONY – Analog semantic AE:** SINFONY – Perfect comm. with ML-based analog communications (AE with regard to $\mathbf{r}$) as additional Tx and Rx modules. It is basically the semantic communication

approach from [19], [21], [28], [32]. For details, see Section 4.6.4 – Semantic vs. Classic Design.

Since meaning is expressed by the RV $\mathbf{z}$, we use classification accuracy to measure semantic transmission quality. For illustration in logarithmic scale, we show the opposite of accuracy in all plots, i.e., classification error rate.

MNIST Dataset

The numerical results of our proposed approach SINFONY on the MNIST validation dataset are shown in Figure 4.4 for $N_{\mathrm{w}} = 56$. To obtain a fair comparison between transmit signals $\mathbf{x}_i \in \mathbb{R}^{N_{\mathrm{Tx}} \times 1}$ of different length N_{Tx}, we normalize the SNR by the spectral efficiency or rate $\eta = N_{\mathrm{Feat}}/N_{\mathrm{Tx}}$. First, we observe that the classification error rate of 0.5% of the central ResNet unit with full image information (Central) is smaller than that of 0.9% of SINFONY – Perfect comm. Note that we assume ideal communication links. However, the difference seems negligible considering that the local agents only see a quarter of the full images and learn features independently based on it.

With noisy communication links (SINFONY – AWGN), the performance degrades especially for SNR < 10 dB, and we can avoid degradation just partly by training with noise (SINFONY – AWGN + training). Introducing the Tx module (SINFONY – Tx/Rx $N_{\mathrm{Tx}} = 56$), we further improve classification accuracy at low SNR. If we encode the features from $N_{\mathrm{Feat}} = 56$ to only $N_{\mathrm{Tx}} = 14$ in the Tx module (SINFONY – Tx/Rx $N_{\mathrm{Tx}} = 14$) to have less channel uses/bandwidth (stronger bottleneck), the error rate is lowest compared to other SINFONY examples with non-ideal links for low normalized SNR. At high SNR, we observe a small error offset, which indicates lossy compression. In fact, our system SINFONY learns a reliable semantic encoding to improve the classification performance of the overall system with non-ideal links. Every design choice in Table 4.1 is well-motivated.

CIFAR10 Dataset

Comparing these results to the classification accuracy on CIFAR10 shown in Figure 4.5, we observe a similar behavior. But a few main differences become apparent. Central performs much better with a 12% error rate than SINFONY – Perfect comm. with 20%. We expect the reason to lie in the more challenging dataset with more color channels. Further, SINFONY – AWGN + training with $N_{\mathrm{Tx}} = N_{\mathrm{Feat}} = 64$ channel uses runs into a rather high error floor. Notably, even SINFONY – Tx/Rx ($N_{\mathrm{Tx}} = 16$) with fewer channel uses performs better than both SINFONY – AWGN and SINFONY –

AWGN + training over the whole SNR range and achieves channel encoding with negligible loss. This means adding more flexible channel encoding, i.e., Tx/Rx module, is crucial for CIFAR10.

Channel Uses Constraint

Since one of the main advantages of semantic communication lies in savings of information rate, we also investigate the influence of the number of channel uses N_{Tx} on MNIST classification error rate shown in Figure 4.6. From a practical point of view, we fix the information bottleneck by the output dimension N_{Tx} and maximize the mutual information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$. Decreasing the number of channel uses from $N_{\mathrm{Tx}} = 14$ to 2 and accordingly the upper bound I_{C} on the mutual information $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$, i.e., compression rate, from (4.19) or (4.20), we observe that the error floor at high SNR increases. We assume that, since the channel capacity decreases with SNR and N_{Tx}, higher compression is required for reliable transmission through the channel in the training SNR interval. For $N_{\mathrm{Tx}} = 56$, almost no error floor occurs at the cost of a smaller channel encoding gain. This means compression and channel coding are balanced based on the channel condition, i.e., training SNR region, to find the optimal trade-off to maximize $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$, which we can also observe in unshown simulations.

Semantic vs. Classic Design

Finally, we compare semantic and classic communication system designs. For the classic digital design, we first assume that the images are compressed lossless and protected by a channel code for transmission and reliable overall image classification by the central unit based on $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{s})$ (Central). We apply Huffman encoding to a block containing 100 images $\mathbf{s}_i$ where each RGB color entry contains 8 bits.

For fairness, we also compare to a SINFONY version where Tx and Rx modules of Table 4.1 are replaced by a classic design (SINFONY – Classic digital comm.). We first compress each element of the feature vector $\mathbf{r}_i$ that is computed in 32-bit floating-point precision in the distributed setting SINFONY – AWGN to 16-bit. Then, we apply Huffman encoding to a block containing 100 feature vectors of length N_{Feat}.

Further, we use a 5G LDPC channel code implementation from [52] with interleaver, rate $R_{\mathrm{C}} = \{0.25, 0.5, 0.25\}$ and long block length of $\{15360, 16000, 15360\}$, and modulate the code bits with {BPSK, BPSK, 16-QAM} such that we have, e.g., parameter set $\{0.25, 15360, \mathrm{BPSK}\}$ in one simulation. For digital image transmission, we use a rate of $R_{\mathrm{C}} = 0.25$ with a block length of 15360 and BPSK modulation. At the receiver, we assume

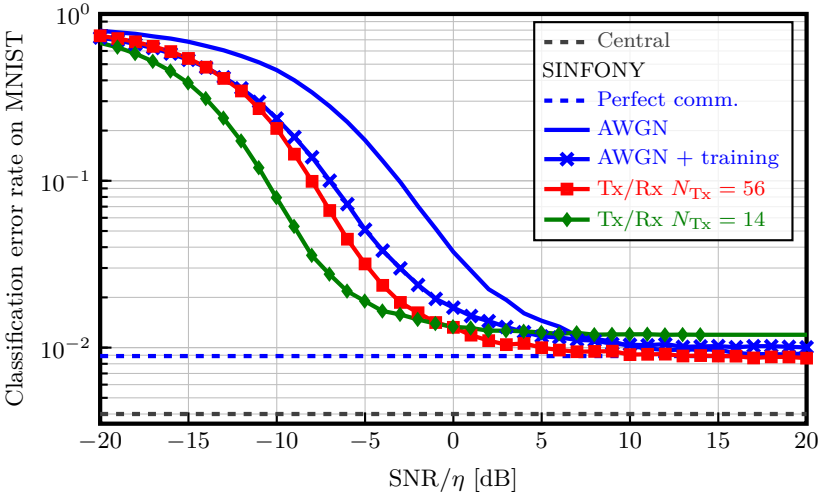

Figure 4.4: Classification error rate of different SINFONY examples (distributed setting) and central image processing on the MNIST validation dataset as a function of normalized SNR.

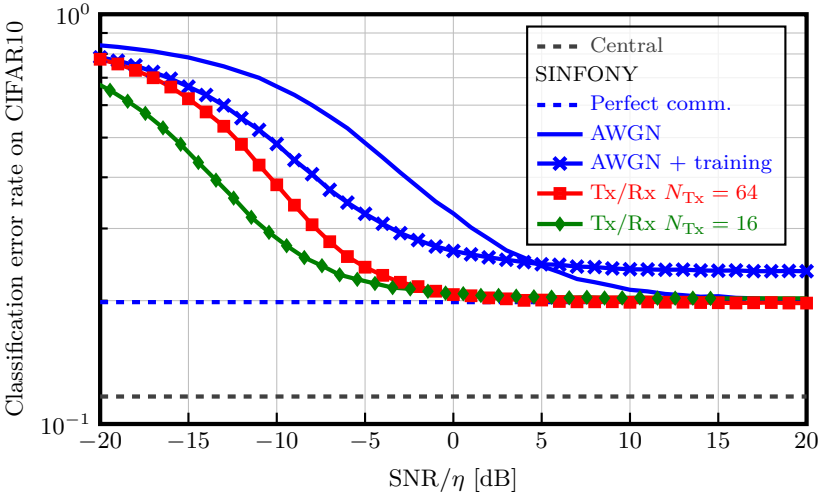

Figure 4.5: Classification error rate of different SINFONY examples (distributed setting) and central image processing on CIFAR10 as a function of normalized SNR.

belief propagation decoding, where the noise variance is perfectly known for LLR computation.

The results in Figure 4.7 reveal tremendous information rate savings for the semantic design with SINFONY. We observe an enormous SNR shift of roughly 20 dB compared to the classic digital design with regard to both image (Central) and feature transmission (SINFONY – Classic digital comm.). Note that the classic design is already near the Shannon limit and even if we improve it by ML we are only able to shift its curve by a few dB. The reason may lie in overall system optimization with SINFONY with regard to semantics and analog encoding of $\mathbf{x}$.

SINFONY vs. Analog "Semantic" Autoencoder

To distinguish both influences, we also implemented the approach of (4.14) according to Shannon by analog AEs. The analog AE has been introduced by O'Shea and Hoydis in [6]. From the viewpoint of semantic communication, it resembles the semantic approach from [19], [21], [28], [32] without differentiating between semantic and channel coding, and the mutual information constraint $I(\mathbf{x};\mathbf{y})$ like in [21]. We trained the AE matching the Tx and Rx module in Table 4.1 with mean square error criterion for reliable transmission of the feature vector $\mathbf{r}$ with SINFONY training settings. The Rx module consists of one ReLU layer of width $N_{\mathrm{w}} = N_{\mathrm{Tx}}$ providing the estimate of $\mathbf{r}$. We provide results (SINFONY – Analog semantic AE) in Figure 4.7. Indeed, most of the shift is due to analog encoding. By this means, we further avoid the typical thresholding behavior of a classic digital system seen at 14 dB.

In conclusion, this surprisingly clear result justifies an analog "semantic" communications design and shows its huge potential to provide bandwidth savings. However, introducing the semantic RV $\mathbf{z}$ by SINFONY, we can further shift the curve by 2 dB and avoid a slightly higher error floor compared to the analog "semantic" AE. We expect a larger performance gap with more challenging image datasets, such as CIFAR10. More importantly, the main benefit of SINFONY lies in its lower training complexity. We avoid separate and possibly iterative semantic and communication training procedures where in the first step we need to train SINFONY with ideal links, which is hard to achieve in practice.

4.7 Conclusions

Motivated by the approach of Bao, Basu et al. [16], [17] and inspired by Weaver's notion of semantic communication [2], we brought the terminus of a semantic source to the context of communications by considering its

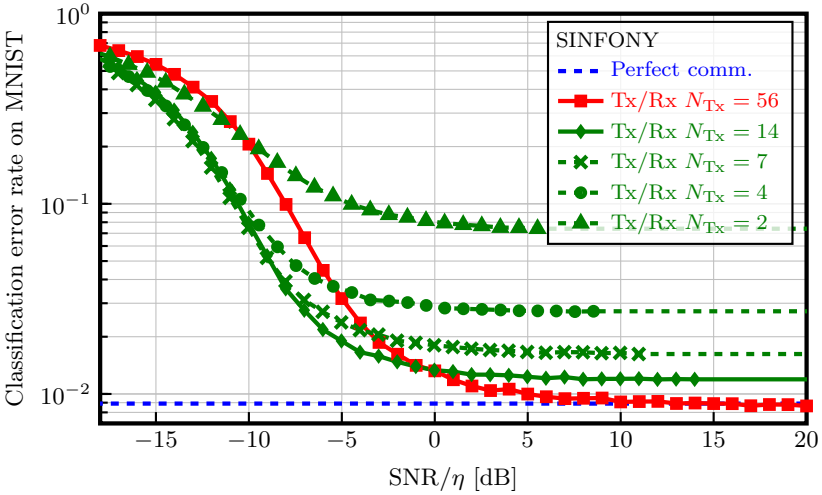

Figure 4.6: Classification error rate of SINFONY on the MNIST validation dataset for different rate/channel uses constraints as a function of normalized SNR.

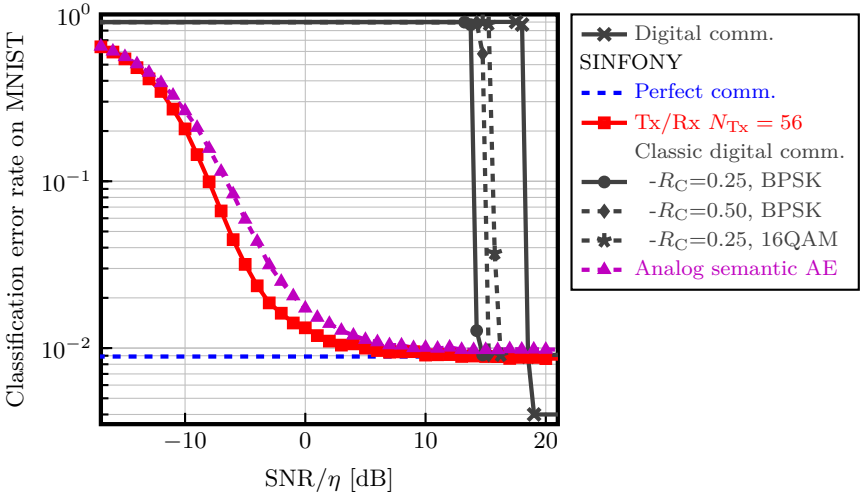

Figure 4.7: Classification error rate of SINFONY with different kinds of optimized Tx/Rx modules and central image processing with digital image transmission on the MNIST validation dataset as a function of normalized SNR.

complete Markov chain. We defined the task of semantic communication in the sense of a data-reduced and reliable transmission of communications sources/messages over a communication channel such that the semantic Random Variable (RV) at a recipient is best preserved. We formulated its design either as an information maximization or as an information bottleneck optimization problem covering important implementations aspects like the reparametrization trick and solved the problems approximately by minimizing the cross-entropy that upper bounds the negative mutual information. With this article, we distinguish from related literature [16], [17], [21], [26], [32] in both classification and perspective of semantic communication and a different ML-based solution approach.

Finally, we proposed the ML-based semantic communication system SINFONY for a distributed multipoint scenario: SINFONY communicates the meaning behind multiple messages that are observed at different senders to a single receiver for semantic recovery. We analyzed SINFONY by processing images as an example of messages. Notably, numerical results reveal a tremendous rate-normalized SNR shift up to 20 dB compared to classically designed communication systems.

Outlook

In this work, we contributed to the theoretical problem description of semantic communication and data-based ML solution approaches with DNNs. There remain open research questions such as:

- **Numerical Comparison to Variational IB:** It remains unclear if solving the variational IB problem (4.21) holds benefits compared to our proposed approach.
- **Implementation:** Optimization with the reparametrization trick requires a known differential channel model and training at one location with dedicated hardware such as graphics processing units [53]. In addition, large amounts of labeled data are required with data-driven ML techniques, which can be expensive and time-consuming to acquire and process. Hence, further research is required to clarify how a semantic design can be implemented efficiently in practice.
- **Semantic Modeling:** Developing effective models of semantics is crucial, and thus we proposed the usage of probabilistic models. If the underlying problem can be described by a well-known model, e.g., a physical process to be measured and processed by a sensor network [32], a promising idea is to apply model-based approaches based on Bayesian inference for encoding and decoding—potentially combined with the

technique of deep unfolding. In the context of NLP, design of knowledge graphs such as ontologies or taxonomies is a promising modeling approach for human language.

- **Inconsistent Knowledge Bases:** We assumed that sender and recipient share the same background knowledge base: How does performance deteriorate if there is a mismatch and how to deal with this problem [27]?

4.8 References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[2] W. Weaver, "Recent Contributions to the Mathematical Theory of Communication," in *The Mathematical Theory of Communication*, Urbana, IL, USA: The University of Illinois Press, 1949, pp. 261–281.

[3] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge Learning for B5G Networks With Distributed Signal Processing: Semantic Communication, Edge Computing, and Wireless Sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 9–39, Jan. 2023. DOI: 10.1109/JSTSP.2023.3239189.

[4] M. Gastpar, B. Rimoldi, and M. Vetterli, "To Code, or Not to Code: Lossy Source-Channel Communication Revisited," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1147–1158, May 2003. DOI: 10.1109/TIT.2003.810631.

[5] M. Gastpar and M. Vetterli, "Source-Channel Communication in Sensor Networks," in *Information Processing in Sensor Networks*, G. Goos, J. Hartmanis, J. van Leeuwen, F. Zhao, and L. Guibas, Eds., vol. 2634, Berlin, Heidelberg: Springer, 2003, pp. 162–177. DOI: 10.1007/3-540-36978-3_11.

[6] 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, Dec. 2017. DOI: 10.1109/TCCN.2017.2758370.

[7] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018. DOI: 10.1109/TCCN.2018.2881442.

[8] E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection With Low Complexity," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8214–8227, Dec. 2021. DOI: 10.1109/TCOMM.2021.3114682.

[9] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-Effectiveness Filtering and Control for Post-5G Wireless Connectivity," *Journal of the Indian Institute of Science*, vol. 100, no. 2, pp. 435–443, Apr. 2020. DOI: 10.1007/s41745-020-00165-6.

[10] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107 930, May 2021. DOI: 10.1016/j.comnet.2021.107930.

[11] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is Semantic Communication? A View on Conveying Meaning in the Era of Machine Intelligence," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, Dec. 2021. DOI: 10.23919/JCIN.2021.9663101.

[12] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, B. Soret, T. Soleymani, and K. H. Johansson, "Semantic Communications in Networked Systems: A Data Significance Perspective," *IEEE/ACM Transactions on Networking*, vol. 36, no. 4, pp. 233–240, Jul. 2022. DOI: 10.1109/MNET.106.2100636.

[13] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, Jan. 2023. DOI: 10.1109/JSAC.2022.3223408.

[14] L. Floridi, "Philosophical Conceptions of Information," in *Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information*, ser. Lecture Notes in Computer Science, Apr. 2009, pp. 13–53. DOI: 10.1007/978-3-642-00659-3_2.

[15] W. Hofkirchner, *Emergent Information: A Unified Theory of Information Framework*. World Scientific: Singapore, Dec. 2013, vol. 3. DOI: 10.1142/7805.

[16] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *IEEE Network Science Workshop (NSW 2011)*, West Point, NY, USA, Jun. 2011, pp. 110–117. DOI: 10.1109/NSW.2011.6004632.

[17] P. Basu, J. Bao, M. Dean, and J. Hendler, "Preserving Quality of Information by Using Semantic Relationships," *Pervasive and Mobile Computing*, vol. 11, pp. 188–202, Apr. 2014. DOI: 10.1016/j.pmcj.2013.07.013.

[18] B. Güler, A. Yener, and A. Swami, "The Semantic Communication Game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, Dec. 2018. DOI: 10.1109/TCCN.2018.2872596.

[19] N. Farsad, M. Rao, and A. Goldsmith, "Deep Learning for Joint Source-Channel Coding of Text," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, Calgary, Canada, Apr. 2018, pp. 2326–2330. DOI: 10.1109/ICASSP.2018.8461983.

[20] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep Learning based Semantic Communications: An Initial Investigation," in *IEEE Global Communications Conference (GLOBECOM 2020)*, Tapei, Taiwan, Dec. 2020, pp. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322296.

[21] 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, Apr. 2021. DOI: 10.1109/TSP.2021.3071210.

[22] Z. Weng, Z. Qin, and G. Y. Li, "Semantic Communications for Speech Signals," in *IEEE International Conference on Communications (ICC 2021)*, Virtual Conference, Jun. 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500590.

[23] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, "Deep Learning Enabled Semantic Communications with Speech Recognition and Synthesis," *IEEE Transactions on Wireless Communications*, vol. 22, no. 9, pp. 6227–6240, Sep. 2023. DOI: 10.1109/TWC.2023.3240969.

[24] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-Oriented Multi-User Semantic Communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022. DOI: 10.1109/JSAC.2022.3191326.

[25] B. Wang, R. Li, J. Zhu, Z. Zhao, and H. Zhang, "Knowledge Enhanced Semantic Communication Receiver," *IEEE Communications Letters*, vol. 27, no. 7, pp. 1794–1798, May 2023. DOI: 10.1109/LCOMM.2023.3274562.

[26] J. Shao, Y. Mao, and J. Zhang, "Learning Task-Oriented Communication for Edge Inference: An Information Bottleneck Approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 197–211, Jan. 2022. DOI: 10.1109/JSAC.2021.3126087.

[27] X. Luo, H.-H. Chen, and Q. Guo, "Semantic Communications: Overview, Open Issues, and Future Research Directions," *IEEE Transactions on Wireless Communications*, vol. 29, no. 1, pp. 210–219, Feb. 2022. DOI: 10.1109/MWC.101.2100269.

[28] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep Joint Source-Channel Coding for Wireless Image Transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, Sep. 2019. DOI: 10.1109/TCCN.2019.2919300.

[29] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *37th Annual Allerton Conference on Communication, Control and Computing (Allerton 1999)*, Sep. 1999, pp. 368–377. DOI: 10.48550/arXiv.physics/0004057.

[30] S. Hassanpour, T. Monsees, D. Wübben, and A. Dekorsy, "Forward-Aware Information Bottleneck-Based Vector Quantization for Noisy Channels," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7911–7926, 2020. DOI: 10.1109/TCOMM.2020.3019447.

[31] S. Hassanpour, D. Wübben, and A. Dekorsy, "Forward-Aware Information Bottleneck-Based Vector Quantization: Multiterminal Extensions for Parallel and Successive Retrieval," *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6633–6646, Jul. 2021. DOI: 10.1109/TCOMM.2021.3097142.

[32] E. Beck, B.-S. Shin, S. Wang, T. Wiedemann, D. Shutin, and A. Dekorsy, "Swarm Exploration and Communications: A First Step towards Mutually-Aware Integration by Probabilistic Learning," *Electronics*, Swarm Communication, Localization and Navigation, vol. 12, no. 8, p. 1908, Apr. 2023. DOI: 10.3390/electronics12081908.

[33] I. E. Aguerri and A. Zaidi, "Distributed Variational Representation Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 120–138, Jan. 2021. DOI: 10.1109/TPAMI.2019.2928806.

[34] J. Shao, Y. Mao, and J. Zhang, "Task-Oriented Communication for Multidevice Cooperative Edge Inference," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 73–87, Jan. 2023. DOI: 10.1109/TWC.2022.3191118.

[35] O. Simeone, "A Brief Introduction to Machine Learning for Engineers," *Foundations and Trends® in Signal Processing*, vol. 12, no. 3-4, pp. 200–431, Aug. 2018. DOI: 10.1561/2000000102.

[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, no. 110, pp. 3371–3408, Dec. 2010.

[37] S. Hassanpour, "Source & Joint Source-Channel Coding Schemes Based on the Information Bottleneck Framework," Ph.D. dissertation, University of Bremen, Bremen, Germany, Aug. 2022, p. 196.

[38] M. Sana and E. C. Strinati, "Learning Semantics: An Opportunity for Effective 6G Communications," in *19th IEEE Annual Consumer Communications Networking Conference (CCNC 2022)*, Virtual Conference, Jan. 2022, pp. 631–636. DOI: 10.1109/CCNC49033.2022.9700645.

[39] N. Farsad, N. Shlezinger, A. J. Goldsmith, and Y. C. Eldar, "Data-Driven Symbol Detection Via Model-Based Machine Learning," in *IEEE Statistical Signal Processing Workshop (SSP 2021)*, Virtual Conference, Jul. 2021, pp. 571–575. DOI: 10.1109/SSP49050.2021.9513859.

[40] Z. Goldfeld and Y. Polyanskiy, "The Information Bottleneck Problem and its Applications in Machine Learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 19–38, May 2020. DOI: 10.1109/JSAIT.2020.2991561.

[41] A. Zaidi, I. Estella-Aguerri, and S. Shamai Shitz, "On the Information Bottleneck Problems: Models, Connections, Applications and Information Theoretic Views," *Entropy*, vol. 22, no. 2, p. 151, Feb. 2020. DOI: 10.3390/e22020151.

[42] B. M. Kurkoski and H. Yagi, "Quantization of Binary-Input Discrete Memoryless Channels," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4544–4552, 2014. DOI: 10.1109/TIT.2014.2327016.

[43] J. Lewandowsky and G. Bauch, "Information-Optimum LDPC Decoders Based on the Information Bottleneck Method," *IEEE Access*, vol. 6, pp. 4054–4071, 2018. DOI: 10.1109/ACCESS.2018.2797694.

[44] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep Variational Information Bottleneck," in *5th International Conference on Learning Representations (ICLR 2017)*, Toulon, France, Apr. 2017, pp. 1–19. DOI: 10.48550/arXiv.1612.00410.

[45] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "MINE: Mutual Information Neural Estimation," in *International Conference on Machine Learning (ICML 2018)*, Stockholm, Sweden, Jun. 2018. DOI: 10.48550/arXiv.1801.04062.

[46] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd. Hoboken, NJ, USA: Wiley-Interscience, Jul. 2006. DOI: 10.1002/047174882X.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *14th European Conference on Computer Vision (ECCV 2016)*, ser. Lecture Notes in Computer Science, Amsterdam, Netherlands, Oct. 2016, pp. 630–645. DOI: 10.1007/978-3-319-46493-0_38.

[49] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from https://www.tensorflow.org/, Nov. 2015. DOI: 10.5281/zenodo.4724125.

[50] E. Beck, *Semantic Information Transmission and Recovery (SINFONY) Software*, version v1.1.0, Zenodo, Jul. 2023. DOI: 10.5281/zenodo.8006567.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision (ICCV 2015)*, vol. 14, Santiago, Chile, Dec. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[52] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, and A. Keller, *Sionna: An Open-Source Library for Next-Generation Physical Layer Research*, arXiv preprint: 2203.11854, Mar. 2022. DOI: 10.48550/arXiv.2203.11854.

[53] E. Beck, C. Bockelmann, and A. Dekorsy, *Model-free Reinforcement Learning of Semantic Communication by Stochastic Policy Gradient*, arXiv preprint: 2305.03571, May 2023. DOI: 10.48550/arXiv.2305.03571.

Chapter 5

Publication 3 – Model-free Reinforcement Learning of Semantic Communication by Stochastic Policy Gradient

This chapter has been published in:

E. Beck, C. Bockelmann, and A. Dekorsy, "Model-free Reinforcement Learning of Semantic Communication by Stochastic Policy Gradient," in *1st IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2024)*, Stockholm, Sweden, May 2024, pp. 367–373. DOI: 10.1109/ICMLCN59089.2024.10625190

The simulation source code is available in [Bec24]. Further analyses and additional details for this publication are provided in Appendix B.

5.1 Abstract

Following the recent success of machine learning tools in wireless communications, the idea of semantic communication by Weaver from 1949 has gained attention. It breaks with Shannon's classic design paradigm by aiming to transmit the meaning, i.e., semantics, of a message instead of its exact version, allowing for information rate savings. In this work, we apply the Stochastic Policy Gradient (SPG) to design a semantic communication system by reinforcement learning, separating transmitter and receiver, and not requiring a known or differentiable channel model — a crucial step towards deployment in practice. Further, we motivate the use of SPG for both classic and semantic communication from the maximization of the mutual information between received and target variables. Numerical results show that our approach achieves comparable performance to a model-aware approach based on the reparametrization trick, albeit with a decreased convergence rate.

Index Terms

Semantic communication, wireless networks, information maximization, information bottleneck, machine learning, reinforcement learning, stochastic policy gradient, task-oriented.

5.2 Introduction

To meet the unprecedented needs of 6G communication efficiency in terms of data rate, latency, and power, attention has been drawn to semantic communication [1]–[4]. It aims to transmit the meaning of a message rather than its exact version, which has been the main focus of digital error-free system design so far [1]. Bao, Basu et al. [5] were the first to define semantic information sources and channels to tackle the semantic design by conventional approaches arguing for the generality of Shannon's theory not only for the technical level but for semantic level design as Weaver [1].

Recently, inspired by [1], [5] and the rise of Machine Learning (ML) in communications research, transformer-based Deep Neural Networks (DNNs) have been introduced to AutoEncoders (AEs) for text transmission to learn compressed hidden representations of semantic content, aiming to improve communication efficiency [6]. In [7], the authors suggest using semantic similarity as the objective function: As most semantic metrics are non-differentiable, they propose a self-critic Reinforcement Learning (RL) solution.

Both [6], [7] improve performance especially at low SNR compared to classical digital transmissions with [7] being slightly superior.

This paper builds on our idea from [4]: There, we define semantic communication as the data-reduced, reliable transmission of semantic sources and cast its design as an Information Bottleneck (IB) problem extending [5]. We apply our ML-based design Semantic INFOrmation TraNsmission and RecoverY (SINFONY) to a distributed multipoint scenario, communicating meaning from multiple image sources to a single receiver for semantic recovery. Numerical results show that SINFONY outperforms classical communication systems.

Semantic communication is a developing field: For a more in-depth survey, we refer the reader to, e.g., [2]–[4]. It remains still unclear how the approaches proposed so far can be implemented in practice which motivates the main contributions of this article:

- We apply the Stochastic Policy Gradient (SPG) to train a semantic communication system, i.e., RL-SINFONY, by RL. By this means, we separate transmitter and receiver, and do not require a known or differentiable channel model — a crucial step towards deployment in practice.

- Further, we derive the application of the SPG for both classic and semantic communication from maximization of the mutual information between target and received variables compared to [8].

- In particular, we investigate a task-oriented system model and a distributed application scenario with multiple sources and transmitters. By this means, our work distinguishes from the RL-based approach in [7] that was extended to handle non-differentiable channels at the time of writing.

- Further, the authors of [7] observed that training does not converge within their time limit to comparable results as the baseline approach in their setup for text transmission. We confirm the problem of slow convergence hinting at solution approaches and demonstrate feasibility in our scenario.

In the following, we revisit our theoretical framework from [4] in Sec. 5.3. For RL-based optimization, we introduce the SPG in Sec. 5.4. Finally, in Sec. 5.5 and 5.6, we provide one numerical example for SINFONY application from [4] and summarize the main results, respectively.

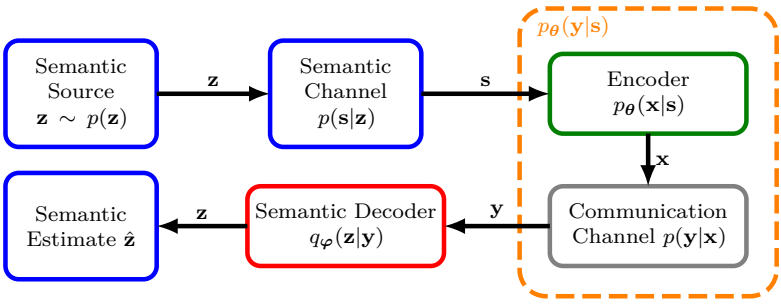

Figure 5.1: Block diagram of the considered semantic system model.

5.3 Semantic Communication Framework

5.3.1 Semantic System Model

Semantic Source and Channel

First, we define our information-theoretic system model of semantic communication shown in Fig. 5.1. Motivated by the approach of Bao, Basu et al. [5], we adopt the terminus of a semantic source as in [4] and describe it as a hidden target multivariate Random Variable (RV) $\mathbf{z} \in \mathcal{M}_z^{N_z \times 1}$ from domain $\mathcal{M}_z$ of dimension N_z distributed according to a probability density function (pdf) or probability mass function (pmf) $p(\mathbf{z})$. To simplify the discussion, we assume it to be discrete and memoryless.[1]

Then, a semantic channel modeled by conditional distribution $p(\mathbf{s}|\mathbf{z})$ generates an observation or source signal, a RV $\mathbf{s} \in \mathcal{M}_s^{N_s \times 1}$, that enters the communication system. Compared to [5] where the semantic channel is the transmission system, we consider probabilistic semantic channels $p(\mathbf{s}|\mathbf{z})$ using the definition from [4]. We refer the reader to [4] for an example of what these RVs may look like.

[1] For the remainder of the article, note that the domain of all RVs $\mathcal{M}$ may be either discrete or continuous. Further, we note that the definition of entropy for discrete and continuous RVs differs. For example, the differential entropy of continuous RVs may be negative whereas the entropy of discrete RVs is always positive [9]. Without loss of generality, we will thus assume all RVs either to be discrete or to be continuous. In this work, we avoid notational clutter by using the expected value operator: Replacing the integral by summation over discrete RVs, the equations are also valid for discrete RVs and vice versa.

Semantic Channel Encoding

Our challenge is to encode the source $\mathbf{s}$ onto the transmit signal $\mathbf{x} \in \mathcal{M}_x^{N_{\mathrm{Tx}} \times 1}$ (see Fig. 5.1) for efficient and reliable semantic transmission through the physical communication channel $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y} \in \mathcal{M}_y^{N_{\mathrm{Rx}} \times 1}$ is the received signal vector, such that the semantic RV $\mathbf{z}$ at a recipient is best preserved [4]. We parametrize the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ by a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}} \times 1}$ and assume $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ to be deterministic in communications with $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) = \delta(\mathbf{x} - \mu_{\boldsymbol{\theta}}(\mathbf{s}))$ and encoder function $\mu_{\boldsymbol{\theta}}(\mathbf{s})$. In summary, we bring the semantic source $\mathbf{z}$ to the context of communications by considering the complete Markov chain $\mathbf{z} \leftrightarrow \mathbf{s} \leftrightarrow \mathbf{x} \leftrightarrow \mathbf{y}$ in contrast to [5].

In classic Shannon design, the posterior $p_{\boldsymbol{\theta}}(\mathbf{s}|\mathbf{y})$ is processed to recover the observation $\mathbf{s}$ as accurately as possible at the receiver side. Instead, we recover semantics $\mathbf{z}$ processing $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$: Since the entropy $\mathcal{H}(\mathbf{z}) = \mathrm{E}_{\mathbf{z} \sim p(\mathbf{z})}[-\ln p(\mathbf{z})]$ of the semantic RV $\mathbf{z}$ is expected to be less or equal to the entropy $\mathcal{H}(\mathbf{s})$ of the source $\mathbf{s}$, i.e., $\mathcal{H}(\mathbf{z}) \leq \mathcal{H}(\mathbf{s})$, we can compress by transmitting the semantic RV $\mathbf{z}$. There, $\mathrm{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{x})]$ denotes the expected value of $f(\mathbf{x})$ w.r.t. both discrete or continuous RVs $\mathbf{x}$.

5.3.2 Semantic Communication Design

Now, we revisit our two design approaches from [4].

InfoMax Principle

First, we like to find the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ that maps $\mathbf{s}$ to a representation $\mathbf{y}$ such that most information of the relevant RV $\mathbf{z}$ is included in $\mathbf{y}$, i.e., we maximize the Mutual Information (MI) $I_{\boldsymbol{\theta}}(\mathbf{z}; \mathbf{y})$ w.r.t. $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$:

$$\begin{aligned}
&\arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} I_{\boldsymbol{\theta}}(\mathbf{z}; \mathbf{y}) && (5.1)\\
=&\arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} \mathrm{E}_{\mathbf{z},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}\left[\ln \frac{p_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{y})}{p(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{y})}\right] && (5.2)\\
=&\arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} \mathcal{H}(\mathbf{z}) - \mathcal{H}(p_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{y}), p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})) && (5.3)\\
=&\arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} \mathrm{E}_{\mathbf{z},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})] \,. && (5.4)
\end{aligned}$$

There, $\mathcal{H}(p(\mathbf{x}), q(\mathbf{x})) = \mathrm{E}_{\mathbf{x} \sim p(\mathbf{x})}[-\ln q(\mathbf{x})]$ is the cross-entropy between two pdfs/pmfs $p(\mathbf{x})$ and $q(\mathbf{x})$.

If the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$ in (5.4) is intractable to compute, we can replace it with a variational distribution $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ with parameters $\boldsymbol{\varphi} \in \mathbb{R}^{N_{\boldsymbol{\varphi}} \times 1}$, i.e.,

the semantic decoder in Fig. 5.1. Then, we can define a MI Lower BOund (MILBO) [4]:

$$I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y}) \geq \mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \tag{5.5}$$

$$= -\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y})}[\mathcal{H}(p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}), q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y}))] \tag{5.6}$$

$$= -\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}\,. \tag{5.7}$$

Now, we can learn optimal parametrizations $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ of the transmitter discriminative model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ and of the variational receiver posterior $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ by minimizing the amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ in (5.6), i.e., marginalized across received signals $\mathbf{y}$ [4]. The encoder can be seen by rewriting:

$$\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} = \mathrm{E}_{\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{z}\sim p_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{x},\mathbf{y},\mathbf{z})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \tag{5.8}$$

$$= \mathrm{E}_{\mathbf{s},\mathbf{z}\sim p(\mathbf{s},\mathbf{z})}\left[\mathrm{E}_{\mathbf{x}\sim p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}\left[\mathrm{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})]\right]\right].$$

The idea is to solve (5.8) by AEs or — in this article — RL. Thus, we use DNNs for the design of both encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ and decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ [6].

Note that in our semantic problem (5.1) or (5.8), we do not auto encode the hidden $\mathbf{z}$ or $\mathbf{s}$ as in [6] itself, but encode $\mathbf{s}$ to obtain $\mathbf{z}$ by decoding. This means our interpretation of semantic information and its recovery deviates from literature: We define semantics $\mathbf{z}$ explicitly compared to, e.g., [6], that optimizes on $\mathbf{s}$ and then measures semantic similarity w.r.t. its estimate $\hat{\mathbf{s}}$ explicitly by some semantic metric $\mathcal{L}(\mathbf{s},\hat{\mathbf{s}})$.

Information Bottleneck View

Further, introducing a constraint on the information rate in (5.1), we can formulate an Information Bottleneck (IB) optimization problem [2], where we like to maximize the relevant information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$ subject to the constraint to limit the compression rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ to a maximum information rate I_{C}:

$$\underset{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\arg\max}\; I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y}) \quad \text{s.t.} \quad I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y}) \leq I_{\mathrm{C}}\,. \tag{5.9}$$

In this article, we set constraint I_{C} by fixing N_{Tx} since then an upper bound on $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ grows as shown in [4]. With fixed constraint I_{C}, we then need to maximize the relevant information $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$. As in the InfoMax problem, we can exploit the MILBO to use the amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ in (5.8) as the optimization criterion.

5.4 Stochastic Policy Gradient-based Reinforcement Learning

If calculating the expected value of the amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ in (5.8) is analytically or computationally intractable as typical with DNNs, we can approximate it using Monte Carlo sampling techniques with N samples $\{(\mathbf{z}_i, \mathbf{s}_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$.

5.4.1 Stochastic Gradient Descent-based Optimization

For Stochastic Gradient Descent (SGD)–based optimization, the gradient w.r.t. $\boldsymbol{\varphi}$ can then be calculated by

$$\frac{\partial \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}}{\partial \boldsymbol{\varphi}} = -\,\mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})p(\mathbf{s}|\mathbf{z})p(\mathbf{z})}\left[\frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})}{\partial \boldsymbol{\varphi}}\right] \tag{5.10}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}\frac{\partial\left[-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)\right]}{\partial \boldsymbol{\varphi}} \tag{5.11}$$

with N being equal to the batch size N_{b} and by application of the backpropagation algorithm in Automatic Differentiation Framework (ADF), e.g., TensorFlow or PyTorch.

Reinforce Gradient

Computing the gradient w.r.t. $\boldsymbol{\theta}$ is not straightforward since we sample w.r.t. the distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ dependent on $\boldsymbol{\theta}$ [9]. For continuous-valued $\mathbf{y}$ and using the log-trick $\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})}{\partial \boldsymbol{\theta}} = \frac{\partial p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})}{\partial \boldsymbol{\theta}} / p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$, we derive:

$$\frac{\partial \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}}{\partial \boldsymbol{\theta}}$$

$$= -\frac{\partial}{\partial \boldsymbol{\theta}}\,\mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})p(\mathbf{s},\mathbf{z})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \tag{5.12}$$

$$= -\,\mathrm{E}_{\mathbf{z},\mathbf{s}\sim p(\mathbf{s},\mathbf{z})}\left[\int_{\mathcal{M}_y^{N_{\mathrm{Rx}}}} \underbrace{\frac{\partial p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})}{\partial \boldsymbol{\theta}}}_{=p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})\cdot\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})}{\partial \boldsymbol{\theta}}} \cdot \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\;\mathrm{d}\mathbf{y}\right] \tag{5.13}$$

$$= -\,\mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})p(\mathbf{s},\mathbf{z})}\left[\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})}{\partial \boldsymbol{\theta}} \cdot \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right] \tag{5.14}$$

$$\approx -\frac{1}{N}\sum_{i=1}^{N}\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{s}_i)}{\partial \boldsymbol{\theta}} \cdot \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)\,. \tag{5.15}$$

We arrive at the same result with discrete RVs $\mathbf{y}$ replacing the integral in (5.13) by a sum. The Monte Carlo approximation (5.15) is the Reinforce gradient w.r.t. $\boldsymbol{\theta}$ [9]. This estimate has high variance since we sample w.r.t. the distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ dependent on $\boldsymbol{\theta}$.

Reparametrization Trick

Leveraging the direct relationship between $\boldsymbol{\theta}$ and $\mathbf{y}$ in $\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ can help reduce the estimator's high variance. Typically, e.g., in Variational AutoEncoders (VAEs), the reparametrization trick is used to achieve this [9]. Here we can apply it if we can decompose the latent variable $\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ into a differentiable function $\mathbf{y} = f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{n})$ and a RV $\mathbf{n} \sim p(\mathbf{n})$ independent of $\boldsymbol{\theta}$. Fortunately, the typical forward model of a communication system $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ fulfills this criterion. Assuming a deterministic (DNN) encoder $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$ and additive noise $\mathbf{n}$ with covariance $\boldsymbol{\Sigma}$, we can thus rewrite $\mathbf{y}$ into $f_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{n}) = \mu_{\boldsymbol{\theta}}(\mathbf{s}) + \boldsymbol{\Sigma}^{1/2} \cdot \mathbf{n}$ and accordingly the amortized cross-entropy gradient (5.12) into:

$$\frac{\partial \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}}{\partial \boldsymbol{\theta}} = -\,\mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{n}\sim p(\mathbf{n})p(\mathbf{s}|\mathbf{z})p(\mathbf{z})}\left[\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{n})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})}{\partial \mathbf{y}}\right] \tag{5.16}$$

$$\approx -\frac{1}{N}\sum_{i=1}^{N} \frac{\partial f_{\boldsymbol{\theta}}(\mathbf{s}_i,\mathbf{n}_i)}{\partial \boldsymbol{\theta}} \cdot \left.\frac{\partial \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y})}{\partial \mathbf{y}}\right|_{\mathbf{y}=f_{\boldsymbol{\theta}}(\mathbf{s}_i,\mathbf{n}_i)}. \tag{5.17}$$

The trick can be easily implemented in ADFs by adding a noise layer after function $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$, typically used for regularization in ML literature. Then, our loss function (5.8) is the empirical cross-entropy:

$$\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}} \approx -\frac{1}{N}\sum_{i=1}^{N} \ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i = f_{\boldsymbol{\theta}}(\mathbf{s}_i,\mathbf{n}_i))\,. \tag{5.18}$$

This allows for joint learning of both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$, as demonstrated in recent works [4], [10], treating unsupervised optimization of AEs and SINFONY as a supervised learning problem.

5.4.2 Stochastic Policy Gradient

We note that optimization of encoder and decoder with both gradients (5.15) or (5.17) requires model-awareness, i.e., a known and differentiable forward model $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$. But the gradient

$$\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})}{\partial \boldsymbol{\theta}} = \frac{\partial \mu_{\boldsymbol{\theta}}(\mathbf{s})}{\partial \boldsymbol{\theta}} \cdot \frac{\partial p(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x}} \cdot \frac{\partial \ln p(\mathbf{y}|\mathbf{x})}{\partial p(\mathbf{y}|\mathbf{x})} \tag{5.19}$$

with deterministic encoder $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$ may not be computable, as the channel model $p(\mathbf{y}|\mathbf{x})$ could be non-differentiable or unknown without any channel estimate. Further, in practice, the transmitter and receiver are separated at different locations and have at most a rudimentary feedback link, requiring independent optimization w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$: The transmitter does not know $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ and the receiver $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$, vice versa.

To tackle these challenges in gradient computation, we now introduce a stochastic policy $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) \neq \delta(\mathbf{x} - \mu_{\boldsymbol{\theta}}(\mathbf{s}))$ that fulfills the reparametrization property:

$$\frac{\partial \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}}{\partial \boldsymbol{\theta}} = -\frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{x},\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})p(\mathbf{s},\mathbf{z})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \tag{5.20}$$

$$= -\mathrm{E}_{\mathbf{z},\mathbf{s}\sim p(\mathbf{s},\mathbf{z})}\left[\int_{\mathcal{M}_x^{N_{\mathrm{Tx}}}} \underbrace{\frac{\partial p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\partial \boldsymbol{\theta}}}_{=p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})\cdot\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\partial \boldsymbol{\theta}}} \cdot \mathrm{E}_{\mathbf{y}\sim p(\mathbf{y}|\mathbf{x})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] \ \mathrm{d}\mathbf{x}\right] \tag{5.21}$$

$$= -\mathrm{E}_{\mathbf{z},\mathbf{s},\mathbf{x},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{s},\mathbf{x},\mathbf{y})}\left[\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})}{\partial \boldsymbol{\theta}} \cdot \ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\right] \tag{5.22}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} \frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{s}_i)}{\partial \boldsymbol{\theta}} \cdot [-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)] \ . \tag{5.23}$$

Again the log-trick is applied in (5.21) to arrive in (5.22) and the results hold for discrete RVs $\mathbf{x}$. Most importantly, (5.22) is the policy gradient and the derivation is equivalent to the Stochastic Policy Gradient (SPG) theorem, a fundamental result of continuous-action RL [11]. For integration into ADFs, usually, an objective function whose gradient is the Monte Carlo policy gradient estimator of (5.22), i.e., the Reinforce gradient (5.23), is constructed:

$$\mathcal{L}^{\mathrm{SPG}}_{\boldsymbol{\theta}} = \frac{1}{N}\sum_{i=1}^{N} \ln p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{s}_i) \cdot [-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)] \ . \tag{5.24}$$

With objective (5.24) or Reinforce gradient (5.23), we can finally optimize $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ w.r.t. $\boldsymbol{\theta}$, since we can sample $\{(\mathbf{z},\mathbf{s},\mathbf{x},\mathbf{y})\} \sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{s},\mathbf{x},\mathbf{y})$ and compute $\frac{\partial \ln p_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{s}_i)}{\partial \boldsymbol{\theta}}$ at the transmitter and $-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)$ being equal to the per-sample cross-entropy at the receiver.

Note that $\mathbf{s}_i$ and $\mathbf{x}_i$ only have to be known at the transmitter and both $\mathbf{z}_i$ and $\mathbf{y}_i$ at the receiver, respectively. This enables the separation or spatial distribution of transmitter and receiver when the following conditions are met:

- Only an a priori known pilot sequence $\mathcal{D}_\mathrm{P} = \{(\mathbf{z}_i, \mathbf{s}_i)\}_{i=1}^{N_\mathrm{pilot}}$ of size N_pilot is required. This sequence translates into the training set $\mathcal{D}_\mathrm{T} = \{(\mathbf{z}_i, \mathbf{s}_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_\mathrm{train}}$ of size N_train which is divided into batches of size N_b for SGD-based optimization.
- Moreover, we require a feedback link to transmit the per-sample cross-entropy $-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)$ to the encoder. This term can be interpreted as a reward or critic known from RL [11]. Accordingly, the transmitter can be seen as an actor with a policy $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$. The best continuous action/policy is then learned by optimization w.r.t. these rewards.

Stochastic Policy

Introducing a stochastic policy means we need to add a probabilistic sampler/explorer function $p(\mathbf{x}|\bar{\mathbf{x}})$ to the encoder as shown in Fig. 5.2. Replacing $p(\mathbf{y}|\mathbf{s})$ and $p(\mathbf{y}|\mathbf{x})$ by $p(\mathbf{x}|\mathbf{s})$ and $p(\mathbf{x}|\bar{\mathbf{x}})$ in (5.19) and applying the result to (5.23), we derive that this function needs to be differentiable. If the encoder output, i.e., the action space, is continuous with $\mathcal{M}_x = \mathbb{R}$, we can achieve this using for example a Gaussian policy, i.e., a multivariate Gaussian pdf

$$p(\mathbf{x}|\bar{\mathbf{x}}) = p\left(\mathbf{x}|\bar{\mathbf{x}}, \sigma^2_\mathrm{exp}\right) = \mathcal{N}\left(\left(1-\sigma^2_\mathrm{exp}\right)^{1/2} \cdot \bar{\mathbf{x}}, \sigma^2_\mathrm{exp} \cdot \mathbf{I}\right) \tag{5.25}$$

with exploration variance $\sigma^2_\mathrm{exp} \in (0,1)$ where scaling of the mean $\bar{\mathbf{x}} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$ is done to ensure the conservation of average energy. Furthermore, the Gaussian policy offers the benefit of simplicity in parametrization, requiring tuning of only two pdf parameters. Hence, we employ it in our numerical experiments. For discrete action spaces $\mathcal{M}_x^{N_\mathrm{Tx}\times 1}$, a continuous differentiable relaxation such as the Gumbel Softmax is required [12].

In the special case $\sigma^2_\mathrm{exp} \to 0$, the Gaussian policy $p(\mathbf{x}|\bar{\mathbf{x}}, \sigma^2_\mathrm{exp})$ approaches a deterministic policy. In [8], the authors show that the true channel gradient $\frac{\partial}{\partial \mathbf{x}} p(\mathbf{y}|\mathbf{x})$ is then perfectly approximated. However, using a near-deterministic policy leads in their experiments to high variance of the gradient estimate (5.23) resulting in slow convergence. To compensate for this effect, we require a much larger and computationally expensive batch size $N = N_\mathrm{b}$. From the view of RL, using a stochastic policy with $\sigma^2_\mathrm{exp} \neq 0$ enables the exploration of the set of possible actions.

5.4.3 Alternating RL-based Training

After introducing the SPG, we now derive an optimization procedure akin to [8] for the whole semantic communication system. It does not require

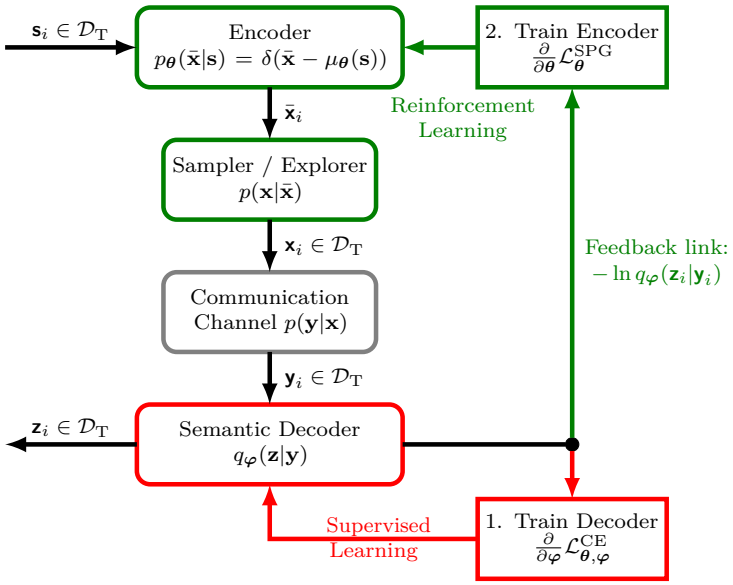

Figure 5.2: Optimization procedure of a semantic encoder and decoder without a differentiable channel model: 1. Train the decoder supervised based on the training sequence and updated encoder but without sampler. 2. Encoder explores transmit signals $\mathbf{x}_i$ and improves its policy according to the decoder reward feedback. 3. Alternate between both steps until convergence.

any channel model but a fixed pilot, i.e., training, sequence and a feedback link. Further, it enables separation of encoder and decoder. We show it in Fig. 5.2:

1. We note that according to (5.11) decoder optimization reduces to supervised learning w.r.t. $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\varphi}}^{\mathrm{CE}}$ and $\boldsymbol{\varphi}$ at the receiver side. Thus, in the first step, we train the decoder based on the training sequence and updated encoder, but without sampler/explorer ($\sigma_{\mathrm{exp}}^2 = 0$).

2. Second, the encoder explores with transmit signals $\mathbf{x}_i$. It is optimized based on the policy gradient of $\mathcal{L}_{\boldsymbol{\theta}}^{\mathrm{SPG}}$ and the reward $-\ln q_{\boldsymbol{\varphi}}(\mathbf{z}_i|\mathbf{y}_i)$ that the decoder feeds back.

3. We alternate between the first and second training steps until convergence. Note that we can use one or multiple SGD steps and batches for each alternating training step, respectively.

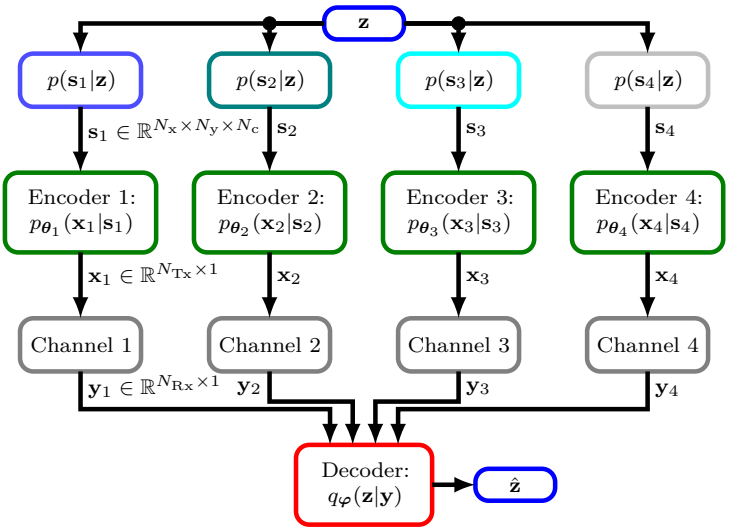

Figure 5.3: RL-SINFONY scenario: Four distributed agents extract features for rate-efficient transmission to a decoder that extracts semantics.

Reminiscent of the RL fashion of the stochastic policy optimization of semantic information transmission and recovery [4], we name this approach RL-SINFONY. Finally, we have derived the SPG for semantic communication starting from the InfoMax problem (5.1). Replacing $I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y})$ by $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$, this result can be generalized to also hold for classic communications.

5.5 Example of Model-free Semantic Recovery

To evaluate the proposed model-free optimization approach RL-SINFONY, we use the numerical example of distributed image classification with SINFONY from [4] shown in Fig. 5.3. Thus, we will now assume the hidden semantic RV to be a one-hot vector $\mathbf{z} \in \{0,1\}^{M \times 1}$ representing one of M image classes. Then, each of the four agents observes its image, i.e., the observation $\mathbf{s}_i \sim p(\mathbf{s}_i|\mathbf{z})$ with $i = 1, \ldots, 4$, through a semantic channel, being generated by the same semantic RV $\mathbf{z}$ and thus belonging to the same class. Based on these images, a central unit shall extract semantics, i.e., perform classification.

We propose to optimize the four encoders $p_{\boldsymbol{\theta}_i}(\mathbf{x}_i|\mathbf{s}_i)$ **jointly** with a decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4]^T)$ w.r.t. cross-entropy (5.8) of the semantic labels (see Fig. 5.3). Hence, we maximize the system's overall semantic measure, i.e.,

classification accuracy.

To show the basic working principle and ease implementation, we use the grayscale MNIST and colored CIFAR10 datasets with $M = 10$ image classes [4]. We assume that the semantic channel generates an image that we divide into four equally sized quadrants and each agent observes one quadrant $\mathbf{s}_i \in \mathbb{R}^{N_\mathrm{x} \times N_\mathrm{y} \times N_\mathrm{c}}$ where N_x and N_y is the number of image pixels in the x- and y-dimension, respectively, and N_c is the color channel number.

5.5.1 Distributed SINFONY Approach

For the design of SINFONY, we rely on the powerful DNN approach ResNet for feature extraction [4]. We use the pre-activation version of ResNet without bottlenecks implemented for CIFAR10 classification. In Tab. 5.1, we show its structure modified for the distributed scenario from Fig. 5.3. There, ResNetBlock is the basic building block of the ResNet architecture. Each block consists of multiple residual unit (res. un.) and we use 2 for the MNIST and 3 for the CIFAR10 dataset. For further implementation details, we refer the reader to the original work [4] and our source code [13].

Our key idea here is to modify ResNet w.r.t. the communication task by splitting it where a low-bandwidth representation of semantic information is present. Therefore, we aim to transmit each agent's local features of length N_Feat provided by the Feature Extractors in Tab. 5.1 instead of all sub-images $\mathbf{s}_i$ and add the component Tx to encode the features into $\mathbf{x}_i \in \mathbb{R}^{N_\mathrm{Tx} \times 1}$ for transmission through the wireless channel (see Fig. 5.3). We note that $\mathbf{x}_i \in \mathbb{R}^{N_\mathrm{Tx} \times 1}$ is analog and that the output dimension N_Tx defines the number of channel uses per agent and thus information rate. To limit the transmit power to one, we constrain the Tx Linear layer output by the norm along the training batch or the encode vector dimension (dim.).

For RL-SINFONY, we add a Gaussian Sampler (5.25) after the Tx output compared to [4]. Further, we assume all agents and the Rx module to share a training set $\mathcal{D}_\mathrm{T}$ and a perfect reward feedback link from the Rx module to all agents.

At the receiver side, we use a single Rx module only with shared DNN layers of width N_w and parameters $\boldsymbol{\varphi}_\mathrm{Rx}$ for all inputs $\mathbf{y}_i$ [4]. Based on an aggregation of the four Rx outputs, a softmax layer with $M = 10$ units finally computes class probabilities $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ whose maximum is the maximum a posteriori estimate $\hat{\mathbf{z}}$.

5.5.2 Optimization Details

We evaluate RL-SINFONY in TensorFlow 2 on the MNIST and CIFAR10 datasets with training set $\mathcal{D}_\mathrm{T}$ [13]. For cross-entropy loss minimization, we

Table 5.1: RL-SINFONY–DNN architecture for image example.

Component	Layer	Dimension
Input	Image (MNIST, CIFAR10)	$(14, 14, 1)$, $(16, 16, 3)$
$4\times$	Conv2D	$(14, 14, 14)$, $(16, 16, 16)$
Feature	ResNetBlock (2/3 res. un.)	$(14, 14, 14)$, $(16, 16, 16)$
Extractor	ResNetBlock (2/3 res. un.)	$(7, 7, 28)$, $(8, 8, 32)$
	ResNetBlock (2/3 res. un.)	$(4, 4, 56)$, $(4, 4, 64)$
	Batch Normalization	$(4, 4, 56)$, $(4, 4, 64)$
	ReLU activation	$(4, 4, 56)$, $(4, 4, 64)$
	GlobalAvgPool2D	(56), (64)
$4\times$ Tx	ReLU	N_{Tx}
	Linear	N_{Tx}
	Normalization (dim.)	N_{Tx}
$4\times$ Sampler	AWGN + Normalization	N_{Tx}
$4\times$ Channel	AWGN	N_{Tx}
Rx	ReLU ($4\times$ shared)	$(2, 2, N_{\mathrm{w}})$
	GlobalAvgPool2D	N_{w}
Classifier	Softmax	$M = 10$

use the gradient approximations from Sec. 5.4 and the SGD-variant Adam with a batch size of $N_{\mathrm{b}} = 500$. We add l_2-regularization with a weight decay of 0.0001. To optimize the transceiver for a wider SNR range, we choose the SNR to be uniformly distributed within $[-4, 6]$ dB where $\mathrm{SNR} = 1/\sigma_{\mathrm{n}}^2$ with noise variance σ_{n}^2. We set $N_{\mathrm{w}} = N_{\mathrm{Feat}}$ as default and refer to [4], [13] for more implementation details. In the following, we compare the performance of[2]:

- **Digital comm.:** Digital transmission baseline from [4] with capacity achieving LDPC code and ResNet classifier.

[2]It is not straightforward to compare the approach from [7] with RL-SINFONY as different models were investigated. We leave a detailed comparison with other approaches from the literature for future work.

- **SINFONY:** The distributed SINFONY design from [4] trained *model-aware* as one DNN with channel noise layer using the reparametrization trick (5.17) to approximate the gradients. We train for $N_{\mathrm{e}} = 100$ epochs with the MNIST dataset.

- **RL-SINFONY:** New approach trained *model-free* via RL as shown in Fig. 5.2 using SPG (5.23). We alternate between 10 decoder and encoder optimization steps. Note that one decoder and encoder step amounts to one iteration of the model-aware approach where the encoder and decoder are optimized jointly. Hence, for a fair comparison, we divide the number of alternating iterations or epochs N_{e} of the SPG approach by 2. We choose $N_{\mathrm{e}} = 3000$ and add $N_{\mathrm{e,rx}} = 600$ epochs of receiver fine-tuning at the end [8]. To decrease the SPG estimator variance, we choose a rather high exploration variance $\sigma^2_{\mathrm{exp}} = 0.15$.

- **Perfect comm.:** SINFONY trained with perfect communication links without Tx and Rx modules, but with Tx normalization. Thus, the plain power-constrained features are transmitted with $N_{\mathrm{Tx}} = 56$ or 64 channel uses. It serves as the benchmark, as it indicates the maximum performance of the distributed design.

- **Tx/Rx N_{Tx}:** Default SINFONY from Tab. 5.1 trained with Tx and Rx module and N_{Tx} channel uses.

5.5.3 Numerical Results

To measure semantic transmission quality, we use classification error rate on semantic RV $\mathbf{z}$ and normalize the SNR by the spectral efficiency $\eta = N_{\mathrm{Feat}}/N_{\mathrm{Tx}}$ [4].

MNIST dataset

The numerical results of our proposed approach RL-SINFONY on the MNIST validation dataset are shown in Fig. 5.4. We observe that both approaches RL-SINFONY and SINFONY with Tx/Rx module approach the benchmark with ideal links (SINFONY – Perfect comm.) at high SNR and beat Digital comm. w.r.t. communication efficiency. Notably, both curves are very close to each other, i.e., the performance gap after training is minor. This means training of RL-SINFONY converged successfully. Note that Digital comm. classifies the entire image at once and thus outperforms at high SNR [4].

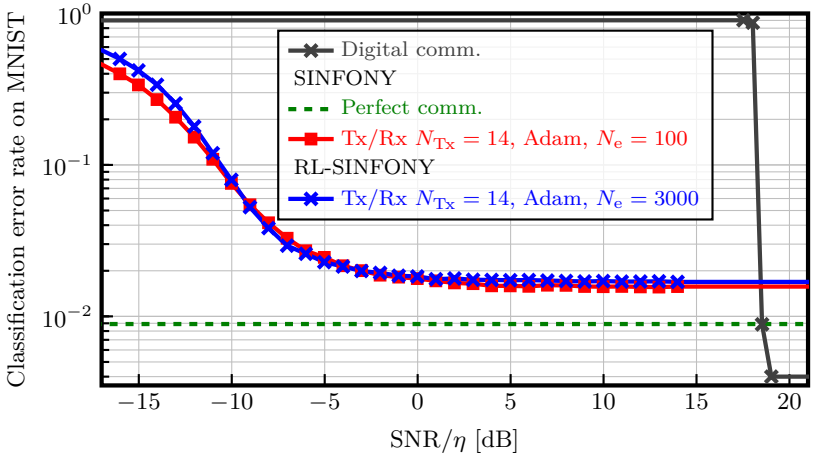

Figure 5.4: Comparison of the classification error rate of RL-SINFONY and SINFONY with $N_{\mathrm{Tx}} = 14$ on MNIST as a function of normalized SNR.

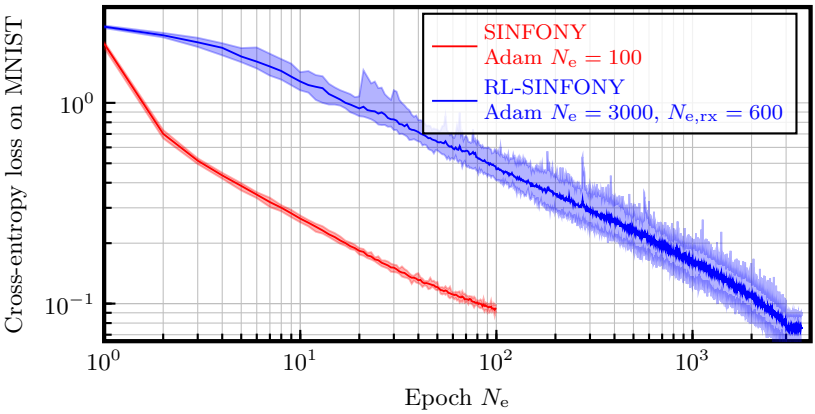

Figure 5.5: Comparison of training convergence between RL-SINFONY and SINFONY with $N_{\mathrm{Tx}} = 14$ in terms of the cross-entropy loss on MNIST averaged over 10 runs as a function of training epochs N_{e}.

Convergence Rate

Since the number of training epochs required to achieve the same performance deviates significantly with $N_\mathrm{e} + N_\mathrm{e,rx} = 3000 + 600 = 3600$ compared to $N_\mathrm{e} = 100$, we take a closer look at training convergence in terms of the cross-entropy loss shown in Fig. 5.5. We averaged the loss over 10 training runs and illustrate the interval between the maximum and minimum loss value using shaded areas. To reach the same loss, we require more than 10 times more epochs with RL-SINFONY compared to SINFONY. The reason for the decreased convergence is the increased variance of the Reinforce gradient (5.23) compared to the reparametrization trick gradient (5.17). Further, we attribute the increased variance in training losses (blue-shaded area) to it.

CIFAR10 dataset and convergence issues

We further evaluate RL-SINFONY on the more challenging CIFAR10 validation dataset with $N_\mathrm{Tx} = 16$ and fine-tuned learning rate $\epsilon = 10^{-4}$. The performance curves of SINFONY and RL-SINFONY with Adam, depicted in Fig. 5.6, closely align, affirming the effectiveness of RL-SINFONY.

Nevertheless, it is crucial to highlight that training with Adam does not converge to a local minimum with the same 80% validation accuracy achieved by the SINFONY benchmark in [4]. In that work, we utilized SGD with a batch size of $N_\mathrm{b} = 64$, ran for $N_\mathrm{e} = 200$ epochs, and employed a dedicated learning rate schedule. Despite exploring various hyperparameter settings, replicating the same performance with RL-SINFONY has proven elusive.

Additionally, we observed that the training of RL-SINFONY on the CIFAR10 dataset exhibits slow convergence. For example, using SGD with $N_\mathrm{b} = 128$ and $\epsilon = 10^{-4}$ (see Fig. 5.6), we achieve a validation accuracy of 50% at high SNR after $N_\mathrm{e} + N_\mathrm{e,rx} = 5000 + 1000 = 6000$ epochs, still gradually improving to a maximum of 60% after an extensive training period of $N_\mathrm{e} + N_\mathrm{e,rx} = 50000 + 10000 = 60000$ epochs.

We assume the slow convergence to be caused by the high variance of the Reinforce gradient (5.23), which increases by decreasing σ^2_exp and increasing the continuous output space N_Tx of $\mathbf{x}$. Training with the more challenging CIFAR10 dataset may require more accurate gradient estimates compared to MNIST. Thus, we suggest exploring variance-reduction techniques in future work [9], [14]. Note that, analogous to the mean $\bar{\mathbf{x}} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$ of the Gaussian policy (5.25), also the exploration variance σ^2_exp can be parametrized by a DNN with shared parameters $\boldsymbol{\theta}$ or independent parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_{\bar{\mathbf{x}}}, \boldsymbol{\theta}_{\sigma^2_\mathrm{exp}}]^T$. Both approaches could facilitate quicker convergence and more efficient hyperparameter tuning, ultimately leading to higher validation accuracy.

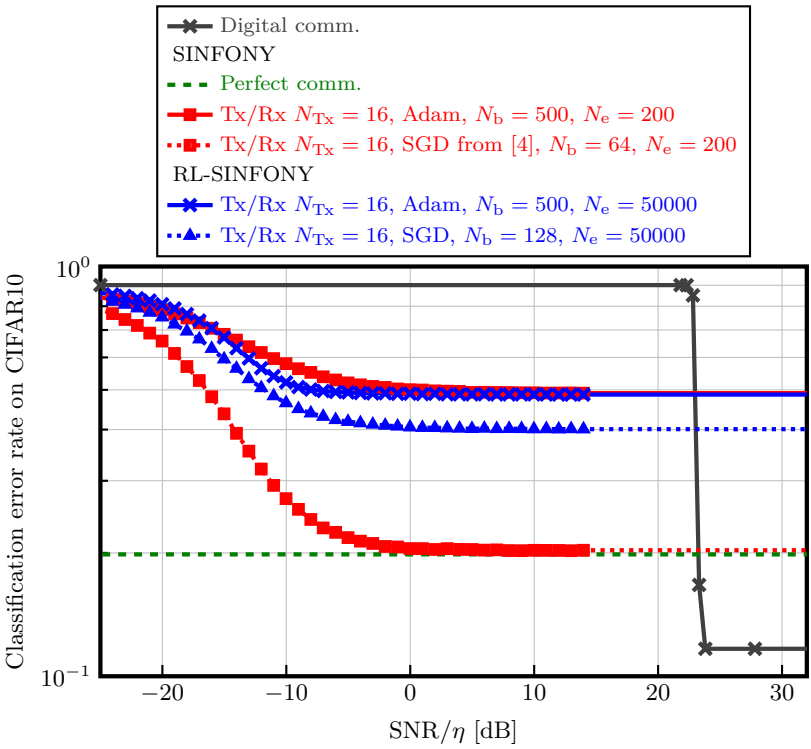

Figure 5.6: Comparison of the classification error rate of RL-SINFONY and SINFONY with $N_{\mathrm{Tx}} = 16$ on CIFAR10 as a function of normalized SNR.

5.6 Conclusion

In this work, we expanded on our previous idea from [4] by introducing the Stochastic Policy Gradient (SPG): We designed a semantic communication system via reinforcement learning, separating transmitter and receiver, and not requiring a known or differentiable channel model — a crucial step towards deployment in practice. Further, we derived the use of the SPG for both classic and semantic communication from the maximization of the Mutual Information (MI) between received and target variables. Numerical results show that our approach achieves comparable performance to a model-aware approach, albeit at the cost of a decreased convergence rate by at least a factor of 10. It remains the question of how to improve the convergence

rate with more challenging datasets.

5.7 References

[1] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, 16th ed. Urbana, IL, USA: The University of Illinois Press, Sep. 1949.

[2] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107 930, May 2021. DOI: 10.1016/j.comnet.2021.107930.

[3] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, Jan. 2023. DOI: 10.1109/JSAC.2022.3223408.

[4] E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic Information Recovery in Wireless Networks," *Sensors*, vol. 23, no. 14, p. 6347, Jul. 2023. DOI: 10.3390/s23146347.

[5] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *IEEE Network Science Workshop (NSW 2011)*, West Point, NY, USA, Jun. 2011, pp. 110–117. DOI: 10.1109/NSW.2011.6004632.

[6] 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, Apr. 2021. DOI: 10.1109/TSP.2021.3071210.

[7] K. Lu, R. Li, X. Chen, Z. Zhao, and H. Zhang, *Reinforcement Learning-powered Semantic Communication via Semantic Similarity*, arXiv preprint: 2108.12121, Apr. 2022. DOI: 10.48550/arXiv.2108.12121.

[8] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019. DOI: 10.1109/JSAC.2019.2933891.

[9] O. Simeone, "A Brief Introduction to Machine Learning for Engineers," *Foundations and Trends® in Signal Processing*, vol. 12, no. 3-4, pp. 200–431, Aug. 2018. DOI: 10.1561/2000000102.

[10] 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, Dec. 2017. DOI: 10.1109/TCCN.2017.2758370.

[11] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *31st International Conference on Machine Learning (ICML 2014)*, Beijing, China, Jan. 2014, pp. 387–395.

[12] E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection With Low Complexity," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8214–8227, Dec. 2021. DOI: 10.1109/TCOMM.2021.3114682.

[13] E. Beck, *Semantic Information Transmission and Recovery (SINFONY) Software*, version v1.1.0, Zenodo, Jul. 2023. DOI: 10.5281/zenodo.8006567.

[14] E. Greensmith, P. L. Bartlett, and J. Baxter, "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning," *Journal on Machine Learning Research*, vol. 5, pp. 1471–1530, Dec. 2004.

Chapter 6

Publication 4 – Integrating Semantic Communication and Human Decision-Making into an End-to-End Sensing-Decision Framework

This chapter was under review at the time of submission and is available as a preprint in:

E. Beck, H.-Y. Lin, P. Rückert, Y. Bao, B. von Helversen, S. Fehrler, K. Tracht, and A. Dekorsy, *Integrating Semantic Communication and Human Decision-Making into an End-to-End Sensing-Decision Framework*, arXiv preprint: 2412.05103, Mar. 2025. DOI: 10.48550/arXiv.2412.05103

The simulation source code is partly available in [Bec24].

6.1 Abstract

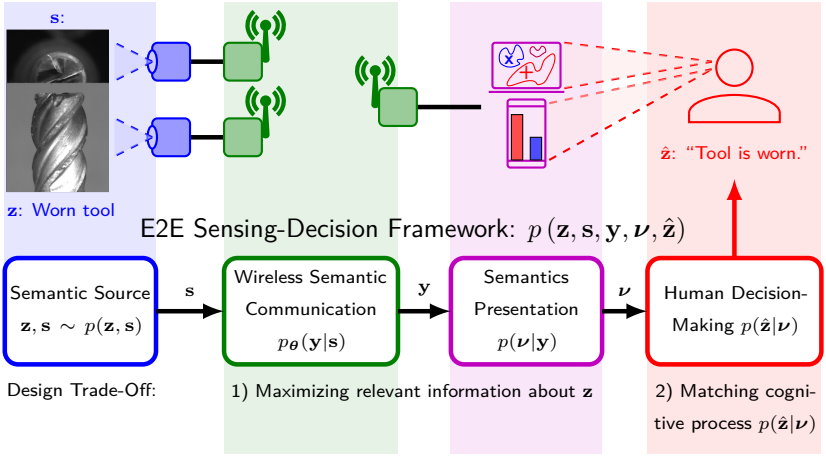

Figure 6.1: Graphical Abstract of [BLR+25].

As early as 1949, Weaver defined communication in a very broad sense to include all procedures by which one mind or technical system can influence another, thus establishing the idea of semantic communication. With the recent success of machine learning in expert assistance systems where sensed information is wirelessly provided to a human to assist task execution, the need to design effective and efficient communications has become increasingly apparent. In particular, semantic communication aims to convey the meaning behind the sensed information relevant for Human Decision-Making (HDM). Regarding the interplay between semantic communication and HDM, many questions remain, such as how to model the entire end-to-end sensing-decision-making process, how to design semantic communication for the HDM and which information should be provided to the HDM. To address these questions, we propose to integrate semantic communication and HDM into one probabilistic end-to-end sensing-decision framework that bridges communications and psychology. In our interdisciplinary framework, we model the human through a HDM process, allowing us to explore how feature extraction from semantic communication can best support HDM both in theory and in simulations. In this sense, our study reveals the fundamental design trade-off between maximizing the relevant semantic information and matching the cognitive capabilities of the HDM model. Our initial analysis shows how semantic communication can balance the level of detail with

human cognitive capabilities while demanding less bandwidth, power, and latency.

Index Terms

6G, assistance systems, human decision-making, human-machine interface, information maximization (InfoMax), machine learning, psychology, semantic communication, task-oriented communication, wireless communications

6.2 Introduction

With recent breakthroughs in Machine Learning (ML), such as generative Artificial Intelligence (AI) or Natural Language Processing (NLP), assistance systems are now finding their way into everyday life [1]. For example, doctors are supported by expert assistance systems that outperform human expertise in evaluating medical image data for disease diagnosis [2]. Many assistance systems acquire information about physical, chemical, and biological processes through sensors or sensor networks and transmit it to humans for decision-making when performing specific tasks. Applications that exploit such assistance systems include remote operation concepts for production, rescue scenarios, healthcare, autonomous driving, underwater repairs, remote sensing for earth observation and swarm exploration [3]. For example, mobile robotic systems equipped with sensors can assist Human Decision-Making (HDM). All of this relies heavily on efficient and effective wireless communications, which is therefore an integral part of the entire end-to-end sensing-decision-making process.

At this point, semantic communication comes into play as it deals with the question of how information from the assistance system can be communicated more effectively to the human to improve HDM in task execution while demanding less bandwidth, power, and latency. Several research questions can be identified from this interplay:

a) How to model the end-to-end sensing-decision-making process that bridges the disciplines communications and psychology?

b) Is semantic communication suitable for providing the information needed in terms of relevance and accuracy to facilitate effective HDM? Given a task, which and how much information should semantic communication provide, i.e., how to design semantic communication for accurate HDM?

c) Given the provided semantic information, how does the HDM process impact the end-to-end sensing-decision-making process?

To address these questions, we propose integrating semantic communication and HDM into a unified probabilistic end-to-end sensing-decision framework, thereby composing all three levels described by Weaver [4]. To showcase our framework's applicability and highlight its key mechanisms, we examine a case study grounded in an empirical categorization example. As a starting point of our study, we will first reflect upon the State of the Art (SotA) in semantic communication and Human Decision-Making (HDM).

6.2.1 Semantic Communication

In the 1949 review of Shannon's general theory of communication [4], Weaver introduces the idea of semantic communication with regard to both humans and technical systems. There, he used the term communication *"in a very broad sense to include all of the procedures by which one mind may affect another. This, of course, involves not only written and oral speech, but also music, the pictorial arts, the theatre, the ballet, and in fact all human behavior. In some connections it may be desirable to use a still broader definition of communication, namely, one which would include the procedures by means of which one mechanism [...] affects another mechanism [...]."* To meet the unprecedented demands of 6G communication efficiency in terms of bandwidth, latency, and power, attention has been drawn to the broad concept of semantic communication [4]–[9]. It aims to transmit the meaning of a message rather than its exact version, which has been the focus of digital error-free system design [4]. Approaches to the description or design of semantic communication can be divided into statistical probability-based [10], logical probability-based [11], knowledge graph-based [12], and kernel-based [13].

Arguing for the generality of Shannon's theory not only for the technical level but for the semantic level design as Weaver [4], Bao, Basu et al. [14], [15] were the first to define semantic information sources and channels to tackle the semantic design by information-theoretic approaches.

With the rise of Machine Learning (ML) in communication research, transformer-based Deep Neural Networks (DNNs) have been introduced to AutoEncoders (AEs) for text transmission to learn compressed hidden representations of semantic content, aiming to improve communication efficiency [16]. However, accurate recovery of the source (text) is the main goal. The approach improves performance in semantic metrics, especially at low Signal-to-Noise Ratio (SNR), compared to classical digital transmissions. It has been adapted to many other problems, e.g., speech transmission [17],

[18]. Meanwhile, also recent advances in large AI models have found their way into semantic communication [19], [20].

From a theoretical perspective, building upon the ideas of Bao, Basu et al. [14], [15], in [9], [21], the authors explicitly define a semantic random variable and identify the Information Maximization (InfoMax) problem and its variation, the Information Bottleneck (IB) problem, as appropriate semantic design criteria. Solving the InfoMax problem with ML tools, the authors obtain their design Semantic INFOrmation TraNsmission and RecoverY (SINFONY). For more details, we refer the reader to Sec. 6.3.2 and Sec. 6.3.2. Furthermore, task-oriented edge-cloud transmission has been formulated as an IB problem [10].

Semantic communication has been extended to process several types of data, i.e., multimodal data, such as image, text, depth map data [22], [23]. In addition, monitoring, planning, and control of real worlds require the processing of multiple tasks. Thus, in [24], [25], the authors extend the concept of a semantic source to include multiple semantic interpretations. To facilitate cooperative multitask processing and improve training convergence, the semantic encoders are divided into common and specific units, extracting common low-level features and separate high-level features.

So far, the human behind the application or task has only been taken into account by theory, with the rate-distortion-perception trade-off [26], [27]. For example, the mean square error distortion is known to be inconsistent with human perception and thus not a good semantic optimization criterion [26]. Precisely because humans make the final decision when performing a task, we aim to fill the research gap of bridging semantic communication and human decision-making into an end-to-end sensing-decision framework.

6.2.2 Human Decision-Making

Even though the decision capability of artificial systems is increasing, in many situations the final decision-maker will be a human, and humans do not always make rational decisions. Therefore, to optimize the results, the needs, and capabilities of the decision-maker must be considered in the semantic communication design, e.g., by definition of the semantic source.

Humans are undoubtedly expert decision-makers who can cope well with uncertainty and complexity [28], [29]. However, it has been repeatedly shown that Human Decision-Making (HDM) can be systematically biased and decisions can be influenced by irrelevant information and context, as shown in the large literature on heuristics and biases [30]. For example, judges' sentencing decisions can be systematically influenced by asking whether a sentence should be higher or lower than a randomly generated number [31],

and decisions differ depending on whether the same information is presented in frequencies or percentages [32].

Rational models of decision-making typically require the decision-maker to consider all relevant information about the decision options and the context [33]. However, humans have limited cognitive resources, such as attention or working memory capacity, which restricts the amount of information they can process at once [34], [35].

It is often assumed that humans deal with these limited capacities by using simplified decision strategies that often consider only a subset of the information and discard "extra" information [30], [33], [35]. For example, the "take-the-best" heuristic assumes that the decision-maker considers only one dimension at a time in the order of validity of the dimension. A decision is made when the decision-maker encounters a dimension that discriminates between alternatives [36]. Importantly, the use of heuristics such as the take-the-best heuristic often leads to decision performance on par with complex decision rules if the most valid predictors are indeed considered [37].

However, humans are not always able to identify the best predictors, especially when the information environment is complex, they lack expertise, are pressed for time, or are distracted [38]. In these situations, as the growing literature on decision-support/assistance systems shows, human decision-making can be supported and improved by highlighting relevant information, providing summary information, or reducing irrelevant information [39]. Even when the human decision-maker has access to all relevant information and is able to integrate the information properly, humans have a tendency to respond probabilistically [40], [41]. When given several options, and each option has a certain probability of being correct, the optimal decision (that has the highest chance of being correct) is to deterministically choose the option with the highest probability of being correct. While humans choose the best option in the majority of the trials, they usually also tend to choose other options. This variability in human decision-making has likely multiple causes [42].

Semantic communication offers the flexibility to adapt the transmitted information to facilitate the achievement of the human decision-maker's goals. The integration of semantic communication and human decision-making leads to a paradigm shift that includes the communication chain in the decision-support/assistance system.

6.2.3 Main Contributions

The main contributions addressing the above-mentioned research questions are the following:

- In this article, we propose a probabilistic end-to-end sensing-decision framework that wirelessly links sensed data with relevant information-based Human Decision-Making (HDM) by semantic communication.
- Based on this framework, we extend the information-theoretic view on semantic communication towards presentation design and HDM model training, revealing the fundamental presentation or semantic communication design trade-off between maximizing the relevant semantic information and matching the cognitive capabilities of the HDM model. In this sense, our study provides new insights for the design/interaction of semantic communication with models of HDM.
- To showcase our framework's applicability and investigate its key mechanisms, we examine a categorization example using effective HDM models. Simulation results show that, when balancing the design trade-off between feature extraction in semantic communication and cognitive constraints of the HDM model, adjusting the level of detail to match human cognitive capabilities is more important for achieving high decision accuracy than simply providing more relevant information. Moreover, uncertainty in the HDM process decreases accuracy.
- Semantic communication is able to provide the HDM model with sufficient information for making accurate decisions, while demanding less bandwidth, power, and latency compared to classical digital Shannon-based approaches.
- Finally, we provide an outlook on open research questions of our approach, including the design of information presentation through visualization, as well as game theory perspectives on sender-receiver conflicts of interest.

6.3 End-to-End Sensing-Decision Framework

To elaborate on our idea, we now describe our proposed end-to-end sensing-decision framework, which consists of multiple steps, exemplarily sketched in Fig. 6.2 and modeled as shown in Fig. 6.3. It is based on the semantic communication model of [9], including the complete communication Markov chain with the HDM model.

6.3.1 Semantic Source

The human performs tasks such as ensuring that machines in production run smoothly, which requires judging whether a tool is damaged or still

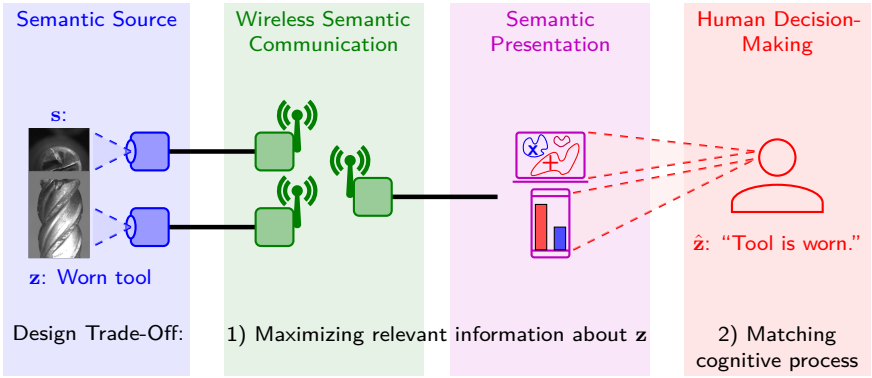

Figure 6.2: Sketch of the end-to-end sensing-decision process for the example of tool wear assessment. It also situates the fundamental design trade-off between semantic communication and human decision-making.

operational. We will refer to this tool categorization task as our guiding example, whose flow is sketched in Fig. 6.2. The task defines the model of the world, i.e., the semantics, and is described by a semantic multivariate Random Variable (RV) $\mathbf{z} \in \mathcal{M}_z^{N_z \times 1}$ from the domain $\mathcal{M}_z$ of dimension N_z, distributed according to a probability density or mass function (pdf/pmf) $p(\mathbf{z})$ [9]. To simplify the discussion, we assume that it is discrete and memoryless.[1] The semantic source $p(\mathbf{s}, \mathbf{z})$ links the semantics expressed by $\mathbf{z}$ with the sensed, observed signal RV $\mathbf{s} \in \mathcal{M}_s^{N_s \times 1}$ that enters the communication system. This sensed data can be, e.g., images of a tool taken from different perspectives, as shown in Fig. 6.2. The semantic link can be modeled in the Markov chain by a semantic channel, a conditional distribution $p(\mathbf{s}|\mathbf{z})$, as shown in Fig. 6.3.

6.3.2 Semantic Communication

The semantic communication system encodes the sensed signal $\mathbf{s}$ with the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$, parametrized by $\boldsymbol{\theta} \in \mathbb{R}^{N_{\boldsymbol{\theta}} \times 1}$, to the transmit signal $\mathbf{x} \in \mathcal{M}_x^{N_{\mathrm{Tx}} \times 1}$ (see Fig. 6.3) for efficient and reliable semantic transmission over the physical communication channel $p(\mathbf{y}|\mathbf{x})$, where $\mathbf{y} \in \mathcal{M}_y^{N_{\mathrm{Rx}} \times 1}$ is the

[1] For the rest of the article, note that the domain of all RVs $\mathcal{M}$ can be either discrete or continuous. Also note that the definition of entropy is different for discrete and continuous RVs. For example, the differential entropy of continuous RVs can be negative, while the entropy of discrete RVs is always positive [43]. Thus, without loss of generality, we will assume that all RVs are either discrete or continuous. In this paper, we avoid notational clutter by using the expectation operator: By replacing the integral with summation over discrete RVs, the equations are valid for continuous RVs and vice versa [9].

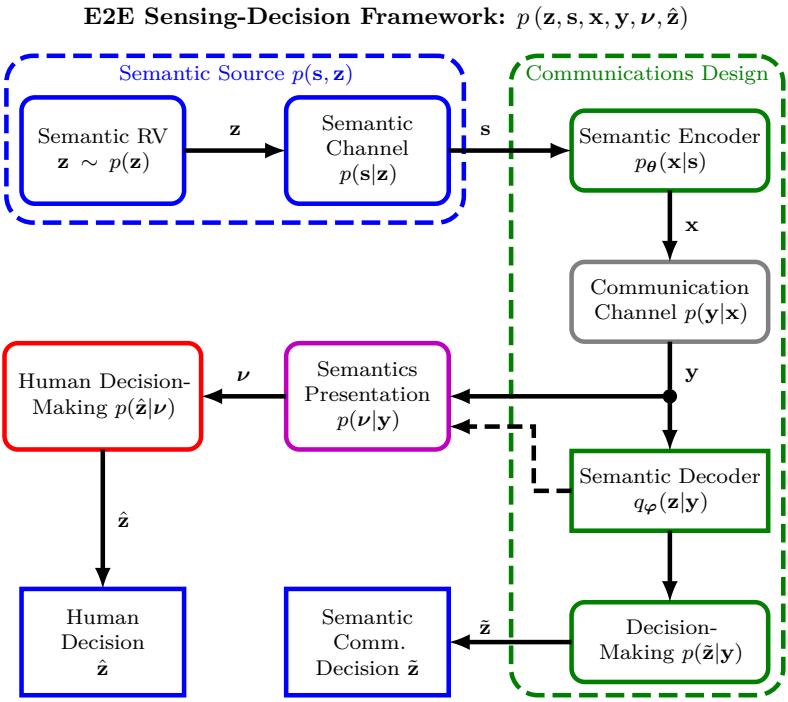

Figure 6.3: Block diagram of the end-to-end sensing-decision framework, i.e., the probabilistic system model including human decision-making.

received signal vector, so that the semantic RV $\mathbf{z}$ is best preserved [9]. At the receiver side, the decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ with parameters $\boldsymbol{\varphi} \in \mathbb{R}^{N_{\boldsymbol{\varphi}} \times 1}$ recovers the semantics $\mathbf{z}$ for the receiver.

In [9], the authors identified the Information Maximization (InfoMax) problem as an appropriate design criterion for semantic communication, since it maximizes the amount of mutual information $I_{\boldsymbol{\theta}}(\mathbf{z}; \mathbf{y})$ of the semantic RV $\mathbf{z}$ contained in the received signal $\mathbf{y}$:

$$\arg\max_{p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})} I_{\boldsymbol{\theta}}(\mathbf{z}; \mathbf{y}) \tag{6.1}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{z},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})} \left[\ln \frac{p_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{y})}{p(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{y})} \right] \tag{6.2}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathcal{H}(\mathbf{z}) - \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y}) \tag{6.3}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathrm{E}_{\mathbf{z},\mathbf{y} \sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})} [\ln p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})] \,. \tag{6.4}$$

There, $\mathrm{E}_{\mathbf{x}\sim p(\mathbf{x})}[f(\mathbf{x})]$ denotes the expected value of $f(\mathbf{x})$ with respect to both discrete and continuous RV $\mathbf{x}$, $\mathcal{H}(\mathbf{z}) = \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})}[-\ln p(\mathbf{z})]$ the entropy of $\mathbf{z}$, and $\mathcal{H}(\mathbf{z}|\mathbf{y})$ the conditional entropy.

If the computation of the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$ in (6.4) is intractable, we can replace it by a variational distribution, i.e., the decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$, to define a MI Lower BOund (MILBO) [9], [44], [45]:

$$\begin{aligned} I_{\boldsymbol{\theta}}(\mathbf{z};\mathbf{y}) &\geq \mathcal{H}(\mathbf{z}) + \mathrm{E}_{\mathbf{z},\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})] && (6.5)\\ &= \mathcal{H}(\mathbf{z}) + \mathrm{E}_{\mathbf{y}\sim p_{\boldsymbol{\theta}}(\mathbf{y})}\left[\mathrm{E}_{\mathbf{z}\sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})}[\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})]\right] && (6.6)\\ &= \mathcal{H}(\mathbf{z}) - \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}\,. && (6.7) \end{aligned}$$

Noting that only the negative amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}$ in (6.7) depends on both $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ and fixing the transmit dimension to N_{Tx}, we can optimize encoder and decoder parameters [9]:

$$\{\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*\} = \arg\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta},\boldsymbol{\varphi}}\,. \tag{6.8}$$

Note that the form of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ must be constrained to avoid learning a trivial identity mapping $\mathbf{y} = \mathbf{s}$. In fact, we constrain the optimization and information rate by our communication channel $p(\mathbf{y}|\mathbf{x})$ and number of channel uses N_{Tx}, which we assume to be given. This introduces an Information Bottleneck (IB). Alternatively, we can explicitly constrain the information rate $I_{\boldsymbol{\theta}}(\mathbf{s};\mathbf{y})$ in an IB problem [9], [21]. To solve (6.8), we use the empirical cross-entropy and ML techniques such as DNNs, stochastic gradient descent, and the reparametrization trick to obtain our ML-based design Semantic INFOrmation TraNsmission and RecoverY (SINFONY) [9], [21].

We note that the semantic communication system is able to make a decision by itself after optimization/training based on the decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$. For the discrete RVs, the most likely option, i.e., the MAP estimate, is optimal:

$$\tilde{\mathbf{z}} = \arg\max_{\mathbf{z}} q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})\,. \tag{6.9}$$

This decision process in operation mode is modeled as $p(\tilde{\mathbf{z}}|\mathbf{y})$.

6.3.3 Semantics Presentation

Finally, semantic communication presents the received signal $\mathbf{y}$ or the extracted probabilistic semantic decoder estimate $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ — in the best case containing maximum amount of information about the semantic RV $\mathbf{z}$

according to (6.4) or (6.5) — to the HDM model. We describe this process by $p(\boldsymbol{\nu}|\mathbf{y})$ with a presentation RV $\boldsymbol{\nu} \in \mathbb{R}^{N_\phi \times 1}$. In practice, the presentation $\boldsymbol{\nu}$ must be tailored to a human, requiring a Human-Machine Interface (HMI) that is typically designed handcrafted, such as visualization (see Fig. 6.2). In this work, we abstract the HMI as in a technical system — where the components are connected by a deterministic function $\boldsymbol{\nu} = f(\mathbf{y})$.

6.3.4 Human Decision-Making Model

Based on the HMI or semantics presentation $\boldsymbol{\nu}$, the human decision-maker then makes a decision to complete the overall task. In this work, we will model the Human Decision-Making (HDM) process probabilistically by $p(\hat{\mathbf{z}}|\boldsymbol{\nu})$ to make a first step towards integrating and evaluating the human with the technical system, i.e., semantic communication and HDM. Finally, by decision, we obtain the estimated semantics $\hat{\mathbf{z}} \in \mathcal{M}_z^{N_z \times 1}$, which can be different from the true semantic RV $\mathbf{z}$ (see Fig. 6.2). In our guiding example, this could mean that the HDM process decides that the tool is damaged even though it is still usable, and vice versa.

Reflecting the variance in decision tasks, the literature on HDM includes a variety of theoretical models and approaches to capture decision processes [46], [47]. The most appropriate model often varies depending on the type of decision task and context. In this example, we focus on categorization tasks where the decision-maker must decide based on their experience whether an object belongs to one of M categories, such as whether a tool can still be used or whether the concentration of a toxic gas is above a certain threshold.

Generalized Context Model

While a large number of increasingly complex models of human categorization have been proposed [48]–[50], the core assumptions of the Generalized Context Model (GCM) [51], [52] are commonly adapted by many successors [53] and have been successfully used to describe categorizations of complex real world stimuli [54], [55].

Despite the relative simplicity of the GCM, the well-studied model and its variants can easily account for HDM under different contexts, e.g., under time pressure [56], [57], capture human judgment under cognitive load [58], and explain common HDM biases, e.g., base-rate bias [59]. At the same time, GCM has been applied to different aspects of human cognition, e.g., artificial grammar [60], leadership competence judgment [61], and mental multiplication [62]. Several extensions of GCM have been created to explain even broader aspects of HDM such as reaction time in decision-making [63]

or learning [64]. Since the GCM is a powerful approximation to human categorization decisions, we choose it to simulate the decision-making process.

An important assumption of the GCM is that categorization decisions are made on the basis of exemplar memory, i.e., previously seen realizations that are retrieved from memory. Accordingly, in the tool example, the model assumes that the decision-maker first experiences N tool realizations and whether those tools need to be replaced. These tools are then remembered as the i-th "exemplar", i.e., realization $\boldsymbol{\nu}_i$, with the corresponding label $\mathbf{z}_i$, so we have an exemplar dataset or HDM knowledge base $\mathcal{D}_{\text{HK}} = \{\boldsymbol{\nu}_i, \mathbf{z}_i\}_{i=1}^{N}$. Since the semantic RV $\mathbf{z}$ is a categorical RV, we can describe it by a one-hot vector $\mathbf{z} = \text{one-hot}(k)$ where all elements are zero except for the element $k \in \{1, \ldots, M\}$ that represents the tool state from a total number of M states. For example, for binary states, we have $k \in \{1, 2\}$ with $M = 2$.

When the decision-maker encounters a new tool presentation $\boldsymbol{\nu}$, the probability $q_{\boldsymbol{\varphi}_{\text{G}}}(\mathbf{z} = \mathbf{z}|\boldsymbol{\nu}, \mathcal{D}_{\text{HK}}) = q(\mathbf{z} = \mathbf{z}|\boldsymbol{\nu}, \mathcal{D}_{\text{HK}}, \boldsymbol{\varphi}_{\text{G}})$ of making the decision $\mathbf{z} = \text{one-hot}(\mathsf{k})$ given this representation $\boldsymbol{\nu}$ is the result of the comparison between $\boldsymbol{\nu}$ and all seen realizations $\boldsymbol{\nu}_i$ from $\mathcal{D}_{\text{HK}}$:

$$q_{\boldsymbol{\varphi}_{\text{G}}}(\mathbf{z} = \mathbf{z}|\boldsymbol{\nu}, \mathcal{D}_{\text{HK}}) = \frac{\sum_{i=1}^{N} \text{sim}(\boldsymbol{\nu}_i, \boldsymbol{\nu}|\boldsymbol{\varphi}_{\text{G}}) \cdot [\mathbf{z}_i = \mathbf{z}]}{\sum_{i=1}^{N} \text{sim}(\boldsymbol{\nu}_i, \boldsymbol{\nu}|\boldsymbol{\varphi}_{\text{G}})} \tag{6.10}$$

with GCM parameters $\boldsymbol{\varphi}_{\text{G}}$ and $[\mathbf{z}_i = \mathbf{z}]$ being the Iverson bracket, which is equal to 1 if $\mathbf{z}_i = \mathbf{z}$ and 0 otherwise. This means the approximating posterior $q_{\boldsymbol{\varphi}_{\text{G}}}(\mathbf{z}|\boldsymbol{\nu}, \mathcal{D}_{\text{HK}})$ is determined by the sum of similarities between $\boldsymbol{\nu}$ and all the seen realizations $\mathbf{z}_i$ that belong to the decision $\mathbf{z}$, and normalized by the similarity to all the seen realizations regardless of the decision. We note that the model (6.10) assumes that the decision-maker has perfect memory of its knowledge base $\mathcal{D}_{\text{HK}}$.

The similarity $\text{sim}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2|\boldsymbol{\varphi}_{\text{G}})$ between two presentations decreases exponentially as the Euclidean distance between two presentations increases and depends on the learnable GCM parameters $\boldsymbol{\varphi}_{\text{G}} = \{\gamma, \mathbf{w}\}$:

$$\text{sim}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2|\boldsymbol{\varphi}_{\text{G}}) = e^{-\gamma \cdot \left(|\boldsymbol{\nu}_1 - \boldsymbol{\nu}_2|^T \cdot \text{diag}\{\mathbf{w}\} \cdot |\boldsymbol{\nu}_1 - \boldsymbol{\nu}_2|\right)^{1/2}} \tag{6.11}$$

with $\text{diag}\{\mathbf{w}\}$ creating a diagonal matrix with elements of $\mathbf{w} \in \mathbb{R}^{N_\phi \times 1}$ on its diagonal.

A unique assumption of the GCM is that each element or feature ν_n of a seen realization $\boldsymbol{\nu}_i$ is weighted by attention weights w_n and hence does not contribute equally to the perceived similarity. To achieve the attention functionality, the weights are constrained by $w_n \geq 0$ and normalized by

$\sum_{n=1}^{N_\phi} w_n = 1$. The parameter γ has two interpretations: First, the similarity gradient γ describes the sharpness of the decline in similarity, with higher γ resulting in a sharper decline of similarity when the distance increases. Second, the parameter γ describes the consistency in making decisions and reflects the probabilistic nature of the HDM process, akin to the temperature parameter in the Boltzmann distribution [65]. Accordingly, a lower parameter γ results in a new tool being more confidently categorized to the category with higher similarity.

HDM-based Probabilistic Decision-Making

After training of the GCM (see Sec. 6.3.6), the strategy of the HDM model is equal to the random process

$$\hat{\mathbf{z}} \sim p(\hat{\mathbf{z}}|\boldsymbol{\nu}) = q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z} = \hat{\mathbf{z}}|\boldsymbol{\nu}, \mathcal{D}_\mathrm{HK})\,. \tag{6.12}$$

When faced with options of varying probabilities, people tend to distribute their choices according to the probability distribution rather than always selecting the most likely option resulting in a suboptimal policy. Thus, the accuracy of human decisions can be worse compared to that of the optimal deterministic policy (6.9) of the technical (semantic communication) system, e.g., SINFONY.

6.3.5 End-to-End Sensing-Decision Model

With all the aforementioned subcomponent models, we are able to create a generative model of the end-to-end sensing-decision process. We note that we can distinguish between four different models corresponding to four system stages:

1. Design of semantic encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ and decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ based on the forward communications model

$$p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}) = p(\mathbf{z}, \mathbf{s}) \cdot p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s}) \cdot p(\mathbf{y}|\mathbf{x})\,. \tag{6.13}$$

2. Semantic communication is executed in operation mode to make decisions via (6.9). Then, the model is

$$p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}, \tilde{\mathbf{z}}) = p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}) \cdot p(\tilde{\mathbf{z}}|\mathbf{y})\,. \tag{6.14}$$

3. The HDM model $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu}, \mathcal{D}_\mathrm{HK})$ is trained based on seen presentation and label realizations from semantic communication in operation mode. The underlying model is

$$p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}, \boldsymbol{\nu}) = p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}) \cdot p(\boldsymbol{\nu}|\mathbf{y})\,. \tag{6.15}$$

4. Semantic communication presents information to the HDM model that finally makes a decision. The overall end-to-end sensing-decision model of Fig. 6.3 in operation mode after all training phases is:

$$p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}, \boldsymbol{\nu}, \hat{\mathbf{z}}) = p(\mathbf{z}, \mathbf{s}, \mathbf{x}, \mathbf{y}, \boldsymbol{\nu}) \cdot p(\hat{\mathbf{z}}|\boldsymbol{\nu}) \,. \tag{6.16}$$

6.3.6 Information-theoretic Overall View on Design in the End-to-End Sensing-Decision Framework

We can exploit our end-to-end sensing-decision framework (6.13)-(6.16), to extend the information-theoretic view of semantic communication to both the semantics presentation and the HDM model to gain new insights.

HDM Model – Training

For optimization of the GCM parameters $\boldsymbol{\varphi}_{\mathrm{G}} = \{\gamma, \mathbf{w}\}$ given a presentation $\boldsymbol{\nu}$ based on a fixed optimized semantic communication system of model (6.15), typically the maximum likelihood criterion is used [52]. We note that maximization of the log-likelihood function is equal to amortized minimization of the empirical cross-entropy on the training set [43]. Transferring the InfoMax view from the semantic communication system in (6.7), this means we optimize a lower bound on the mutual information $I_{\boldsymbol{\theta}}(\mathbf{z}; \boldsymbol{\nu})$, but now between $\mathbf{z}$ and $\boldsymbol{\nu}$ with respect to $\boldsymbol{\varphi}_{\mathrm{G}}$:

$$\begin{aligned} I_{\boldsymbol{\theta}}(\mathbf{z}; \boldsymbol{\nu}) &\geq \mathcal{H}(\mathbf{z}) + \mathrm{E}_{\mathbf{z}, \boldsymbol{\nu} \sim p_{\boldsymbol{\theta}}(\mathbf{z}, \boldsymbol{\nu})}[\ln q_{\boldsymbol{\varphi}_{\mathrm{G}}}(\mathbf{z}|\boldsymbol{\nu}, \mathcal{D}_{\mathrm{HK}})] & (6.17) \\ &= \mathcal{H}(\mathbf{z}) - \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\varphi}_{\mathrm{G}}} \,. & (6.18) \end{aligned}$$

From an information-theoretic view, we conclude that the choice of the GCM optimization criterion

$$\boldsymbol{\varphi}^{*}_{\mathrm{G}} = \arg\min_{\boldsymbol{\varphi}_{\mathrm{G}}} \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\varphi}_{\mathrm{G}}} \tag{6.19}$$

is well-motivated. To solve (6.19), computer search methods are typically used [52]. In this work, we employ a variant known as the differential annealing algorithm. By integrating differential evolution's population-based search with simulated annealing's probabilistic acceptance of solutions, differential annealing aims to balance exploration and exploitation in complex search spaces to enhance global optimization capabilities [66].

Semantics Presentation – Design Optimization

Moreover, if we add the presentation process as a tunable encoder $p_{\boldsymbol{\theta}_{\mathrm{P}}}(\boldsymbol{\nu}|\mathbf{y})$ with parameters $\boldsymbol{\theta}_{\mathrm{P}}$ to the optimization problem (6.19), we arrive at the

MILBO objective function:

$$\begin{aligned} I_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z};\boldsymbol{\nu}) \geq & \mathcal{H}(\mathbf{z}) \\ & + \mathrm{E}_{\mathbf{z},\mathbf{y},\boldsymbol{\nu}\sim p_{\boldsymbol{\theta}}(\mathbf{z},\mathbf{y})\cdot p_{\boldsymbol{\theta}_\mathrm{P}}(\boldsymbol{\nu}|\mathbf{y})}[\ln q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK})] \\ = & \mathcal{H}(\mathbf{z}) - \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta}_\mathrm{P},\boldsymbol{\varphi}_\mathrm{G}} . \end{aligned} \tag{6.20}$$

Decomposing the amortized cross-entropy $\mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta}_\mathrm{P},\boldsymbol{\varphi}_\mathrm{G}}$ as in [9] into

$$\begin{aligned} \mathcal{L}^{\mathrm{CE}}_{\boldsymbol{\theta}_\mathrm{P},\boldsymbol{\varphi}_\mathrm{G}} = & \mathcal{H}(\mathbf{z}) - I_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z};\boldsymbol{\nu}) \\ & + \mathrm{E}_{\boldsymbol{\nu}\sim p_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\boldsymbol{\nu})}[D_\mathrm{KL}(p_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z}|\boldsymbol{\nu}) \parallel q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK}))] \end{aligned} \tag{6.21}$$

reveals two possibly conflicting design criteria:

1. The presentation encoder $p_{\boldsymbol{\theta}_\mathrm{P}}(\boldsymbol{\nu}|\mathbf{y})$ should maximize the mutual information $I_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z};\boldsymbol{\nu})$ that depends solely on it through the true posterior $p_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z}|\boldsymbol{\nu})$ (see (6.4)).
2. Both true posterior $p_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z}|\boldsymbol{\nu})$ and hence the presentation encoder $p_{\boldsymbol{\theta}_\mathrm{P}}(\boldsymbol{\nu}|\mathbf{y})$ and the HDM model $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK})$ are matched by minimizing the Kullback–Leibler (KL) divergence.

In a technical semantic communication system from Sec. 6.3.2, we can avoid the design conflict by using a model $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ expressive enough to approximate $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{y})$ arbitrarily well, such that the focus lies on the InfoMax term. However, in case of the end-to-end sensing-decision training model (6.15), if the HDM model (or human) constrains the form of $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK})$, i.e., the solution space, to some degree, the two optimization terms in (6.21) are traded-off: Then, the true posterior $p_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z}|\boldsymbol{\nu})$ has to be fit to $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK})$ and we do not maximize $I_{\boldsymbol{\theta},\boldsymbol{\theta}_\mathrm{P}}(\mathbf{z};\boldsymbol{\nu})$ alone which could lead to a loss in mutual information.

Example 1: These abstract information-theoretic insights explain well what we observe in practice with handcrafted presentations. In reality, it is difficult to understand and subsequently visualize the received raw communications signal $\mathbf{y}$ for a human without any preprocessing:

- We have to match the presentation encoder $p_{\boldsymbol{\theta}_\mathrm{P}}(\boldsymbol{\nu}|\mathbf{y})$ to the cognitive process $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK})$ according to the KL divergence term in (6.21). Fortunately, the semantic decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ obtained by maximizing the MILBO extracts the semantic information of $\mathbf{z}$ from $\mathbf{y}$ and allows for meaningful presentation to and interpretation by the HDM model $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu},\mathcal{D}_\mathrm{HK})$ (or human).
- However, we may lose relevant information about $\mathbf{z}$ fitting the presentation to the cognitive process including the semantic decoder

preprocessing in the Markov chain $\mathbf{z} \to \mathbf{y} \to q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y}) \to \boldsymbol{\nu}$ according to the data processing inequality

$$I(\mathbf{z};\mathbf{y}) \geq I(\mathbf{z}; q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})) \geq I(\mathbf{z};\boldsymbol{\nu}) \ . \tag{6.22}$$

Example 2: Another example of how the HDM process influences presentation design is that research on HDM models focuses on the interplay between relevant features $\boldsymbol{\nu}$ and require certain level of feature extraction from raw images $\mathbf{s}$ (or $\mathbf{y}$) for HDM model processing. For an overview of this research, we refer the reader to [67]. These HDM models were not built to process raw images $\mathbf{s}$ directly, i.e., $q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z}|\boldsymbol{\nu} = \mathbf{s}, \mathcal{D}_\mathrm{HK})$, which would lead to unrealistically poor performance despite maximum relevant information in $\mathbf{s}$ about $\mathbf{z}$. Thus, in the numerical results of Sec. 6.4.4, we cannot compare to a setup where the raw data of the images $\mathbf{s}$ are digitally communicated and then directly processed by the HDM model.

Based on our end-to-end sensing-decision framework, we conclude that it highly depends on the processing capabilities of the HDM model if it can extract more or less information about $\mathbf{z}$ from $\mathbf{y}$ than the semantic decoder. Moreover, we conclude that balancing of two possibly conflicting criteria is key for presentation design:

1. **Relevant information preservation:** On the one hand, careful design of $\boldsymbol{\nu}$ is required to not lose any relevant information about $\mathbf{z}$ for the final decision. For example, the higher the dimension N_ϕ of the presentation RV $\boldsymbol{\nu}$, the more detailed the presentation to the HDM model and the more information it contains.

2. **Presentation alignment to the HDM model:** On the other hand, the presentation has to be in a form that can be understood by the HDM model, effectively restricting the set of possible presentations $\boldsymbol{\nu}$. For example, compressing the relevant information about $\mathbf{z}$ into $\boldsymbol{\nu}$ may be required to ease cognitive processing.

To investigate how to balance these two design rules, we compare two handcrafted presentations in our numerical example of Sec. 6.4.4. For an outlook on the inclusion of HMIs in practice, please refer to Sec. 6.5.

6.4 Simulative Investigation

In this section, we evaluate first numerical results of our joint framework using the example of image classification. The inclusion of diverse datasets for semantic source modeling, such as the standard MNIST and CIFAR10

datasets in addition to our guiding tool example, enables the generalization of conclusions beyond the specific case of tool wear.

6.4.1 Performance Measures

We measure the performance of decision-making for both semantic communication and the HDM model by the categorical accuracy

$$\mathcal{A} = \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})}[p\left(\tilde{\mathbf{z}} = \mathbf{z}|\mathbf{z}\right)] = \sum_{\mathbf{z}\in\mathcal{M}_z^{N_z\times 1}} p\left(\tilde{\mathbf{z}} = \mathbf{z}, \mathbf{z}\right) \tag{6.23}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N}[\tilde{\mathbf{z}}_i = \mathbf{z}_i] \tag{6.24}$$

— comparing predicted and true category realizations $\tilde{\mathbf{z}}_i$ and $\mathbf{z}_i$ — or the classification error rate $1-\mathcal{A}$ common in communications. Since the HDM model decides probabilistically based on the input $\boldsymbol{\nu}_i$, we can calculate the accuracy based on the end-to-end sensing-decision model (6.16) shown in Fig. 6.3 by the sum of the probabilities of the GCM responding to the correct category $\mathbf{z}_i$ [51]:

$$\mathcal{A} = \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})}[p\left(\hat{\mathbf{z}} = \mathbf{z}|\mathbf{z}\right)] \tag{6.25}$$

$$= \mathrm{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\mathrm{E}_{\boldsymbol{\nu}\sim p(\boldsymbol{\nu}|\mathbf{z})}[p\left(\hat{\mathbf{z}} = \mathbf{z}|\boldsymbol{\nu}\right)]\right] \tag{6.26}$$

$$= \mathrm{E}_{\mathbf{z},\boldsymbol{\nu}\sim p(\mathbf{z},\boldsymbol{\nu})}[p\left(\hat{\mathbf{z}} = \mathbf{z}|\boldsymbol{\nu}\right)] \tag{6.27}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} p(\hat{\mathbf{z}} = \mathbf{z}_i|\boldsymbol{\nu} = \boldsymbol{\nu}_i) \tag{6.28}$$

$$= \frac{1}{N}\sum_{i=1}^{N} q_{\boldsymbol{\varphi}_\mathrm{G}}(\mathbf{z} = \mathbf{z}_i|\boldsymbol{\nu} = \boldsymbol{\nu}_i, \mathcal{D}_\mathrm{HK})\,. \tag{6.29}$$

This method of calculating accuracy is commonly used in psychology studies for HDM models [51].

6.4.2 Example Semantic Source Datasets

Tool wear and tool replacement decisions pose a common challenge in the metal cutting industry to reduce production costs [68], and represent an exemplary semantic source of this work. In this decision-making problem, the semantic RV $\mathbf{z}$ is modeled as a binary variable with two states, where $\mathbf{z} = [1,0]^T$ indicates a worn tool and $\mathbf{z} = [0,1]^T$ indicates a usable tool. Although optical measurement techniques provide accurate assessments

of tool wear, small and medium-sized companies often rely on machine operators to manually assess tool wear. To automate this process, a dataset was recorded where human experts were presented two different grayscale images of each tool [69]: One image $\mathbf{s}_1 \in \{0, 1, \ldots, 255\}^{218\times380\times1}$ taken from the side and another $\mathbf{s}_2 \in \{0, 1, \ldots, 255\}^{487\times380\times1}$ taken from the top (see Fig. 6.2). Based on these observations $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2\}$, the experts labeled the tools into the binary states $\mathbf{z}$.

This process resulted in a dataset $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{z}_i)\}_{i=1}^{N}$ modeling our semantic source $p(\mathbf{s}, \mathbf{z})$ and consisting of $N = 1632$ data pairs, with an 85% split between training and validation data. We also revisit the MNIST and CIFAR10 examples from [9], [70] to extend our analysis.

6.4.3 Semantic Communication Analysis

As the design approach for semantic communication and to solve (6.8) with the model (6.13), we use our ML-based SINFONY approach from [9], [21]. However, we note that the conclusions derived from the results about the interplay between semantic communication and HDM models are not limited to this approach. These also extend to other, e.g., model-based, methods capable of providing the same quality of soft information at inference runtime. For example, RL-SINFONY leverages reinforcement learning to train the design via (6.7) to comparable performance [21].

SINFONY Design

As shown in [9], [21], we apply SINFONY to a distributed multipoint scenario, where meaning from multiple image sources is communicated to a single receiver for semantic recovery of the RV $\mathbf{z}$. In the numerical example of [9], four distributed agents extract features from different image views with an encoder based on the famous and powerful ResNet architecture [71] for rate-efficient transmission. Based on the received signals, the decoder recovers semantics by classification. Numerical results of [9] on images from the MNIST and CIFAR10 datasets show that SINFONY outperforms classical digital communication systems in terms of bandwidth, latency and power efficiency.

In this article, we reuse the SINFONY approach for integration with the HDM model. SINFONY is particularly well-suited for integration because it can be easily adapted to any semantic source $p(\mathbf{s}, \mathbf{z})$, i.e., use case, including tool damage classification, by changing the data samples and specifically designing its DNN architecture.

In the guiding tool example, two image sensors provide different views of the tool (see Fig. 6.2). This results in a SINFONY design (see Communica-

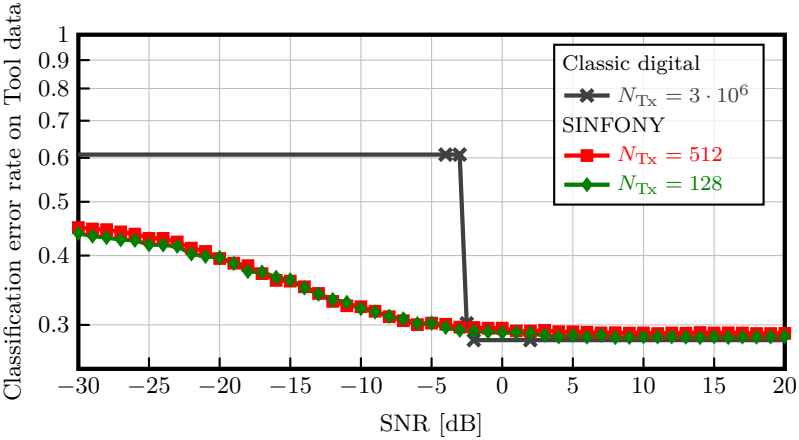

Figure 6.4: Comparison of the classification error rate of SINFONY with different number of channel uses N_{Tx} per encoder and central image processing with digital image transmission on the tool validation dataset as a function of SNR.

tions Design in Fig. 6.3) with two encoders $p^i_{\boldsymbol{\theta}}(\mathbf{x}_i|\mathbf{s}_i)$ with $i = \{1, 2\}$ that can be concatenated into one virtual encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$, and one decoder $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$. Owing to the large image dimensions of $\mathbf{s}_i$, we adopt the ImageNet version of ResNet18 to reduce numerical complexity in feature extraction [72], resulting in $N_{\mathrm{Feat}} = 512$ features per encoder. We test two SINFONY Tx module designs that map those features onto the transmit signal $\mathbf{x}_i \in \mathcal{M}_x^{N_{\mathrm{Tx}} \times 1}$: one with feature compression ($N_{\mathrm{Tx}} = 128$) and one without ($N_{\mathrm{Tx}} = 512$). Note that the number of channel uses N_{Tx} is proportional to bandwidth, i.e., $B \sim N_{\mathrm{Tx}}$. The signals $\mathbf{x}_i$ are transmitted over an AWGN channel $p(\mathbf{y}_i|\mathbf{x}_i)$ to the decoder that consists of a common Rx layer of width $N_{\mathrm{w}} = 1024$ processing the concatenated received signals $\mathbf{y}_i \in \mathcal{M}_y^{N_{\mathrm{Rx}} \times 1}$, each of length $N_{\mathrm{Rx}} = N_{\mathrm{Tx}}$, and a final softmax layer with $M = 2$ classes. As in [9], we train for $\mathrm{SNR}_{\mathrm{train}} \in [-4, 6]$ dB in model (6.13). We also reuse the SINFONY designs for the MNIST and CIFAR10 datasets from [9] and combine them with the HDM model. The source code including all details is available in [70].

SINFONY-based Decision-Making

First, we evaluate the performance of SINFONY-based decision-making within the semantic communication operation mode model (6.14), where

SINFONY makes the final decision via (6.9). Fig. 6.4 shows the classification error rate $1 - \mathcal{A}$ with $\mathcal{A}$ from (6.24) on the tool dataset as a function of SNR. The key findings are similar to those for MNIST and CIFAR10 [9] but become much more obvious: Using less channel uses per encoder with SINFONY $N_{\mathrm{Tx}} = 128$ than the number of features ($N_{\mathrm{Feat}} = 512$) results in the same performance compared to SINFONY Tx/Rx $N_{\mathrm{Tx}} = 512$. This indicates that feature compression and thus a reduction in bandwidth is possible.

Moreover, we compare to central image processing by a ResNet classifier [70] with classic digital transmission of the sensed images (Classic digital): We assume that the RGB image bits are Huffman encoded, protected by an LDPC code with rate 0.25 and BPSK modulated. At the receiver side, we use belief propagation for decoding. On average, the channel is utilized over $23,400$ times more frequently per encoder, with $N_{\mathrm{Tx}} \approx 2,998,626.82 \approx 3 \cdot 10^6$ uses. Furthermore, at low SNR, significantly more power is needed to achieve the same classification error rate, e.g., about 10 dB more for 35%. Instead of graceful degradation as for SINFONY, we observe a cliff effect typical for digital communication at a SNR threshold of -2.5 dB: Communication quality remains robust as long as channel capacity exceeds code rate and the LDPC code operates within its working point, but rapidly breaks down otherwise. This sharp contrast in performance and bandwidth highlights the huge potential of semantic communication.

6.4.4 End-to-end Sensing-Decision Analysis

Now, we assume our end-to-end sensing-decision model, i.e., the overall model (6.16): The semantic information in the images is transmitted by SINFONY over an AWGN channel and then fed into the HDM model, i.e., the GCM. Note that, in contrast to the SINFONY scenario of Sec. 6.4.3, the HDM model now makes the final decision.

Semantics Presentation Design

In Sec. 6.3.6, we derive two design criteria for the semantics presentation $\boldsymbol{\nu}$: 1) It should keep all relevant information about $\mathbf{z}$. 2) It should fit to cognitive processing capabilities of the HDM model. We note that since HDM models are not capable to process the raw data of the images $\mathbf{s}$ directly [67] as outlined in Sec. 6.3.6, we cannot simply compare to a setup where the raw data of the images $\mathbf{s}$ are digitally communicated and processed. This means the design choice $\boldsymbol{\nu} = \mathbf{s}$ is ruled out in this work.

Therefore, we aim to gain first insights on the design trade-off by comparing HDM model performance with different presentations that reflect a different

weighting of the two design criteria. To reflect practical considerations as outlined in Sec. 6.3.6, we design the presentation handcrafted based on the semantic decoder (see Fig. 6.2). In this context, in other words, we investigate how to balance the feature extraction of semantic communication and HDM models to achieve the best task performance, i.e., to minimize the classification error rate.

We present either the categorical probability outputs (E2E categorical) or the relevant decision features (E2E N_ϕ) from SINFONY as $\boldsymbol{\nu}$ to the HDM model, i.e., the GCM:

1. **E2E categorical:** The low-dimensional and interpretable probability estimate of SINFONY for each category (E2E categorical) that fulfills design rule 2), e.g., whether the tool is damaged or not:

$$\boldsymbol{\nu} = f_1(q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})) = \begin{bmatrix} q_{\boldsymbol{\varphi}}\left(\mathbf{z} = \text{one-hot}(1)^T|\mathbf{y}\right) \\ \vdots \\ q_{\boldsymbol{\varphi}}\left(\mathbf{z} = \text{one-hot}(M)^T|\mathbf{y}\right) \end{bmatrix}. \tag{6.30}$$

2. **E2E $\boldsymbol{N_\phi}$:** To provide the HDM model at an abstract level with potentially more relevant semantic information about $\mathbf{z}$ according to data processing inequality (6.22) and design rule 1) for decision-making, we use the relevant decision features

$$\boldsymbol{\nu} = f_2(\mathbf{y}) = \mathbf{v}_{q_{\boldsymbol{\varphi}}}^{(N_\mathrm{L}-1)}(\mathbf{y}), \tag{6.31}$$

where $\mathbf{v}_{q_{\boldsymbol{\varphi}}}^{(l)}$ is the output of the l-th layer of $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$ and N_L the depth of the DNN. This means we extract the inputs to the final dense softmax layer of the SINFONY decoder used for probability estimation, similar to a previous study that aims to model categorization with natural material [73].

To further vary the level of detail or dimension of the presentation, we extract the most important $N_\phi = \{5, 10, 20, 40\}$ of the $N_\mathrm{h}^{(N_\mathrm{L}-1)}$ final layer features to facilitate effective processing of the HDM model according to design rule 2). The importance of the i-th feature $[\mathbf{v}_{q_{\boldsymbol{\varphi}}}^{(N_\mathrm{L}-1)}]_i$, with respect to all output nodes $q_{\boldsymbol{\varphi}}(\mathbf{z} = \text{one-hot}(k)^T|\mathbf{y})$, is quantified by the sum of the absolute weight values in each column of the last-layer weight matrix $\mathbf{W}^{(N_\mathrm{L}-1)}$, given by $\sum_{k=1}^{M} |w_{ki}^{(N_\mathrm{L}-1)}|$.

Based on the selected presentation, i.e., SINFONY features, the GCM classifies into M categories, e.g., into the binary tool states of wear and

non-wear. To present the SINFONY features in the numerical evaluation, we use the SINFONY version with $N_{\mathrm{Tx}} = 128$ from Fig. 6.4 for semantic communication (see Sec. 6.3.2) on the tool dataset. For MNIST and CIFAR10 datasets, we use the trained SINFONY versions with $N_{\mathrm{Tx}} = 56$ and $N_{\mathrm{Tx}} = 64$ from [9], [70]. Note that N_{Tx} differs per dataset, since we tailored the SINFONY architecture to the specific dataset.

Simulation Scenarios

We evaluate our proposed framework on three datasets: Tools, MNIST, and CIFAR10. Furthermore, we perform two main simulations — a simulation of the accuracy as a function of the SNR typical for communications, and a simulation of the expertise of the HDM model touching a psychological aspect:

1. In the SNR simulation, we assume that the HDM model (6.10) has sufficient experience with the presented SINFONY features, i.e., it has perfect memory of the training set with $\mathcal{D}_{\mathrm{HK}} = \mathcal{D}_{\mathrm{T}}$, i.e., the seen presented realizations encompass the entire training dataset of semantic communication. Its attention weights $\mathbf{w}$ and the similarity gradient γ from (6.11) are optimized to maximize the classification accuracy on the training set at a training SNR of 20 dB. For evaluation, the HDM model receives the output of SINFONY under varying SNR.

2. In the expert simulation, we assume the highest evaluated SNR of 20 dB during communication and vary the number of seen presentation realizations, i.e., images randomly selected from the training set. We define the number of seen realizations $|\mathcal{D}_{\mathrm{HK}}|$ of classified tools as the expertise of the HDM model and simulate the performance at this expertise independently. Accordingly, a HDM model with high expertise has a larger knowledge base $\mathcal{D}_{\mathrm{HK}}$ compared to a HDM model with low expertise. The GCM parameters were optimized for the specific HDM knowledge bases $\mathcal{D}_{\mathrm{HK}}$.

In both simulations, the accuracy was calculated based on the validation dataset. We performed multiple Monte Carlo runs for evaluation: For the SNR simulations, we iterated ten times through the dataset for each SNR value. For the expertise simulations, we iterated 100 times for each expertise level.

Numerical Results

We present the simulation results in Fig. 6.5, comparing to SINFONY-based decision-making (SINFONY) of the technical system on $\tilde{\mathbf{z}}$ via (6.9) as the

baseline. First, we note that the accuracy of the GCM is worse than that of SINFONY. This can be explained by the probabilistic decision process (6.12) of the HDM model, which deviates from the optimal strategy to choose the most likely option.

SNR Simulation

Furthermore, the SNR curves show a similar trend for all three datasets. Accuracy increases as a function of SNR and plateaus after a certain SNR is reached. The performance of the HDM model is best when receiving the categorical probability input of SINFONY (E2E categorical) compared to receiving the N_ϕ most important feature dimensions (E2E N_ϕ). With the latter input, more features yield better accuracy, and the performance reaches an asymptote after a certain number of feature dimensions. The number of features needed to reach saturation varies depending on the dataset.

Reaching saturation indicates that the HDM model is not able to extract more relevant information about $\mathbf{z}$ from many feature inputs, i.e., from a more detailed representation. In contrast, the model performs better with SINFONY's preprocessed probability estimates, indicating SINFONY's ability to efficiently extract the semantic information. This indicates that InfoMax optimized outputs are suitable as an input for human decision-making: SINFONY's graceful degradation translates directly into the GCM curve.

Expertise Simulation

The expertise simulation (bottom row in Fig. 6.5) shows that accuracy increases with the number of seen presentation realizations. Regardless of expertise, using the probability output of SINFONY again results in the best performance compared to receiving the N_ϕ important feature dimensions. This shows that the GCM's semantic information processing was not as effective as that of SINFONY.

Moreover, unlike in the SNR simulation where a larger number of N_ϕ features always yields better performance, the accuracy does not always increase with the number of features under different expertise levels. For example, the accuracy on the tool dataset with $N_\phi = 10$ and 20 features exceeds the accuracy simulated with 40 features at lower expertise. Even with the highest expertise on the CIFAR10 dataset, the accuracy with 20 features still beats that with 40 features.

This means that GCM is not always capable to learn to effectively extract the semantic information when provided with extra information. This

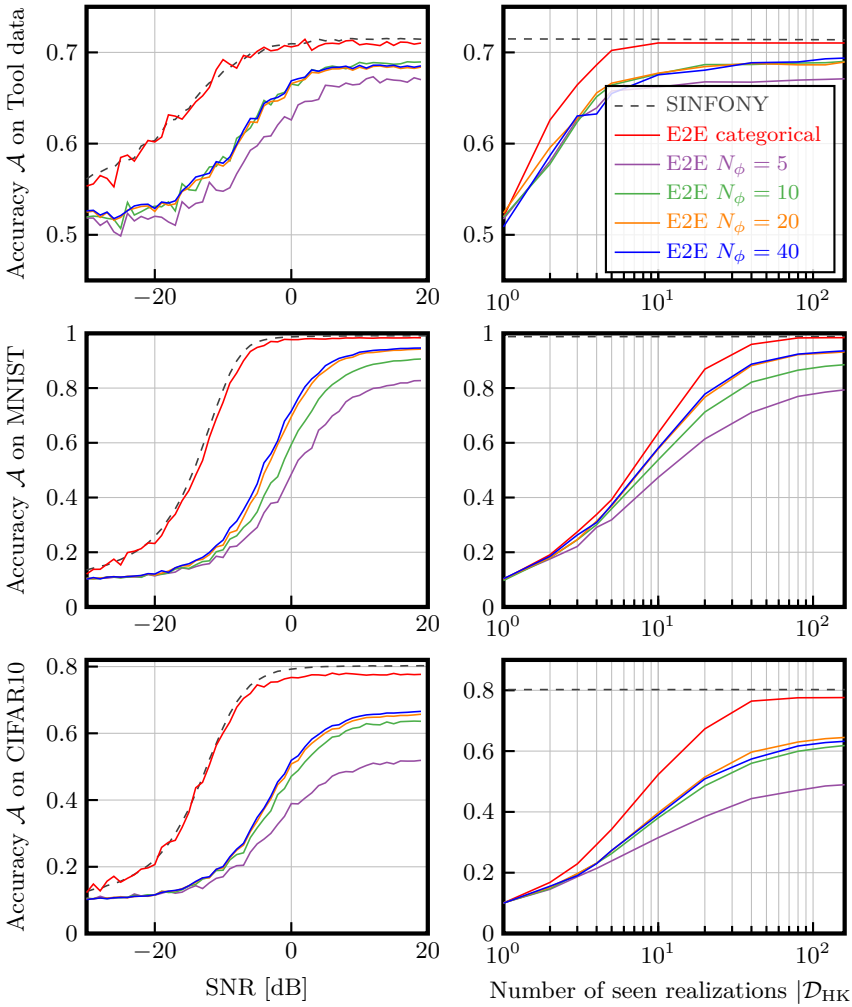

Figure 6.5: The simulated performance of the proposed end-to-end sensing-decision framework, including SINFONY and human decisions modeled by the GCM. Each column shows the classification accuracy on different datasets. From left to right: Tools, MNIST, and CIFAR10. The top row shows the accuracy as a function of SNR. The bottom row shows the simulated accuracy as a function of the number of seen realizations. Within each figure, the color of the lines indicates the number of features presented to the GCM.

behavior is consistent with the bias-variance trade-off from statistical learning, which explains why low-capacity models generalize better with limited data [43]: GCMs with fewer parameters constrain the hypothesis space of solutions, effectively regularizing the learning process. In contrast, high-capacity GCMs with more inputs and attention weights tend to overfit to a limited knowledge base $\mathcal{D}_{\mathrm{HK}}$ based on a few seen realizations. We conclude that providing more features, i.e., details, to the decision-maker with a small knowledge base $\mathcal{D}_{\mathrm{HK}}$ can lead to a suboptimal decision compared to providing less information.

Main Conclusions

Recalling the design trade-off (6.20) from Sec. 6.3.6, we conclude from both simulation scenario observations that it is more important to match the cognitive capabilities of the GCM by a low-dimensional presentation, i.e, more elaborate SINFONY preprocessing, in this task than providing more relevant information about $\mathbf{z}$ by raw decision features.

However, providing more features instead of the final SINFONY output can have other benefits. For one, we have a slight reduction in processing complexity, since some nodes are removed. Also, in this case, SINFONY is optimized for solving a single task, i.e., deciding whether a tool needs to be replaced or not. In more complex situations, however, HDM may need to deal with unexpected events or changing goals not covered by the current form of our end-to-end sensing-decision framework. A more detailed representation allows the HDM model to react to these changes compared to the probability estimates. Finally, the study focused on decision accuracy alone, using a simulated decision-maker. With human decision-makers, motivational and emotional aspects such as experienced autonomy and competence will affect performance in addition to information processing ability [74], as we will discuss in the outlook in Sec. 6.5.

The main takeaways provide answers to the research questions stated at the beginning:

b1) *Which information should semantic communication provide for accurate HDM?* The semantic information provided by the SINFONY architecture supports HDM, as motivated by the InfoMax principle. However, using raw decision features leads to imperfect information integration of the HDM model compared to SINFONY, evident through saturation in the simulations. This is consistent with the derived design trade-off and shows that it is more important to match the cognitive capabilities of the GCM by more elaborate SINFONY feature extraction than providing more relevant information.

b2) *How much information should semantic communication provide for accurate HDM?* Providing more detailed representations, i.e., more features, does not always increase HDM decision accuracy. The saturation indicates that the HDM model at some point misses subtle details in the additional features. The effect of extra features also depends on the context. For example, with little expertise, more information can misguide instead of help the HDM model which requires carefully balancing the design trade-off, i.e., the information provided by semantic communication with the HDM process.

c) *How does the HDM process impact the end-to-end sensing-decision-making process?* The accuracy of the HDM model can be inferior to that of semantic communication systems due to the probabilistic nature of HDM.

6.5 Outlook – Open Questions and Challenges

The proposed end-to-end sensing-decision framework is a first step towards integrating semantic communication and the human receiver. We will now explore remaining open questions and challenges with respect to all our framework components from Fig. 6.3, and what they mean for semantic communication.

6.5.1 Challenge: Optimization of Semantic Communication for Human Decisions

In this article, we have examined how semantic communication affects the decisions of a HDM model in theory and simulations. There remains the question of how semantic communication can be optimized directly for the given human or HDM model to improve decisions.

One idea is to include the human or HDM model in the optimization process. This idea is supported in theory by our extension of the information-theoretic framework on semantic communication via (6.20) from Sec. 6.3.6, originally aimed to understand both presentation design and HDM model training given a fixed optimized semantic communication system. Including the semantic encoder parameters $\boldsymbol{\theta}$ in maximization of the MILBO (6.20) allows for joint end-to-end optimization of all components with respect to all framework parameters $\boldsymbol{\theta}, \boldsymbol{\theta}_{\mathrm{P}}, \boldsymbol{\varphi}_{\mathrm{G}}$, leading to a unified design.

However, since both the human and the HDM model are essentially a black box that is not known or differentiable in practice as assumed in this work, this seems difficult. However, it is possible to evaluate the cross-entropy

loss or another target metric for the human or HDM model decision and feed it back to SINFONY as a reward. So one idea could be to use the stochastic policy gradient as in [21] to allow optimization over the whole chain, including SINFONY and the human/HDM model.

6.5.2 Challenge: Limitations of Human Decision-Making Models

To include the HDM model in an optimization process, it is essential to have an accurate representation of the HDM process. Here, we simulated the decision-making process by applying a computational model, the GCM, for illustrative purposes. While our simulation reflects many traits of HDM with human participants, not all assumptions will apply to realistic categorization decisions. For instance, given the lack of HDM data for the simulated tasks, we assumed perfect memory and optimal performance on the training set for the GCM. These assumptions are difficult to achieve in a real-life situations, but more appropriate assumptions are likely to strongly depend on individuals and tasks.

Accordingly, just like in most of psychological research on HDM, future HDM models will need to be carefully selected and designed to accurately reflect human decision processes for the task of interest, e.g., by accounting for limitations in human information retrieval [53], [75] and contextual influences on the decision process such as limited cognitive resources due to multitasking, acute stress, or time pressure, e.g., [56], [57]. As outlined in Sec. 6.3.4, it is possible to extend the basic GCM (6.10) used in this work to model many of these HDM aspects. Nevertheless, experimental validation with human participants, typical for psychological experiments, is required to develop and validate appropriate HDM models for the decision problem at hand, assess the beneficial effects of semantic communication and their visualizations, and understand the trade-offs between performance facilitation and potential negative impacts on motivation.

6.5.3 Challenge: Presentation of Semantic Information

For tractability in the simulations and to reflect constraints on realistic cognitive processing, we provided the HDM model with two interpretable, handcrafted presentations containing varying amount of information — probability outputs or the most important decision features of semantic communication. The key question is how this abstract representation $\boldsymbol{\nu} = f(\mathbf{y})$ or process $p(\boldsymbol{\nu}|\mathbf{y})$, shown in Fig. 6.3, translates into real-world scenarios with human subjects. To help humans interpret the output $\mathbf{y}$ or $q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{y})$, it

is essential to present it through a Human-Machine Interface (HMI) that connects human users with semantic communication. In practice, this could involve visualizing, e.g., tool damage probabilities, through symbols and colors on a screen or in augmented/virtual reality, with varying levels of detail (see Fig. 6.2) — from basic machine learning outputs (e.g., tool wear status) to more detailed insights such as algorithm certainty, textual explanations, and even image-based class activation mapping [76].

The HMI is a critical component, as the presentation format can strongly influence the HDM process [77], [78]. It must hence present complex information in a way that enables informed decisions while maintaining essential context. Designing a successful HMI requires an understanding of the domain in which it operates, including the industry, the user base, and the types of decisions being made [79]. A context-aware HMI that adapts to the user's history, preferences, and current situation can further improve decision-making by providing personalized and relevant information [80].

6.5.4 Challenge: Variability in Human Decision Goals and Expertise

For many tasks, human decision-makers will differ in their preferences, intentions, and levels of expertise. In addition, task goals may be multifaceted and subject to change of time, requiring to adjust transmitted information to the individual and the current situation. For example, when communicating information about tools, a person interested in understanding how different machines affect tool usability will need different information than someone focused on identifying tools that need to be replaced. In addition, an expert is likely to prefer a detailed, rich presentation, while a novice may benefit more from clear, concise support.

Furthermore, individual differences can lead to different information even when the decision objective is the same. For example, people differ in how much risk they are willing to accept [81]. Given the same information about the probability that the tool will fail, a risk-seeking person may conclude that the risk is acceptable, while a more risk-averse person would choose to exchange it. Thus, semantic communication that attempts to optimize tool use while keeping the failure rate below a tolerable threshold may require adapting the information conveyed to the decision-maker, for example by changing how potential risks are presented [82], [83]. While the core model (6.10) of the GCM investigated in this work does not accommodate all individual differences, it can be extended to simulate variability.

6.5.5 Challenge: Conflict of Interest between Sender and Receiver

The interests or goals of a human sender may not be well aligned with those of the human receiver. A fundamental factor contributing to such a misalignment of interests could be that human receivers are risk-averse. For example, even if a tool remains functional, the receivers may classify it as defective in order to avoid potential errors, since they are reluctant to take responsibility for using a worn tool. The sender thus has an incentive to manipulate the message in order to influence the receiver's decisions. If the difference in interests is too large, the receiver could ignore any message the sender sends.

This means that successful semantic communication also depends on trust between sender and receiver. Economists, following [84], have long studied this sender-receiver problem using game theory. For a recent overview of this literature, see [85]. They found that the amount of information that can be transmitted depends on how large the difference in interests is. Considering how much the sender wants to manipulate the information to influence the receiver's action is important in semantic communication. Even if the technology allows for very accurate transmission of semantic meaning, the best transmission strategy would still depend on the characteristics of the sender and receiver.

6.6 Conclusion

In this paper, integrating an interdisciplinary perspective from communications and psychology, we proposed a probabilistic end-to-end sensing-decision framework that wirelessly links sensed data with Human Decision-Making (HDM) by semantic communication. We analyzed this integration exemplarily using SINFONY and an effective HDM model based on generalized context models for specific datasets. The theoretical and numerical results indicate that semantic communication can optimize task performance by balancing information detail with human cognitive processes, achieving accurate decisions while demanding less bandwidth, power, and latency compared to classical methods.

This work is intended to inspire further interdisciplinary research on higher semantic levels of communication. Open questions include how to optimize semantic communication for human decisions, how to extend the HDM model, how to design human-machine interfaces that convey meaning more effectively, and how to account for different intentions between sender and receiver as well as individual differences among receivers.

6.7 References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *31st Conference on Advances in Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, Dec. 2017, pp. 6000–6010.

[2] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks," in *16th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2013)*, Nagoya, Japan: Springer, Sep. 2013, pp. 411–418. DOI: 10.1007/978-3-642-40763-5_51.

[3] A. Suresh, E. Beck, A. Dekorsy, P. Rückert, and K. Tracht, "Human-integrated Multi-agent Exploration using Semantic Communication and Extended Reality Simulation," in *10th IEEE International Conference on Automation, Robotics and Applications (ICARA 2024)*, Athens, Greece, May 2024, pp. 419–426. DOI: 10.1109/ICARA60736.2024.10553050.

[4] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, 16th ed. Urbana, IL, USA: The University of Illinois Press, Sep. 1949.

[5] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107 930, May 2021. DOI: 10.1016/j.comnet.2021.107930.

[6] X. Luo, H.-H. Chen, and Q. Guo, "Semantic Communications: Overview, Open Issues, and Future Research Directions," *IEEE Transactions on Wireless Communications*, vol. 29, no. 1, pp. 210–219, Feb. 2022. DOI: 10.1109/MWC.101.2100269.

[7] D. Wheeler and B. Natarajan, "Engineering Semantic Communication: A Survey," *IEEE Access*, vol. 11, pp. 13 965–13 995, Feb. 2023. DOI: 10.1109/ACCESS.2023.3243065.

[8] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, Jan. 2023. DOI: 10.1109/JSAC.2022.3223408.

[9] E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic Information Recovery in Wireless Networks," *Sensors*, vol. 23, no. 14, p. 6347, Jul. 2023. DOI: 10.3390/s23146347.

[10] J. Shao, Y. Mao, and J. Zhang, "Learning Task-Oriented Communication for Edge Inference: An Information Bottleneck Approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 197–211, Jan. 2022. DOI: 10.1109/JSAC.2021.3126087.

[11] R. Carnap and Y. Bar-Hillel, "An Outline of a Theory of Semantic Information," Research Laboratory of Electronics, Massachusetts Institute of Technology, Technical Report 247, Oct. 1952, p. 54.

[12] F. Zhou, Y. Li, M. Xu, L. Yuan, Q. Wu, R. Q. Hu, and N. Al-Dhahir, "Cognitive Semantic Communication Systems Driven by Knowledge Graph: Principle, Implementation, and Performance Evaluation," *IEEE Transactions on Communications*, vol. 72, no. 1, pp. 193–208, Jan. 2024. DOI: 10.1109/TCOMM.2023.3318605.

[13] S. R. Pokhrel and J. Choi, "Understand-Before-Talk (UBT): A Semantic Communication Approach to 6G Networks," *IEEE Transactions on Vehicular Communications*, vol. 72, no. 3, pp. 3544–3556, Mar. 2023. DOI: 10.1109/TVT.2022.3219363.

[14] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *IEEE Network Science Workshop (NSW 2011)*, West Point, NY, USA, Jun. 2011, pp. 110–117. DOI: 10.1109/NSW.2011.6004632.

[15] P. Basu, J. Bao, M. Dean, and J. Hendler, "Preserving Quality of Information by Using Semantic Relationships," *Pervasive and Mobile Computing*, vol. 11, pp. 188–202, Apr. 2014. DOI: 10.1016/j.pmcj.2013.07.013.

[16] 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, Apr. 2021. DOI: 10.1109/TSP.2021.3071210.

[17] Z. Weng, Z. Qin, and G. Y. Li, "Semantic Communications for Speech Signals," in *IEEE International Conference on Communications (ICC 2021)*, Virtual Conference, Jun. 2021, pp. 1–6. DOI: 10.1109/ICC42927.2021.9500590.

[18] Z. Weng, Z. Qin, X. Tao, C. Pan, G. Liu, and G. Y. Li, "Deep Learning Enabled Semantic Communications with Speech Recognition and Synthesis," *IEEE Transactions on Wireless Communications*, vol. 22, no. 9, pp. 6227–6240, Sep. 2023. DOI: 10.1109/TWC.2023.3240969.

[19] F. Jiang, Y. Peng, L. Dong, K. Wang, K. Yang, C. Pan, and X. You, "Large AI Model-Based Semantic Communications," *IEEE Wireless Communications*, vol. 31, no. 3, pp. 68–75, Jun. 2024. DOI: 10.1109/MWC.001.2300346.

[20] H. Cui, Y. Du, Q. Yang, Y. Shao, and S. C. Liew, *LLMind: Orchestrating AI and IoT with LLM for Complex Task Execution*, arXiv preprint: 2312.09007, Aug. 2024. DOI: 10.48550/arXiv.2312.09007.

[21] E. Beck, C. Bockelmann, and A. Dekorsy, "Model-free Reinforcement Learning of Semantic Communication by Stochastic Policy Gradient," in *1st IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2024)*, Stockholm, Sweden, May 2024, pp. 367–373. DOI: 10.1109/ICMLCN59089.2024.10625190.

[22] H. Xie, Z. Qin, X. Tao, and K. B. Letaief, "Task-Oriented Multi-User Semantic Communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 9, pp. 2584–2597, Sep. 2022. DOI: 10.1109/JSAC.2022.3191326.

[23] X. Luo, R. Gao, H.-H. Chen, S. Chen, Q. Guo, and P. N. Suganthan, "Multimodal and Multiuser Semantic Communications for Channel-Level Information Fusion," *IEEE Wireless Communications*, vol. 31, no. 2, pp. 117–125, Apr. 2024. DOI: 10.1109/MWC.011.2200288.

[24] A. Halimi Razlighi, C. Bockelmann, and A. Dekorsy, "Semantic Communication for Cooperative Multi-Task Processing Over Wireless Networks," *IEEE Wireless Communications Letters*, vol. 13, no. 10, pp. 2867–2871, Oct. 2024. DOI: 10.1109/LWC.2024.3451139.

[25] A. Halimi Razlighi, M. H. V. Tillmann, E. Beck, C. Bockelmann, and A. Dekorsy, "Cooperative and Collaborative Multi-Task Semantic Communication for Distributed Sources," in *IEEE International Conference on Communications (ICC 2025)*, Montreal, Canada, Jun. 2025, pp. 1–6. DOI: 10.48550/arXiv.2411.02150.

[26] Y. Blau and T. Michaeli, "Rethinking Lossy Compression: The Rate-Distortion-Perception Tradeoff," in *36th International Conference on Machine Learning (ICML 2019)*, Long Beach, CA, USA: PMLR, May 2019, pp. 675–685. DOI: 10.48550/arXiv.1901.07821.

[27] J. Chai, Y. Xiao, G. Shi, and W. Saad, "Rate-Distortion-Perception Theory for Semantic Communication," in *31st IEEE International Conference on Network Protocols (ICNP 2023)*, Reykjavik, Iceland, Oct. 2023, pp. 1–6. DOI: 10.1109/ICNP59255.2023.10355575.

[28] J. W. Payne, J. R. Bettman, E. Coupey, and E. J. Johnson, "A constructive process view of decision making: Multiple strategies in judgment and choice," *Acta Psychologica*, vol. 80, no. 1-3, pp. 107–141, Aug. 1992. DOI: 10.1016/0001-6918(92)90043-D.

[29] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired Artificial Intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, Jul. 2017. DOI: 10.1016/j.neuron.2017.06.011.

[30] D. Kahneman, "A perspective on judgment and choice: Mapping bounded rationality," in *Progress in Psychological Science around the World*, ser. Neural, Cognitive and Developmental Issues, vol. 1, Psychology Press, Jul. 2013, pp. 1–47. DOI: 10.4324/9780203783122.

[31] B. Englich, T. Mussweiler, and F. Strack, "Playing Dice With Criminal Sentences: The Influence of Irrelevant Anchors on Experts' Judicial Decision Making," *Personality and Social Psychology Bulletin*, vol. 32, no. 2, pp. 188–200, Feb. 2006. DOI: 10.1177/0146167205282152.

[32] M. Thomas and V. Morwitz, "Heuristics in numerical cognition: Implications for pricing," in *Handbook of pricing research in marketing*, Cheltenham, UK: Edward Elgar Publishing, Mar. 2008, pp. 132–149. DOI: 10.4337/9781848447448.00015.

[33] G. Gigerenzer and W. Gaissmaier, "Heuristic Decision Making," *Annual Review of Psychology*, vol. 62, no. 1, pp. 451–482, Jan. 2011. DOI: 10.1146/annurev-psych-120709-145346.

[34] N. Cowan, "The magical number 4 in short-term memory: A reconsideration of mental storage capacity," *Behavioral and Brain Sciences*, vol. 24, no. 1, pp. 87–114, Feb. 2001. DOI: 10.1017/S0140525X01003922.

[35] H. A. Simon, "Bounded Rationality and Organizational Learning," *Organization Science*, vol. 2, no. 1, pp. 125–134, Feb. 1991. DOI: 10.1287/orsc.2.1.125.

[36] G. Gigerenzer and D. G. Goldstein, "Betting on One Good Reason: The Take The Best Heuristic," in *Simple Heuristics That Make Us Smart*, G. Gigerenzer, P. M. Todd, and the ABC Research Group, Eds., New York: Oxford University Press, Apr. 1999, pp. 75–95.

[37] H. Brighton, "Robust inference with simple cognitive models," in *AAAI spring symposium: Between a rock and a hard place: Cognitive science principles meet AI-hard problems*, vol. 15, Menlo Park, CA, USA, Mar. 2006, pp. 17–22.

[38] G. Gigerenzer and D. G. Goldstein, "Reasoning the fast and frugal way: Models of bounded rationality," *Psychological Review*, vol. 103, no. 4, pp. 650–669, Oct. 1996. DOI: 10.1037/0033-295x.103.4.650.

[39] S. Paul and D. L. Nazareth, "Input information complexity, perceived time pressure, and information processing in GSS-based work groups: An experimental investigation using a decision schema to alleviate information overload conditions," *Decision Support Systems*, vol. 49, no. 1, pp. 31–40, Apr. 2010. DOI: 10.1016/j.dss.2009.12.007.

[40] D. Friedman, "Monty Hall's three doors: Construction and deconstruction of a choice anomaly," *The American Economic Review*, vol. 88, no. 4, pp. 933–946, Sep. 1998.

[41] A. N. Sanborn, J.-Q. Zhu, J. Spicer, P. León-Villagrá, L. Castillo, J. K. Falbén, Y.-X. Li, A. Tee, and N. Chater, "Noise in Cognition: Bug or Feature?" *Perspectives on Psychological Science*, vol. 19, no. 1, pp. 123–145, Jan. 2024.

[42] F. I. Seitz, J. B. Jarecki, J. Rieskamp, and B. von Helversen, "Disentangling Perceptual and Process-Related Sources of Behavioral Variability in Categorization," *PsyArXiv*, Sep. 2024. DOI: 10.31234/osf.io/g3bpa.

[43] O. Simeone, "A Brief Introduction to Machine Learning for Engineers," *Foundations and Trends® in Signal Processing*, vol. 12, no. 3-4, pp. 200–431, Aug. 2018. DOI: 10.1561/2000000102.

[44] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, vol. 11, no. 110, pp. 3371–3408, Dec. 2010.

[45] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018. DOI: 10.1109/TCCN.2018.2881442.

[46] D. J. Koehler and N. Harvey, *Blackwell handbook of judgment and decision making*. Malden, MA, USA: John Wiley & Sons, Apr. 2008. DOI: 10.1002/9780470752937.

[47] B. R. Newell, D. A. Lagnado, and D. R. Shanks, *Straight Choices: The Psychology of Decision Making*. London: Psychology Press, May 2022, vol. 3. DOI: 10.4324/9781003289890.

[48] J. R. Anderson, "The adaptive nature of human categorization," *Psychological Review*, vol. 98, no. 3, p. 409, Jul. 1991. DOI: 10.1037/0033-295X.98.3.409.

[49] K. J. Kurtz, "Chapter Three - Human Category Learning: Toward a Broader Explanatory Account," in *Psychology of Learning and Motivation*, B. H. Ross, Ed., vol. 63, Academic Press, Jan. 2015, pp. 77–114. DOI: 10.1016/bs.plm.2015.03.001.

[50] R. Schlegelmilch, A. J. Wills, and B. von Helversen, "A cognitive category-learning model of rule abstraction, attention learning, and contextual modulation," *Psychological Review*, vol. 129, no. 6, pp. 1211–1248, Sep. 2021. DOI: 10.1037/rev0000321.

[51] R. M. Nosofsky, "Choice, similarity, and the context theory of classification," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, no. 1, p. 104, Jan. 1984. DOI: 10.1037/0278-7393.10.1.104.

[52] R. M. Nosofsky, "The generalized context model: An exemplar model of classification," in *Formal Approaches in Categorization*, A. J. Wills and E. M. Pothos, Eds., Cambridge: Cambridge University Press, Mar. 2011, pp. 18–39. DOI: 10.1017/CBO9780511921322.002.

[53] R. M. Nosofsky and T. J. Palmeri, "An Exemplar-Based Random-Walk Model of Categorization and Recognition," in *The Oxford Handbook of Computational and Mathematical Psychology*, J. R. Busemeyer, Z. Wang, J. T. Townsend, and A. Eidels, Eds., New York: Oxford University Press, Dec. 2015, pp. 142–164. DOI: 10.1093/oxfordhb/9780199957996.013.7.

[54] S. Bhatia and N. Stewart, "Naturalistic multiattribute choice," *Cognition*, vol. 179, pp. 71–88, Oct. 2018. DOI: 10.1016/j.cognition.2018.05.025.

[55] B. J. Meagher and R. M. Nosofsky, "Testing formal cognitive models of classification and old-new recognition in a real-world high-dimensional category domain," *Cognitive Psychology*, vol. 145, p. 101 596, Sep. 2023. DOI: 10.1016/j.cogpsych.2023.101596.

[56] K. Lamberts, "Categorization under time pressure," *Journal of Experimental Psychology: General*, vol. 124, no. 2, pp. 161–180, Jun. 1995. DOI: 10.1037/0096-3445.124.2.161.

[57] F. I. Seitz, B. von Helversen, R. Albrecht, J. Rieskamp, and J. B. Jarecki, "Testing three coping strategies for time pressure in categorizations and similarity judgments," *Cognition*, vol. 233, p. 105 358, Apr. 2023. DOI: 10.1016/j.cognition.2022.105358.

[58] J. A. Hoffmann, B. von Helversen, and J. Rieskamp, "Deliberation's blindsight: How cognitive load can improve judgments," *Psychological Science*, vol. 24, no. 6, pp. 869–879, Apr. 2013. DOI: 10.1177/0956797612463581.

[59] D. R. Shanks, "A connectionist account of base-rate biases in categorization," *Connection Science*, vol. 3, no. 2, pp. 143–162, Jun. 1991. DOI: 10.1080/09540099108946582.

[60] E. M. Pothos and T. M. Bailey, "The role of similarity in artificial grammar learning," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 26, no. 4, pp. 847–862, Jul. 2000. DOI: 10.1037/0278-7393.26.4.847.

[61] D. K. Sewell, T. Ballard, and N. K. Steffens, "Exemplifying "Us": Integrating social identity theory of leadership with cognitive models of categorization," *The Leadership Quarterly*, vol. 33, no. 4, p. 101 517, Aug. 2022. DOI: 10.1016/j.leaqua.2021.101517.

[62] T. L. Griffiths and M. L. Kalish, "A multidimensional scaling approach to mental multiplication," *Memory & Cognition*, vol. 30, no. 1, pp. 97–106, Jan. 2002. DOI: 10.3758/BF03195269.

[63] R. M. Nosofsky and T. J. Palmeri, "An exemplar-based random walk model of speeded classification," *Psychological Review*, vol. 104, no. 2, pp. 266–300, Apr. 1997. DOI: 10.1037/0033-295X.104.2.266.

[64] M. D. Lee and D. J. Navarro, "Extending the alcove model of category learning to featural stimulus domains," *Psychonomic Bulletin & Review*, vol. 9, no. 1, pp. 43–58, Mar. 2002. DOI: 10.3758/BF03196256.

[65] R. D. Luce, "On the possible psychophysical laws," *Psychological Review*, vol. 66, no. 2, p. 81, Jul. 1959. DOI: 10.1037/h0043178.

[66] Y. Zhang, L. Wang, and Q. Wu, "Differential Annealing for Global Optimization," in *Advances in Swarm Intelligence: Third International Conference on Swarm Intelligence (ICSI 2012)*, ser. Lecture Notes in Computer Science, vol. 7331, Berlin, Heidelberg: Springer, Jun. 2012, pp. 382–389. DOI: 10.1007/978-3-642-30976-2_46.

[67] T. Serre, "Models of Visual Categorization," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, no. 3, pp. 197–213, May 2016. DOI: 10.1002/wcs.1385.

[68] K. N. Prasad and B. Ramamoorthy, "Tool Wear Evaluation by Stereo Vision and Prediction by Artificial Neural Network," *Journal of Materials Processing Technology*, vol. 112, no. 1, pp. 43–52, May 2001. DOI: 10.1016/S0924-0136(00)00896-7.

[69] B. Papenberg, S. Hogreve, T. Bochardt, C. Bornholdt, T. Heinrich, and K. Tracht, "Influence of Illumination on the Image-Based Classification Accuracy of Wear on Milling Tools," in *17th CIRP Conference on Intelligent Computation in Manufacturing Engineering (ICME 23)*, Ischia, Italy, Jul. 2023, pp. 366–371. DOI: 10.1016/j.procir.2024.08.377.

[70] E. Beck, *Semantic Information Transmission and Recovery (SINFONY) Software*, Zenodo, version v1.2.2, Dec. 2024. DOI: 10.5281/zenodo.8006567.

[71] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *14th European Conference on Computer Vision (ECCV 2016)*, ser. Lecture Notes in Computer Science, Amsterdam, Netherlands, Oct. 2016, pp. 630–645. DOI: 10.1007/978-3-319-46493-0_38.

[72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[73] R. M. Battleday, J. C. Peterson, and T. L. Griffiths, "Capturing human categorization of natural images by combining deep networks and cognitive models," *Nature Communications*, vol. 11, no. 1, p. 5418, Oct. 2020. DOI: 10.1038/s41467-020-18946-z.

[74] R. M. Ryan and E. L. Deci, "Self-Regulation and the Problem of Human Autonomy: Does Psychology Need Choice, Self-Determination, and Will?" *Journal of Personality*, vol. 74, no. 6, pp. 1557–1586, Dec. 2006. DOI: 10.1111/j.1467-6494.2006.00420.x.

[75] N. Stewart, N. Chater, and G. D. A. Brown, "Decision by sampling," *Cognitive Psychology*, vol. 53, no. 1, pp. 1–26, Aug. 2006. DOI: 10.1016/j.cogpsych.2005.10.003.

[76] S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, "Steel Surface Defect Diagnostics Using Deep Convolutional Neural Network and Class Activation Map," *Applied Sciences*, vol. 9, no. 24, p. 5449, Dec. 2019. DOI: 10.3390/app9245449.

[77] S. T. Hawley, B. Zikmund-Fisher, P. Ubel, A. Jancovic, T. Lucas, and A. Fagerlin, "The impact of the format of graphical presentation on health-related knowledge and treatment choices," *Patient Education and Counseling*, 4th International Conference on Shared Decision Making (ISDM 2007), vol. 73, no. 3, pp. 448–455, Nov. 2008. DOI: 10.1016/j.pec.2008.07.023.

[78] A. Strathie, G. Netto, G. H. Walker, and G. Pender, "How presentation format affects the interpretation of probabilistic flood risk information," *Journal of Flood Risk Management*, vol. 10, no. 1, pp. 87–96, Mar. 2017. DOI: 10.1111/jfr3.12152.

[79] K. Nikolova, "Framework and Practical Guidelines for Creating a Current State Analysis for (Re)Designing an HMI within the Industry 4.0 Era: Case Study," M.S. thesis, University of Twente, Enschede, Netherlands, Aug. 2022.

[80] S. Meftah, M. Sahnoun, M. Messaadia, and S. M. Benslimane, "Context Aware Human Machine Interface for Decision Support," in *International Conference on Cyber Management and Engineering (CyMaEn 2023)*, Bangkok, Thailand, Jan. 2023, pp. 143–147. DOI: 10.1109/CyMaEn57228.2023.10051078.

[81] C. Wang, S. M. O'Neill, N. Rothrock, R. Gramling, A. Sen, L. S. Acheson, W. S. Rubinstein, D. E. Nease, and M. T. Ruffin, "Comparison of risk perceptions and beliefs across common chronic diseases," *Preventive Medicine*, vol. 48, no. 2, pp. 197–202, Feb. 2009. DOI: 10.1016/j.ypmed.2008.11.008.

[82] C. Franco, J. L. Hougaard, and K. Nielsen, "Handling Risk Attitudes for Preference Learning and Intelligent Decision Support," in *Modeling Decisions for Artificial Intelligence*, ser. Lecture Notes in Computer Science, V. Torra and T. Narukawa, Eds., vol. 9321, Springer International Publishing, Sep. 2015, pp. 78–89. DOI: 10.1007/978-3-319-23240-9_7.

[83] J. Liesiö, M. Kallio, and N. Argyris, "Incomplete Risk-Preference Information in Portfolio Decision Analysis," *European Journal of Operational Research*, vol. 304, no. 3, pp. 1084–1098, Feb. 2023. DOI: 10.1016/j.ejor.2022.04.043.

[84] V. P. Crawford and J. Sobel, "Strategic Information Transmission," *Econometrica*, vol. 50, no. 6, pp. 1431–1451, Nov. 1982. DOI: 10.2307/1913390.

[85] A. Blume, E. K. Lai, and W. Lim, "Chapter 13 Strategic information transmission: a survey of experiments and theoretical foundations," in *Handbook of Experimental Game Theory*. Cheltenham, UK: Edward Elgar Publishing, Oct. 2020. DOI: 10.4337/9781785363337.00022.

Chapter 7

Conclusion

New Artificial Intelligence (AI) technologies find their way into everyday life enabling many new possibilities such as medical diagnosis, chat assistance, and autonomous driving. The question arises of how these recent Machine Learning (ML) techniques can be leveraged in wireless communications with its well-established channel models and high-end designed standards. In this thesis, this question was tackled identifying that algorithm and model deficit are indicative of beneficial usage. These deficits motivate the split of this thesis into two parts, respectively: Improving digital communication (Part I) and enabling semantic communication (Part II). Further, we laid the theoretical foundation for a unified view of both design problems through the lens of information maximization.

In Chapter 2, we introduce key Machine Learning (ML) concepts like amortized Monte Carlo (MC) Variational Inference (VI) and Deep Neural Network (DNN), linking them to communication design through information theory. A central contribution is the use of the Information Maximization (InfoMax) criterion as a unified learning framework for receiver and transceiver design. Building on this unified view, we reflect on the approaches employed throughout this thesis in the broader context of ML theory, thereby motivating both their choice and possible alternatives.

In Chapter 3, we propose a hybrid approach to address the algorithm deficit in soft detection for large systems like digital massive Multiple Input Multiple Output (MIMO). First, we introduce a continuous relaxation of the prior probability mass function (pmf) into a probability density function (pdf) for Maximum A Posteriori (MAP) detection, enabling computationally cheaper optimization via gradient descent. We then combine model- and ML-based techniques to create Concrete MAP Detection Network (CMDNet), a

deep unfolding model that optimizes detection accuracy while maintaining low complexity. Using an information-theoretic approach, we derive the optimization criterion from the Kullback–Leibler (KL) divergence and show that CMDNet can learn an approximation of the ideal detector reducing computational cost. Numerical results demonstrate CMDNet's effectiveness and reliability, providing accurate soft outputs for MIMO decoders, outperforming recent ML-based methods, and offering a strong trade-off between accuracy and complexity.

In Appendix A, we extend the analysis of CMDNet, providing a full derivation of binary Concrete MAP Detection (CMD) and proving that CMD and binary CMD are distinct algorithms for Binary Phase Shift Keying (BPSK) symbols. We systematically explore the optimization process for CMDNet, discussing key ML aspects like hyperparameters, optimization algorithms, batch size, and layer depth. Our analysis highlights how communications design challenges standard ML practices, and vice versa, such as tracking of the validation loss, revealing that suboptimal approaches, such as using Adaptive Moment Estimation (Adam) or Mean Square Error (MSE) optimization, can yield comparable results to theoretically well-motivated approaches.

In Chapter 4, we contribute to theoretical modeling and problem formulation in semantic communication by incorporating a semantic source into the full communications Markov chain, extending information theory to include semantic information. Using the InfoMax principle and the Information Bottleneck (IB) as design criteria, we propose a unified approach that integrates all levels of communication to preserve meaning. Unlike existing IB-based methods, we maximize Mutual Information (MI) for a fixed encoder output. We introduce Semantic INFOrmation TraNsmission and RecoverY (SINFONY), a data-driven method addressing the model deficit in semantic communication, demonstrating an improved trade-off between data rate and energy efficiency over classic digital communication in distributed multipoint scenarios.

In Chapter 5, we tackle the practical problems of purely data-driven approaches like SINFONY in semantic communication, especially when adapting to unknown channels. We optimize the transmitter and receiver separately using Reinforcement Learning (RL) with Stochastic Policy Gradient (SPG), derived from the InfoMax principle without needing a known or differentiable channel model. This allows for online refinement of the semantic design once deployed. Numerical evaluations in the distributed SINFONY scenario show performance comparable to channel model-aware methods, albeit with slower convergence.

In Appendix B, we delve deeper into semantic communication, incor-

porating philosophical and interdisciplinary insights such that meaning (semantics) is linked to emergence in the universe, with multiple hierarchical levels. With extended numerical evaluations, we demonstrate a more pronounced performance advantage of SINFONY compared to traditional digital communications in more complex scenarios and validate our design choice, e.g., finding minimal gains with separate receiver modules. A key insight is the semantic information barrier caused by hard Variable-Length Codes (VLC), suggesting the removal of block-wise structures, as seen in SINFONY.

In Chapter 6, we propose a probabilistic end-to-end sensing-decision framework that wirelessly links sensed data from assistance systems with Human Decision-Making (HDM) by semantic communication. By integrating perspectives from communications and psychology, our interdisciplinary framework enhances understanding of how semantic communication impacts HDM and improves decision-making effectiveness in human-assisted tasks. To investigate this interplay, we model HDM as a cognitive process and reveal both in theory and simulations the fundamental design trade-off between maximizing the relevant semantic information and matching the cognitive capabilities of the HDM model. Using the examples of SINFONY and a HDM model, i.e., the Generalized Context Model (GCM), we demonstrate how semantic communication balances information detail in feature extraction with human cognitive capabilities, achieving accurate decisions while demanding less bandwidth, power, and latency compared to classical digital Shannon-based methods.

7.1 Open Questions and Future Work

Despite the development of ML-based algorithms for communications design, several open challenges remain. The following questions present opportunities for future exploration beyond the scope of this thesis:

- In Appendix A, we reveal that CMDNet can be extended easily to large-scale MIMO systems with sparse activity patterns and nonlinear system models of other research domains. Moreover, joint optimization of CMDNet soft detector and channel decoder could lead to overall systems' performance improvements. Numerical evaluation of the effectiveness of the latter approaches remains an open question. Further, a deeper online training analysis could shed light on its training efficiency.

- Moreover, the large hyperparameter space makes it difficult to examine the true potential of the ML approaches. This challenge may be tackled

with a more efficient random search.

- Other promising ML application areas include detection in highly quantized systems, semi-supervised learning for channel estimation, and precoding at the transmitter.

- One of the initial plans was to conduct a practical evaluation of the developed methods under real conditions using a Software Defined Radio (SDRadio). The motivation is that data-driven ML approaches can mitigate the non-idealities present in traditional expert models of wireless communication. A comparison with expert-designed systems could reveal additional advantages and limitations of these new methods.

- An early idea that ended after algorithm design was to apply variational inference in a non-linear forward model, e.g., with a non-linear amplifier. This enables Bayesian inference in non-linear systems using an approximation with the aid of the first-order Taylor series [CGWW09]. The resulting iterative scheme, which is similar to the non-linear Least Squares (LS) approach, can also be applied to dynamic systems.

- While this work primarily focuses on wireless communications, fiber-optical links can also benefit significantly from ML application. Fiber-optical channel models are typically described by differential equations. But non-idealities such as time-continuity are difficult to capture. To estimate these channels more accurately, neural ordinary differential equations [CRBD18] could be exploited. This approach leverages existing ordinary differential equation solvers and can be interpreted as a continuous version of ResNet.

- In this work, we contributed to the theoretical description of semantic communication and data-driven ML using Deep Neural Networks (DNNs). Open questions remain, such as the benefits of solving the variational IB problem over our proposed method, developing effective semantic models with solution approaches such as Bayesian inference and deep unfolding, and addressing the impact of mismatched knowledge bases on system performance.

- In this thesis, we have primarily focused on processing a single meaning or task in semantic communication based on one type of observations, i.e., images. However, real-world scenarios require managing multiple-tasks for efficient implementation. One design approach is presented in [HBD24; HTB+25]. Further, numerous sensors types are employed including, e.g., camera and temperature sensors, delivering

multimodal data. It remains the question of how to seamlessly fuse and interpret the multimodal data in semantic communication.

- In Chapter 5, Reinforcement Learning-based SINFONY (RL-SINFONY) suffers from slow training convergence. Gradient variance reduction techniques or application of the Deep Deterministic Policy Gradient (DDPG) could help to alleviate this problem.
- The analysis of the end-to-end sensing-decision-making framework in Chapter 6 highlights critical areas for future exploration of the interplay between communication and HDM, such as optimizing semantic communication for human decisions, extending the HDM model, designing Human-Machine Interfaces (HMIs) that more effectively convey meaning, and addressing sender-receiver conflicts and individual differences among users.

The future in the field of AI will remain exciting and offers a lot of potential. We personally believe that this potential will also be further exploited in the field of communications and that AI will find its way into new wireless communication standards. However, ML approaches need to be cleverly tailored to the communication problem so that an improvement can be achieved compared to previous standards.

Part III

Extensions and End Matter

Appendix A

CMDNet Extensions

A.1 Overview

This chapter extends the article [BBD21], corresponding to Chapter 3, by further detailed explanations such as a complete derivation of CMDNet for the special case of binary Random Variables (RVs). Moreover, we contribute deeper insights on CMDNet training revealing common ML guidelines and how these can be transferred to problems of the communications domain shedding new light on them. Finally, we propose and investigate algorithm extensions meant to enhance CMDNet's performance.

A.2 Extended CMDNet Analysis and Explanation

In this section, we provide more details regarding a collection of important aspects that were abbreviated or cut in [BBD21] including a detailed explanation of the correlated massive MIMO channel model in Appendix A.2.2.

Non-convexity of the Objective Function: In Sec. 3.3.3, we show that both using the concrete distribution and applying the reparametrization trick results in a non-convex objective function. Here, we like to provide more details and visualization.

We note that from non-convexity of the relaxed MAP objective $-\ln p(\mathbf{y}|\tilde{\mathbf{x}}) - \ln p(\tilde{\mathbf{x}})$ in (3.9) does not directly follow non-convexity of $L(\mathbf{G}, \tau)$. But indeed we are able to find examples where $L(\mathbf{G}, \tau)$ is non-convex. For

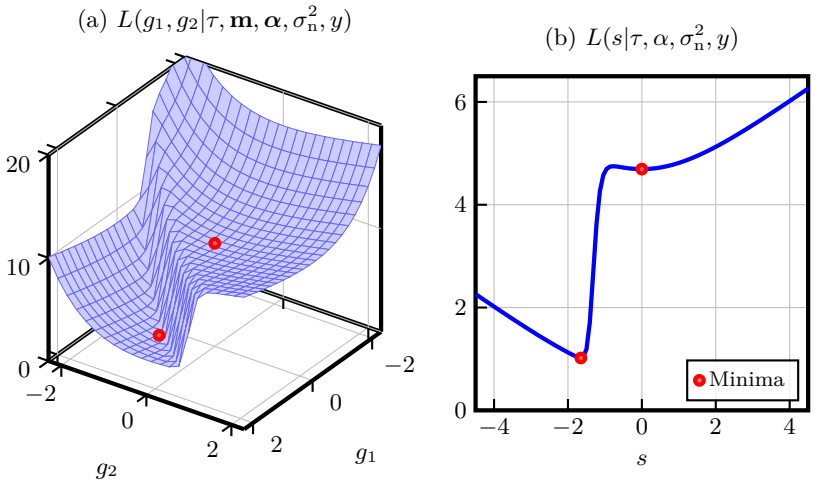

Figure A.1: (a) Example of the surface (blue) of the objective function $L(g_1, g_2|\tau, \mathbf{m}, \boldsymbol{\alpha}, \sigma_n^2, y)$ for $y = -1$, $\alpha_1 = 0.2$, $\alpha_2 = 0.8$, $\tau = 0.1$, $\sigma_n^2 = 0.25$, $m_1 = -1$ and $m_2 = 1$. The minima are indicated as red points. (b) Example of the objective function $L(s|\tau, \alpha, \sigma_n^2, y)$ for the special case of binary RVs for $y = -1$, $\alpha = 0.2$, $\tau = 0.1$ and $\sigma_n^2 = 0.25$. Since there are multiple minima in (a) and (b), the functions are non-convex.

real-valued model (3.1), $N_T = 1$, $M = 2$, $m_1 \neq m_2$ and $\mathbf{H} = 1$, we have:

$$\begin{aligned} L(\mathbf{G}, \tau) =& L(g_1, g_2|\tau, \mathbf{m}, \boldsymbol{\alpha}, \sigma_n^2, y) \\ =& \frac{1}{\sigma_n^2} \cdot \left(y - \frac{m_1 \cdot e^{(\ln(\alpha_1)+g_1)/\tau} + m_2 \cdot e^{(\ln(\alpha_2)+g_2)/\tau}}{e^{(\ln(\alpha_1)+g_1)/\tau} + e^{(\ln(\alpha_2)+g_2)/\tau}} \right)^2 \\ & + g_1 + g_2 + e^{-g_1} + e^{-g_2} \,. \end{aligned} \tag{A.1}$$

The first term is a vertically shifted two-dimensional sigmoid function with respect to (w.r.t.) g_1 and g_2 being squared and scaled. The operations applied to the sigmoid do not change non-convexity. Also, the sum of this non-convex term and convex functions, i.e., linear and exponential functions, remains non-convex. In Fig. A.1 (a), we illustrate one example for $y = -1$, $\alpha_1 = 0.2$, $\alpha_2 = 0.8$, $\tau = 0.1$, $\sigma_n^2 = 0.25$, $m_1 = -1$ and $m_2 = 1$. Note that a detailed mathematical analysis or proof in which cases the objective function is non-convex, convex or invex goes beyond the scope of this thesis. As a final remark, non-convexity may also hold for the special case of binary RVs as illustrated in Fig. A.1 (b).

AMP Performance Degradation at High SNR: In [BBD21] or Sec. 3.5.6, we note that the output statistics of Approximate Message Passing (AMP) become unreliable for high Signal-to-Noise Ratio (SNR) in finite dimensional systems. Here, we explain this aspect in more detail.

In fact, LArge MIMO Approximate message passing (LAMA) from [JGMS15; JGMS18] is able to achieve the error-rate performance of the Individual Optimal (IO) detector under certain assumptions. These assumptions include the large system limit, i.e., $\beta = N_{\mathrm{T}}/N_{\mathrm{R}}$ with $N_{\mathrm{T}} \to \infty$, since then interference or equalized interference of a huge number of transmit symbols can be assumed to be Gaussian-distributed according to the central limit theorem.

Therefore, the estimates obtained by LAMA/AMP correspond to the true signal perturbed by independent and identically distributed (i.i.d.) Gaussian noise. The smaller the finite-dimensional system dimensions, the Gaussian approximation becomes less accurate and the statistics computed by AMP become less reliable, especially in the interference-limited region where noise is small (high SNR). At low SNR, where external Gaussian noise mostly limits performance, interference plays a minor role and AMP is able to denoise translating into excellent performance.

For a more detailed explanation, we refer the reader to [JGMS15; JGMS18].

Numerical Results of the Massive MIMO Simulation with i.i.d. Channel: In [BBD21] and Sec. 3.5.5, we just shortly report the results of a 64×32 massive MIMO system with Quadrature Phase Shift Keying (QPSK) modulation for i.i.d. Gaussian channel taps. For completeness, in the thesis, we also provide the numerical results in Fig. A.2.

CMDNet and Massive MIMO Network (MMNet) are trained for $E_{\mathrm{b}}/N_0 \in [4, 11]$ like Detection Network (DetNet). Being more complex, the curves of OAMP Network (OAMPNet) and SemiDefinite Relaxation (SDR) are not shown. AMP runs into a noticeable error floor at 10 dB. The BER curves of learning based approaches and SDR follow that of Sphere Detector (SD) very closely.

As noted in Sec. 3.5.5, the close performance of all approaches motivates the numerical evaluation with more realistic and challenging channel models such as the One-Ring model with correlated channel taps used in Sec. 3.5.5 and described in Appendix A.2.2.

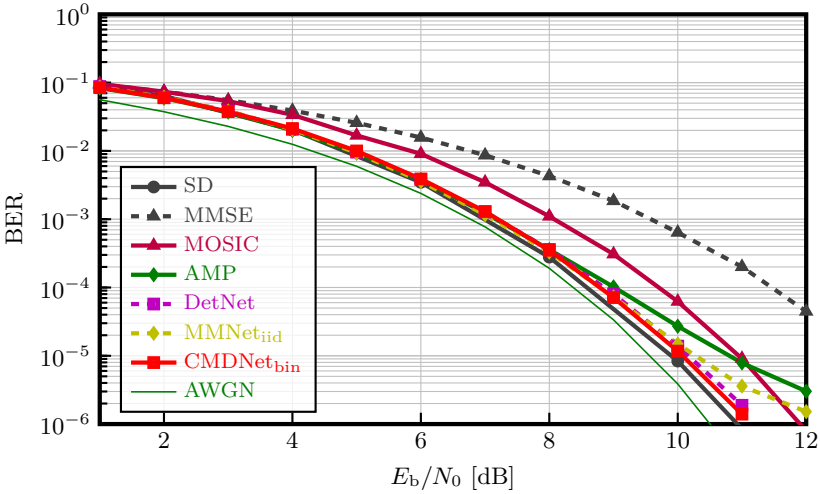

Figure A.2: Bit Error Rate (BER) curves of various detection methods in a 64×32 massive MIMO system with QPSK modulation. Effective system dimension is 128×64 and for iterative algorithms $N_{it} = N_L = 64$.

A.2.1 Extended Derivation for the Special Case of Binary Random Variables

In [BBD21] and Sec. 3.3.5, we briefly summarize the derivation result of binary CMD. In this section, we provide a more detailed derivation of [BBD20] assuming a real-valued system model with BPSK modulation as an example at specific steps.

Noting that the softmax function (3.7) is normalized, we are able to eliminate one degree of freedom in matrix $\mathbf{G} \in \mathbb{R}^{M\times N_T}$ along dimension M. For the special case of binary RVs, i.e., $M = 2$ classes, this means that the matrix $\mathbf{G}$ can be reduced to a vector $\mathbf{s} \in \mathbb{R}^{N_T\times 1}$ of logistic RVs to derive a different algorithm of low complexity.

First, we eliminate one variable in $\tilde{\mathbf{z}}_n$ and assume a symmetric BPSK modulation, i.e., $\mathbf{m} = [-1, 1]^T$:

$$\tilde{x}_n(\mathbf{g}_n) = \tilde{\mathbf{z}}_n^T\mathbf{m} = \begin{bmatrix}\tilde{z}_{1n} & \tilde{z}_{2n}\end{bmatrix} \cdot \begin{bmatrix}-1\\ 1\end{bmatrix} = \begin{bmatrix}\tilde{z}_{1n} & 1-\tilde{z}_{1n}\end{bmatrix} \cdot \begin{bmatrix}-1\\ 1\end{bmatrix}$$
$$= -2\tilde{z}_{1n} + 1\,. \tag{A.2}$$

Second, $\tilde{z}_{1n}$ still depends on two variables. We now reduce this to one. By

rewriting

$$\tilde{z}_{1n} = \sigma_\tau \left([g_{1n}, g_{2n}]^T \right) = \frac{e^{\frac{\ln \alpha_1 + g_{1n}}{\tau}}}{e^{\frac{\ln \alpha_1 + g_{1n}}{\tau}} + e^{\frac{\ln \alpha_2 + g_{2n}}{\tau}}} = \frac{1}{1 + e^{\frac{\ln \alpha_2 - \ln \alpha_1 + g_{2n} - g_{1n}}{\tau}}}, \tag{A.3}$$

we notice that the difference of two i.i.d. Gumbel RVs $s_n = g_{2n} - g_{1n}$ is distributed according to the logistic distribution $p(s_n) = \exp(-s_n)/(1 + \exp(-s_n))^2$. By transforming the two Gumbel RVs g_{2n} and g_{1n} into one RV s_n and making use of $\alpha = \alpha_1 = 1 - \alpha_2$, we have

$$\tilde{z}_{1n} = \frac{1}{1 + e^{\frac{\ln(1/\alpha - 1) + s_n}{\tau}}} \tag{A.4}$$

$$= \rho \left(-\frac{\ln(1/\alpha - 1) + s_n}{\tau} \right). \tag{A.5}$$

Finally, we combine (A.2) and (A.4) to arrive at

$$\tilde{\mathbf{x}}(\mathbf{s}) = \tanh \left(\frac{\ln(1/\alpha - 1) + \mathbf{s}}{2\tau} \right). \tag{A.6}$$

This means we only need to determine one variable $\mathbf{s}$. Now, we are able to reparametrize the objective function for binary RVs in terms of logistic RVs $\mathbf{s} \in \mathbb{R}^{N_\mathrm{T} \times 1}$ with the new prior pdf $p(\mathbf{s})$:

$$L(\mathbf{s}, \tau) = -\ln p(\mathbf{y}|\mathbf{s}) - \sum_{n=1}^{N_\mathrm{T}} \ln p(s_n) \tag{A.7}$$

$$= -\ln p(\mathbf{y}|\mathbf{s}) + \mathbf{1}^T \mathbf{s} + 2 \cdot \mathbf{1}^T \ln\left(1 + e^{-\mathbf{s}}\right) \tag{A.8}$$

$$\overset{(3.1)}{=} \frac{1}{2\sigma_\mathrm{n}^2} (\mathbf{y} - \mathbf{H}\tilde{\mathbf{x}}(\mathbf{s}))^T (\mathbf{y} - \mathbf{H}\tilde{\mathbf{x}}(\mathbf{s})) + \frac{N_\mathrm{R}}{2} \ln(2\pi\sigma_\mathrm{n}^2) + \mathbf{1}^T \mathbf{s} + 2 \cdot \mathbf{1}^T \ln\left(1 + e^{-\mathbf{s}}\right). \tag{A.9}$$

The prior pdf acts like a regularization contributing the second term. The first term is finally derived in (A.9) for a linear Gaussian model (3.1).

Analogously to (3.14), we derive the gradient descent step of binary CMD:

$$\mathbf{s}^{(j+1)} = \mathbf{s}^{(j)} - \delta^{(j)} \cdot \left.\frac{\partial L(\mathbf{s},\tau)}{\partial \mathbf{s}}\right|_{\mathbf{s}=\mathbf{s}^{(j)}} \tag{A.10a}$$

$$\frac{\partial L(\mathbf{s},\tau)}{\partial \mathbf{s}} = -\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \frac{\partial \ln p(\mathbf{y}|\mathbf{s})}{\partial \tilde{\mathbf{x}}} + \tanh\left(\frac{\mathbf{s}}{2}\right) \tag{A.10b}$$

$$\overset{(3.1)}{=} \frac{1}{\sigma_{\mathrm{n}}^2} \cdot \frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \left[\mathbf{H}^T\mathbf{H}\tilde{\mathbf{x}}(\mathbf{s}) - \mathbf{H}^T\mathbf{y}\right] + \tanh\left(\frac{\mathbf{s}}{2}\right) \tag{A.10c}$$

$$\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} = \frac{1}{2\tau^{(j)}} \cdot \operatorname{diag}\left\{1 - \tilde{\mathbf{x}}^2(\mathbf{s})\right\} \tag{A.10d}$$

$$\tilde{\mathbf{x}}(\mathbf{s}) = \tanh\left(\frac{\ln\left(1/\alpha - 1\right) + \mathbf{s}}{2\tau^{(j)}}\right) . \tag{A.10e}$$

The final step consists again of quantization of $\tilde{\mathbf{x}}$ with the quantization operator $\mathcal{Q}_{\mathcal{M}}\{\tilde{\mathbf{x}}\}$ onto the set of symbols $\mathcal{M}$. In this case, (3.15) simplifies to the sign function:

$$\hat{\mathbf{x}} = \mathcal{Q}_{\mathcal{M}}\left\{\tilde{\mathbf{x}}\left(\mathbf{s}^{(N_{\mathrm{it}})}\right)\right\} = \underset{\mathbf{x}\in\mathcal{M}^{N_{\mathrm{T}}\times 1}}{\arg\min} \left\|\mathbf{x} - \tilde{\mathbf{x}}\left(\mathbf{s}^{(N_{\mathrm{it}})}\right)\right\|_2 \tag{A.11}$$

$$= \operatorname{sign}\left(\tilde{\mathbf{x}}\left(\mathbf{s}^{(N_{\mathrm{it}})}\right)\right) . \tag{A.12}$$

As a concluding remark, comparing (A.10) with (3.14), we note that it is not clear whether both algorithms — CMD and the special binary version of CMD — are different for $M = 2$ classes.

The following theorem answers this question.

Theorem 1 (CMD $\neq$ binary CMD)**.** *CMD and binary CMD are different algorithms for BPSK symbols* $\mathbf{m} = [-1, 1]^T$*. We now provide a proof by contradiction.*

Proof. First, we rewrite (3.14) into

$$\left(\mathbf{G}^{(j+1)}\right)^T = \begin{bmatrix}\mathbf{G}_{1,*}^{(j+1)} \\ \mathbf{G}_{2,*}^{(j+1)}\end{bmatrix}^T \tag{A.13a}$$

$$= \begin{bmatrix}\mathbf{G}_{1,*}^{(j)} & \mathbf{G}_{2,*}^{(j)}\end{bmatrix} - \delta^{(j)} \cdot \left(\frac{\partial L(\mathbf{G})}{\partial \mathbf{G}}\right)^T \Bigg|_{\mathbf{G}=\mathbf{G}^{(j)}} \tag{A.13b}$$

$$= \begin{bmatrix}\mathbf{G}_{1,*}^{(j)} & \mathbf{G}_{2,*}^{(j)}\end{bmatrix} - \delta^{(j)} \cdot \mathbf{A}^T \tag{A.13c}$$

to note that the following two equalities should hold since $s_n = g_{2,n} - g_{1,n}$:

$$\begin{aligned}\mathbf{s}^{(j+1)} &= \mathbf{s}^{(j)} - \delta^{(j)} \cdot \frac{\partial L(\mathbf{s})}{\partial \mathbf{s}}\bigg|_{\mathbf{s}=\mathbf{s}^{(j)}} \\ &\stackrel{!}{=} \left[\mathbf{G}_{2,*}^{(j)} - \mathbf{G}_{1,*}^{(j)}\right] - \delta^{(j)} \cdot [\mathbf{A}_{2,*} - \mathbf{A}_{1,*}] \qquad \text{(A.14)} \\ \Rightarrow \frac{\partial L(\mathbf{s})}{\partial \mathbf{s}} &\stackrel{!}{=} \mathbf{A}_{2,*} - \mathbf{A}_{1,*}\,. \qquad \text{(A.15)}\end{aligned}$$

Both sides of (A.15) can be divided into two parts — the log-likelihood and the log-prior gradients:

$$\begin{aligned}\frac{\partial L(\mathbf{s})}{\partial \mathbf{s}} &= -\frac{\partial \ln p(\mathbf{y}|\mathbf{s})}{\partial \mathbf{s}} - \frac{\partial \ln p(\mathbf{s})}{\partial \mathbf{s}} \qquad \text{(A.16)} \\ \mathbf{A}_{2,*} - \mathbf{A}_{1,*} &= -\frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \mathbf{G}_{2,*}} - -\frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \mathbf{G}_{1,*}} \\ &\quad - \frac{\partial \ln p(\mathbf{G})}{\partial \mathbf{G}_{2,*}} - -\frac{\partial \ln p(\mathbf{G})}{\partial \mathbf{G}_{1,*}}\,. \qquad \text{(A.17)}\end{aligned}$$

We now prove if both sides differ. *First*, the log-prior gradients are different:

$$\begin{aligned}-\frac{\partial \ln p(\mathbf{s})}{\partial \mathbf{s}} &\stackrel{!}{=} -\frac{\partial \ln p(\mathbf{G})}{\partial \mathbf{G}_{2,*}} + \frac{\partial \ln p(\mathbf{G})}{\partial \mathbf{G}_{1,*}} \qquad \text{(A.18a)} \\ \tanh(\mathbf{s}/2) &\stackrel{!}{=} \left(1 - e^{-\mathbf{G}_{2,*}}\right) - \left(1 - e^{-\mathbf{G}_{1,*}}\right) \qquad \text{(A.18b)} \\ \frac{e^{-\mathbf{G}_{1,*}} - e^{-\mathbf{G}_{2,*}}}{e^{-\mathbf{G}_{1,*}} + e^{-\mathbf{G}_{2,*}}} &\neq e^{-\mathbf{G}_{1,*}} - e^{-\mathbf{G}_{2,*}}\,. \qquad \text{(A.18c)}\end{aligned}$$

Second, comparing both sides of the log-likelihood gradients

$$-\frac{\partial \ln p(\mathbf{y}|\mathbf{s})}{\partial \mathbf{s}} = -\frac{\partial \tilde{\mathbf{x}}(\mathbf{s})}{\partial \mathbf{s}} \cdot \frac{\partial \ln p\left(\mathbf{y}|\tilde{\mathbf{x}}(\mathbf{s})\right)}{\partial \tilde{\mathbf{x}}} \qquad \text{(A.19)}$$

and

$$\begin{aligned}-\frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \mathbf{G}_{2,*}} + \frac{\partial \ln p(\mathbf{y}|\mathbf{G})}{\partial \mathbf{G}_{1,*}} &= -\left[\begin{bmatrix}-1 & 1\end{bmatrix} \cdot \frac{\partial \tilde{x}_1(\mathbf{g}_1)}{\partial \mathbf{g}_1} \cdots \begin{bmatrix}-1 & 1\end{bmatrix} \cdot \frac{\partial \tilde{x}_{N_\mathrm{T}}(\mathbf{g}_{N_\mathrm{T}})}{\partial \mathbf{g}_{N_\mathrm{T}}}\right] \\ &\quad \cdot \operatorname{diag}\left\{\frac{\partial \ln p\left(\mathbf{y}|\tilde{\mathbf{x}}(\mathbf{G})\right)}{\partial \tilde{\mathbf{x}}}\right\}, \qquad \text{(A.20)}\end{aligned}$$

and noting that $\frac{\partial \ln p(\mathbf{y}|\tilde{\mathbf{x}}(\mathbf{G}))}{\partial \tilde{\mathbf{x}}} = \frac{\partial \ln p(\mathbf{y}|\tilde{\mathbf{x}}(\mathbf{s}))}{\partial \tilde{\mathbf{x}}}$, we only need to prove the equality:

$$\frac{\partial \tilde{x}_n(s_n)}{\partial s_n} \stackrel{!}{=} \frac{\partial \tilde{x}_n(g_{2,n})}{\partial g_{2,n}} - \frac{\partial \tilde{x}_n(g_{1,n})}{\partial g_{1,n}}\,. \qquad \text{(A.21)}$$

However, using (3.14), (A.2), and the sigmoid function $\rho(x) = 1/(1+e^{-x})$ (see Appendix C), the following results hold:

$$\frac{\partial \tilde{x}_n(\mathbf{g}_n)}{\partial \mathbf{g}_n} = \frac{1}{\tau^{(j)}} \cdot \left[\operatorname{diag}\left\{\sigma_\tau(\mathbf{g}_n)\right\} - \sigma_\tau(\mathbf{g}_n) \cdot \sigma_\tau(\mathbf{g}_n)^T\right] \cdot \mathbf{m} \tag{A.22}$$

$$= \frac{1}{\tau^{(j)}} \cdot \left[\operatorname{diag}\left\{\tilde{\mathbf{z}}\right\} - \tilde{\mathbf{z}} \cdot \tilde{\mathbf{z}}^T\right] \cdot \begin{bmatrix} -1 \\ 1 \end{bmatrix} \tag{A.23}$$

$$= \frac{2}{\tau^{(j)}} \cdot \begin{bmatrix} \tilde{z}_{1,n}^2 - \tilde{z}_{1,n} \\ -(\tilde{z}_{1,n}^2 - \tilde{z}_{1,n}) \end{bmatrix} \tag{A.24}$$

$$\Rightarrow \frac{\partial \tilde{x}_n(\mathbf{g}_{2,n})}{\partial \mathbf{g}_{2,n}} - \frac{\partial \tilde{x}_n(\mathbf{g}_{1,n})}{\partial \mathbf{g}_{1,n}} \tag{A.25}$$

$$= \frac{4}{\tau^{(j)}} \cdot \left(\tilde{z}_{1,n} - \tilde{z}_{1,n}^2\right) \tag{A.26}$$

$$= \frac{4}{\tau^{(j)}} \cdot \left(\rho\left(\frac{g_{2,n} - g_{1,n}}{\tau^{(j)}}\right) - \rho^2\left(\frac{g_{2,n} - g_{1,n}}{\tau^{(j)}}\right)\right) \tag{A.27}$$

$$= \frac{4}{\tau^{(j)}} \cdot \rho'\left(\frac{g_{2,n} - g_{1,n}}{\tau^{(j)}}\right) \tag{A.28}$$

$$= \frac{1}{\tau^{(j)}} \cdot \left[1 - \tanh^2\left(\frac{g_{2,n} - g_{1,n}}{2\tau^{(j)}}\right)\right] \tag{A.29}$$

$$\neq \frac{\partial \tilde{x}_n(s_n)}{\partial s_n} \tag{A.30}$$

$$= \frac{1}{2\tau^{(j)}} \cdot \left[1 - \tanh^2\left(\frac{s_n}{2\tau^{(j)}}\right)\right]. \tag{A.31}$$

After noting that the sums of log-likelihoods and log-priors in (A.16) and (A.17) also differ, i.e., (A.15) does not hold, this final contradiction completes our proof. □

As final remark, we note that (A.29) and (A.31) only differ by a constant factor of 2, and also both sides in (A.18c) only by a normalization with $\exp(-\mathbf{G}_{1,*}) - \exp(-\mathbf{G}_{2,*})$. The high similarity of CMDNet and binary CMDNet ($\text{CMDNet}_{\text{bin}}$) could explain why both algorithms are able to achieve a similar performance as shown in Fig. 3.7.

A.2.2 Local Scattering Massive MIMO Model

In order to assess the performance of the proposed symbol detection algorithms from Chapter 3 and Appendix A, we use the local scattering model

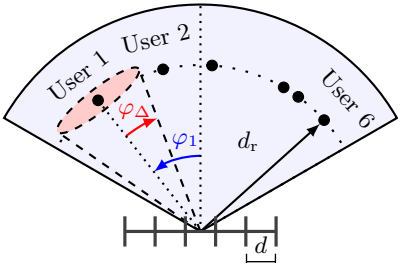

Figure A.3: Local scattering model for the far-field of massive systems. With uniformly distributed angular deviation $\epsilon \sim \mathcal{U}(-\varphi_\Delta, \varphi_\Delta)$, we arrive at the so-called One-Ring model.

in this work shown in Fig. A.3 as a first step towards a realistic massive MIMO model [BHS17]. In this model, we assume that a Base Station (BS) is equipped with a horizontal uniform linear antenna array with N_R receive antennas. The antenna spacing d is measured by

$$D = \frac{d}{\lambda} \in (0, 1/2] \tag{A.32}$$

in multiple of the wavelength λ at the carrier frequency and usually below $D \leq 1/2$ to achieve a good spatial resolution of the array. Further, the BS serves multiple User Equipments (UEs) usually located at fixed positions in the far-field of the BS as shown in Fig. A.3.

Since the BS is elevated, it has no scatterers in its near-field, and scattering is concentrated around the UEs. Considering an uplink without connection via Line-Of-Sight (LOS), the transmit signal of one UE $n \in \{1, \ldots, N_\mathrm{T}\}$ is diffracted and reflected from multiple nearby scatterers towards the BS array resulting in a high number N_path of multipaths. Each of these multipath components reaches the Uniform Linear Array (ULA) in a plane wave from a particular angle $\tilde{\varphi}_i$ with specific gain and phase-rotation ξ_i. Since the plane wave further arrives phase-shifted at each antenna, the array response is:

$$\mathbf{a}_i = \xi_i \cdot \begin{bmatrix} 1 & e^{2\pi j D \sin(\tilde{\varphi}_i)} & e^{2\pi j D (N_\mathrm{R}-1) \sin(\tilde{\varphi}_i)} \end{bmatrix}^T . \tag{A.33}$$

Now, each of the channel responses in the massive MIMO channel matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \cdots & \mathbf{h}_{N_\mathrm{T}} \end{bmatrix} \tag{A.34}$$

results from the superposition of multipath components:

$$\mathbf{h} = \sum_{i=1}^{N_{\text{path}}} \mathbf{a}_i \,. \tag{A.35}$$

Assuming $\tilde{\varphi}_i$ and ξ_i to be i.i.d. RVs with pdfs $p(\tilde{\varphi}_i) = p(\tilde{\varphi})$ and $p(\xi_i)$, respectively, and ξ_i to be zero-mean — a reasonable assumption due to random phase rotation — we can approximate the statistics of $\mathbf{h}$ by application of the multidimensional central limit theorem as the number of paths grows very large ($N_{\text{path}} \to \infty$) [BHS17]:

$$\mathbf{h} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{R}_{\mathbf{h}}) \quad \text{for} \quad N_{\text{path}} \to \infty \,. \tag{A.36}$$

To conclude the derivation of the local scattering model, calculation of the correlation matrix $\mathbf{R}_{\mathbf{h}}$ is required. By rewriting $\mathbf{R}_{\mathbf{h}} = \mathrm{E}[\sum_{i=1}^{N_{\text{path}}} \mathbf{a}_i \mathbf{a}_i^H]$, each entry in $\mathbf{R}_{\mathbf{h}}$ is [BHS17]:

$$[\mathbf{R}_{\mathbf{h}}]_{l,m} \tag{A.37a}$$

$$= \sum_{i=1}^{N_{\text{path}}} \mathrm{E}_{\xi_i \sim p(\xi_i)} \left[|\xi_i|^2 \right] \cdot \mathrm{E}_{\tilde{\varphi}_i \sim p(\tilde{\varphi})} \left[e^{2\pi j D(l-1)\sin(\tilde{\varphi}_i)} e^{-2\pi j D(m-1)\sin(\tilde{\varphi}_i)} \right] \tag{A.37b}$$

$$= \mathrm{E}_{\tilde{\varphi} \sim p(\tilde{\varphi})} \left[e^{2\pi j D(l-m)\sin(\tilde{\varphi})} \right] \cdot \underbrace{\sum_{i=1}^{N_{\text{path}}} \mathrm{E}_{\xi_i \sim p(\xi_i)} \left[|\xi_i|^2 \right]}_{=\zeta} \tag{A.37c}$$

$$= \underbrace{\zeta}_{=1} \cdot \int_{-\infty}^{\infty} e^{2\pi j D(l-m)\sin(\tilde{\varphi})} p(\tilde{\varphi}) \, \mathrm{d}\tilde{\varphi} \,. \tag{A.37d}$$

If we assume perfect power allocation or equivalently that each UE has the same distance d_{r} to the BS, we can set the path gain to $\zeta = 1$. Since $[\mathbf{R}_{\mathbf{h}}]_{l,m}$ only depends on the difference or distance between antennas $l - m$, $\mathbf{R}_{\mathbf{h}}$ is further a Toeplitz matrix. Its diagonal entries ($l = m$) are $\zeta = 1$. The integral in (A.37d) can be evaluated numerically for any angular distribution $p(\tilde{\varphi})$ of multipath components.

To finally arrive at the local scattering model, we assume these components to originate from one scattering cluster around the UE as shown in Fig. A.3 [BHS17]. Then, the angle $\tilde{\varphi}$ can be decomposed into the deterministic nominal angle between UE and ULA φ and a random deviation ϵ by $\tilde{\varphi} = \varphi + \epsilon$.

Table A.1: Typical parameters of the local scattering model.

Environment	σ_φ	φ_Δ
urban	$10°$	$17.32° \approx 20°$
flat rural	$< 10°$	
hilly	$> 10°$	

We note that it is possible to assume different deviation distributions $p(\epsilon)$ like, e.g., a Gaussian or Laplace distribution. In our work, we use uniformly distributed $\epsilon \sim \mathcal{U}(-\varphi_\Delta, \varphi_\Delta)$ with angular spread $\varphi_\Delta = \sqrt{3} \cdot \sigma_\varphi$ and angular standard deviation σ_φ. Thus, we assume all scatterers to lie on a circle centered at the UE (see Fig. A.3). For this reason, this model is called One-Ring model. It leads to high spatial correlation [BHS17] making it a worst-case scenario. By inserting the uniform distribution into (A.37d), we finally arrive at

$$[\mathbf{R}_\mathbf{h}]_{l,m} = \frac{1}{2\varphi_\Delta} \int_{\varphi-\varphi_\Delta}^{\varphi+\varphi_\Delta} e^{2\pi j D(l-m)\sin(\tilde{\varphi})} \, \mathrm{d}\tilde{\varphi} \, . \tag{A.38}$$

Typical local scattering model parameters of $p(\epsilon)$ are shown in Tab. A.1 [BHS17].

In our simulations of this work, we assume a value of $\varphi_\Delta = 20°$ to resemble an urban cellular network scenario. Furthermore, we assume each ULA to cover one specific area or direction within the cellular network. Inside this cell sector of size φ_cell, the ULA serves multiple UEs and each UE n is located at a random uniformly distributed angle $\varphi_n \sim \mathcal{U}(-\varphi_\text{cell}/2, \varphi_\text{cell}/2)$. Assuming the BS to be equipped with 3 ULAs, we choose a $\varphi_\text{cell} = 120°$ cell sector to cover the whole $360°$ radius for massive MIMO system simulations. Moreover, we sample N_T angles uniformly from integer values without replacement such that $\varphi = i$ with $i \in \mathbb{Z}$ between $i \in [-\varphi_\text{cell}/2, -\varphi_\text{cell}/2[$ and $\varphi_n \neq \varphi_{n+1}$. After computation of $\mathbf{R}_{\mathbf{h},n}$ according to (A.38), we sample the respective columns n of the channel matrix $\mathbf{H}$ as:

$$\mathbf{h}_n = \mathbf{R}_{\mathbf{h},n}^{1/2} \cdot \mathbf{n} \quad \text{with} \quad \mathbf{n} \sim \mathcal{N}_\mathbb{C}\left(\mathbf{0}, \mathbf{I}_{N_\mathrm{T}}\right) . \tag{A.39}$$

As a final remark, we note that three key effects of today's massive MIMO systems are not captured by the local scattering model: scattering in the near-field of the BS, multiple scattering clusters around the UEs and shadowing over the array [BHS17].

A.3 Deeper Insights into Training of CMDNet

Unfortunately, not all results of our extensive investigations on CMDNet found their way into publications. In this section, we provide these additional results. In particular, we want to shed light on training aspects which have fallen short in [BBD21] and are only mentioned as part of the training hyperparameter setup. These aspects are of major importance since the number of training hyperparameters is quite large and makes it thus difficult to find a combination leading to promising accuracy. For example, these aspects include the aforementioned starting point initialization but also the selection of the training criterion / optimization loss, the visualization of the training progress, the selection of the optimization algorithm and of the architecture, i.e., the number of layers. Further, we present a first investigation of CMDNet's online training capability and point out the effects of various training mismatches.

A.3.1 Soft Information Measure

In the following sections, we draw on results of soft information, i.e., cross-entropy, besides BER accuracy to gain deeper insights since usually subsequent soft decoding is assumed in communications. Therefore, we now detail our empirical computation of the cross-entropy $\mathcal{H}(p,q)$: To avoid numerical computation of $\ln(0)$ leading to NaN outputs, the inputs $q(\mathbf{x}_i|\mathbf{y}_i,\boldsymbol{\theta}) < 10^{-7}$ of the cross-entropy are clipped to 10^{-7}. In this chapter, the cross-entropy is calculated empirically and normalized by N_T to enable comparison of MIMO systems with different dimensions:

$$\hat{\mathcal{H}}(p,q) = -\frac{1}{NN_\mathrm{T}} \sum_{i=1}^{N} \ln q(\mathbf{x}_i|\tilde{\mathbf{y}}_i,\boldsymbol{\theta}) \approx \frac{\mathcal{H}(p,q)}{N_\mathrm{T}} \,. \tag{A.40}$$

However, numerical optimization according to (A.40) without modification may be numerical instable. A combination of softmax and loss into one layer simplifies computation of gradients and improves numerical stability. Surprisingly, we observed faster convergence without this combination at optimization run time. Throughout this thesis, in CMDNet experiments, we consequently used the Keras backend implementation of the categorical cross-entropy function with `from_logits = False` from [AAB+15] for training being equal to our own computation for validation.

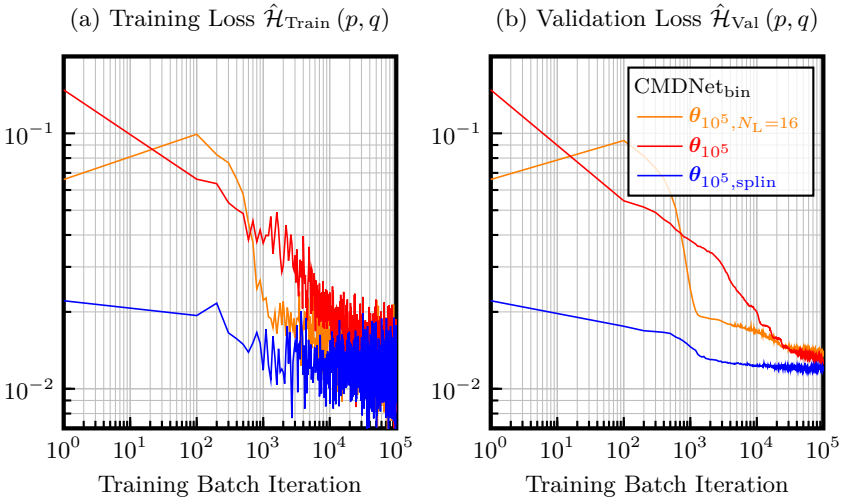

Figure A.4: Empirical cross-entropy loss $\hat{\mathcal{H}}(p,q)$ with (a) training and (b) validation data as a function of training batch iteration of CMDNet, trained with different parametrization in a 32×32 MIMO system.

A.3.2 Training Progress

First, we deal with an essential part of DNN training: observation and hence visualization of the training progress. Tracking the value of the objective function (3.23) equivalent to the loss is key to understand the training progress. This makes it possible to recognize if and how fast training converges, i.e., how many training iterations N_{e} are required. It is computed in every training iteration besides the gradient updates of the parameters by Stochastic Gradient Descent (SGD) and backpropagation and a measure of the quality of the soft information.

In Fig. A.4 (a), we show the training loss of CMDNet, i.e., cross-entropy $\hat{\mathcal{H}}_{\text{Train}}(p,q)$, computed with the current training data batch for different training initialization in a 32×32 MIMO system as a function of training iteration. In the scenario of MIMO detection, the x-axis represents batch iterations rather than epochs, which are the default measure used in the ML domain. This is because *the classic definition of an epoch — one full pass through the entire training dataset — cannot be applied when new data is generated in each batch iteration according to a well-defined model, e.g., a MIMO system model.*

The loss was tracked with a resolution of 100 training iterations to save

memory, and computation time in case of validation data. It starts at a high value depending on the starting point initialization. First, it decreases fast and then converges slowly. In the area of convergence (after ≈ 1000 iterations), the curve becomes very noisy with a small training data batch size $N_\mathrm{b} = 500$, making it difficult to evaluate whether training has stopped or progresses minimally.

Owing to the noisy training loss, we decided to introduce a constant validation dataset (see Chapter 2) with large batch size $N_\mathrm{b} = 10,000$. The reason different from those of the ML domain makes the use of validation data an innovation. Usually, the limited amount of data typical for problems of the ML community necessitates validation of the progress of training or the selection of the model with a separated dataset to avoid overfitting. However, in this case, we are able to compute infinite data according to our model and the training loss resembles the training progress closely but noisy.

As a result of our introduction, the validation loss, i.e., cross-entropy $\hat{\mathcal{H}}_\mathrm{Val}(p, q)$, shown in Fig. A.4 (b), depicts the training progress much more clearly. Depending on the architecture and the parameter starting point, the progress behaves different: For example, a starting point $\boldsymbol{\theta}_{0,\mathrm{splin}}$ with linear decreasing parameters $\tau^{(j)}$ and $\delta^{(j)}$ from (3.36) leads to a low loss in the beginning and ending over the whole training SNR region in contrast to training with default starting point $\boldsymbol{\theta}_0$ from (3.35). Note that the BER performance in the high SNR region is worse than with default parametrization being the reason for choosing the default one. The loss may even increase after the first iterations as being the case with $N_\mathrm{L} = 16$ layers. Even after $10,000$ iterations, we still observe noise. We conjecture that Adam has found a local (or global) minimum and oscillates around it, making only very little progress since the step size may be too large. The training progress of DNNs in general has been investigated from an information theoretic view in [TZ15; ST17] and was also found to have two SGD phases with high and low gradient means and variances, vice versa. These both phases may lead to fitting and compression in DNNs, respectively. Further, we observe that the validation loss of CMDNet with $N_\mathrm{L} = 64$ layers falls below of that with $N_\mathrm{L} = 16$ beyond 20000 iterations. The increased accuracy of CMDNet with $N_\mathrm{L} = 64$ seems to originate from training in this region.

At this point, we note that comparison of different validation losses is only possible up to a certain accuracy determined by the size of the validation dataset. In general, visualization of the validation loss serves as a quick tool to assess the quality of the selected training hyperparameters. As a final remark, we notice that the larger the validation dataset is, the more precise the loss computation. But also training time increases since the computation of the validation loss requires inference through CMDNet, which, although

low in complexity, still adds overhead compared to the SGD training itself. Besides validation, a third evaluation — resembling the test set in the ML domain (see Chapter 2) and performed in this thesis — is necessary in the end to reliably determine BER accuracy and other metrics with the required precision.

A.3.3 Optimization Criterion and Training Loss

As noted in Sec. 3.4.2 and [BBD21], we combine Cross-Entropy (CE) and CMDNet with probability output, i.e., softmax layer, since this choice is grounded in information theory and enables optimization of soft outputs approximating those of the IO detector. In contrast, recent literature [SDW19; GAH20; KAHF20] considers mainly the MSE as the loss function. This implies an estimation, not the actual detection problem, and means that a Gaussian variational posterior distribution is assumed — as shown in Chapter 2 in (2.62) — with the mean equivalent to the symbol estimate $\hat{\mathbf{x}}$, i.e., $\tilde{\mathbf{x}}(\mathbf{G}^{(N_{\mathrm{it}})})$ from (3.15) for CMDNet. Likewise, the RV $\mathbf{x}$ is relaxed into the entire complex domain $\mathbb{C}^{N_{\mathrm{T}}\times 1}$, and not limited into $[\min(\mathcal{M}), \max(\mathcal{M})]^{N_{\mathrm{T}}\times 1}$ as in the case of CMDNet. After noting the independence of the variance from the optimization parameters $\boldsymbol{\theta}$, the empirical loss function w.r.t. the symbol estimator output of CMDNet reads:

$$\hat{\mathcal{H}}_{\mathrm{MSE}}\left(p,q\right) = \frac{1}{NN_{\mathrm{T}}} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \tilde{\mathbf{x}}\left(\mathbf{G}^{(N_{\mathrm{it}})} | \tilde{\mathbf{y}}_i, \boldsymbol{\theta}\right) \right\|_2^2 \approx \frac{\mathcal{H}\left(p,q\right)}{N_{\mathrm{T}}} \,. \tag{A.41}$$

Although it may seem like a crude assumption, it leads not only to promising results for detectors from [SDW19; GAH20; KAHF20] but also if applied to CMDNet. Surprisingly, the detection accuracy of CMDNet with MSE loss and symbol estimation output shown in Fig. A.5 (a) increases beyond default CMDNet (CE loss) and becomes even similar to that of OAMPNet, i.e., the best considered suboptimal detector in a 32×32 MIMO system. Examining a measure of soft information, i.e., Cross-Entropy (CE), we are able to understand this both surprising and impressive result: We note that the cumulative integral, i.e., the area below the curve, is larger between 4 and 6 dB within the SNR training interval of $E_{\mathrm{b}}/N_0 \in [4, 27]$ dB when using the MSE criterion compared to cross-entropy. Since the cross-entropy loss in this SNR region is significantly larger than outside it, it dominates the total cross-entropy. Thus, default CMDNet still minimizes cross-entropy over the whole training interval. However, this implies that optimization with MSE loss places greater emphasis on errors at higher SNR values compared to the cross-entropy. The same observation holds for OAMPNet being trained with

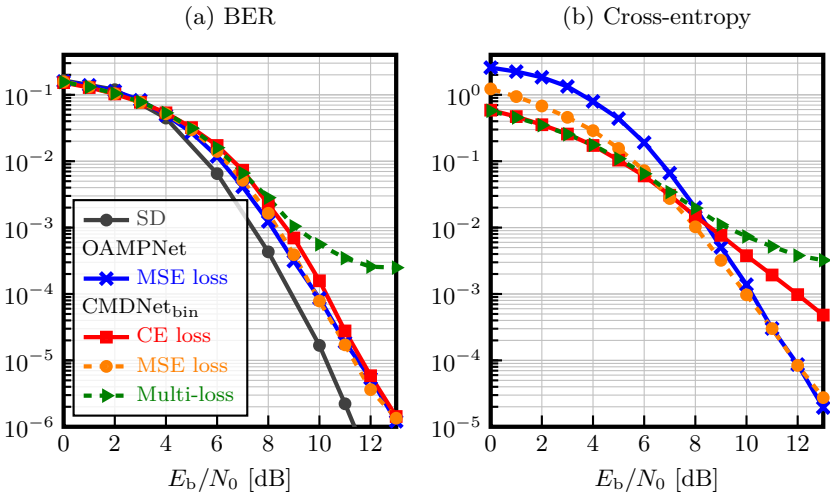

Figure A.5: BER curves and cross-entropy of CMDNet trained according to different optimization criteria in a 32×32 MIMO system.

MSE loss as well. *In fact, CMDNet is able to perform as well as OAMPNet when trained with the same loss.*

Furthermore, inspired by the notion of auxiliary classifiers from GoogLeNet, the authors of [SDW19; KAHF20] use a weighted sum of the MSE losses w.r.t. the outputs of each layer j as the loss function. This so-called multi-loss addresses multiple challenges when training DNNs such as vanishing gradients or initialization sensitivity. Hence, we also experimented with multi-loss and applied it to default CMDNet over multiple layers:

$$\begin{aligned}\hat{\mathcal{H}}_{\text{multi-loss}}\left(p,q\right) &= -\frac{1}{NN_{\text{T}}}\sum_{i=1}^{N}\sum_{j=1}^{N_{\text{it}}} j\cdot\ln q^{(j)}\left(\mathbf{x}_i^{(j)}|\tilde{\mathbf{y}}_i,\boldsymbol{\theta}\right)\\ &\approx \sum_{j=1}^{N_{\text{it}}}\frac{\mathcal{H}\left(p,q^{(j)}\right)}{N_{\text{T}}}\end{aligned}$$

$$\text{with}\quad \ln q^{(j)}\left(\mathbf{x}^{(j)}|\tilde{\mathbf{y}},\boldsymbol{\theta}\right) = \begin{bmatrix} x_n = m_1 \\ \vdots \\ x_n = m_M \end{bmatrix}^T \cdot \ln\left(\sigma_{\tau^{(j)}}\left(\mathbf{g}_n^{(j)}\right)\right). \qquad \text{(A.42)}$$

From Fig. A.5, we observe that the accuracy deteriorates significantly in the

high SNR region and that the multi-loss does not provide any benefit. Using a multi-MSE-loss, we observe the same performance. The reason may lie in constraining of the hidden layer outputs $\sigma_{\tau^{(j)}}(\mathbf{g}_n^{(j)})$. Further, we see that selection of a model-based unfolding approach may alleviate the need for further regularization or modification compared to DNNs, and enables fast training convergence and high detection accuracy.

A.3.4 Optimization Algorithm

Besides selection of a loss function including regularization, the choice of an optimization algorithm is another hyperparameter crucial for DNN training. Usually, first order gradient descent methods, i.e., variants of SGD are used since, e.g., second order Newton methods require analytical or numerical computation of the second derivative being computationally demanding or intractable. Basically, as outlined in Chapter 2, SGD is equivalent to gradient descent steps with not all, but with a random subset of the training data also known as a batch with size N_b. Using a subset, allows adjusting training to computing resources. Further, the variance of the gradient steps becomes larger or noisy with decreasing batch size allowing to leave a local minimum and to find one inside a flat objective landscape which is known to improve generalization [WRS+17].

In related works [SDW19; GAH20; KAHF20] and our work [BBD21], Adam is used — a variant of SGD. The reason for this choice may be its popularity: It is common in recent ML literature [WRS+17] since it is easy to use, i.e., requires little hyperparameter tuning, and provides fast convergence speed. One major drawback is that Adam tends to find drastically different solutions compared to SGD and sharp minima known to generalize poorly to unseen data points [WRS+17]. *However, in contrast to typical data-based ML problems, we are able to neglect or avoid these generalization problems in MIMO detection*: Since we have a model, we are able to generate an amount of training data large enough to fulfill (3.33) by (3.34) approximately. *This makes application of Adam a valid option.*

Motivated by the findings from [WRS+17] that SGD finds different solutions from Adam, we experimented and also optimized CMDNet by SGD. In Fig. A.6, we compare the BER results of CMDNet optimized by SGD and Adam for different batch sizes N_b, both with standard parametrization according to [AAB+15]. This means we applied basic SGD with learning rate 10^{-2} without momentum. The number of training iterations $N_\mathrm{e} = \{10^5, 1.5 \cdot 10^6, 4 \cdot 10^4\}$ is chosen to allow for convergence with each batch size $N_\mathrm{b} = \{500, 32, 5000\}$.

Considering Adam, we observe that especially a smaller batch size ($N_\mathrm{b} =$

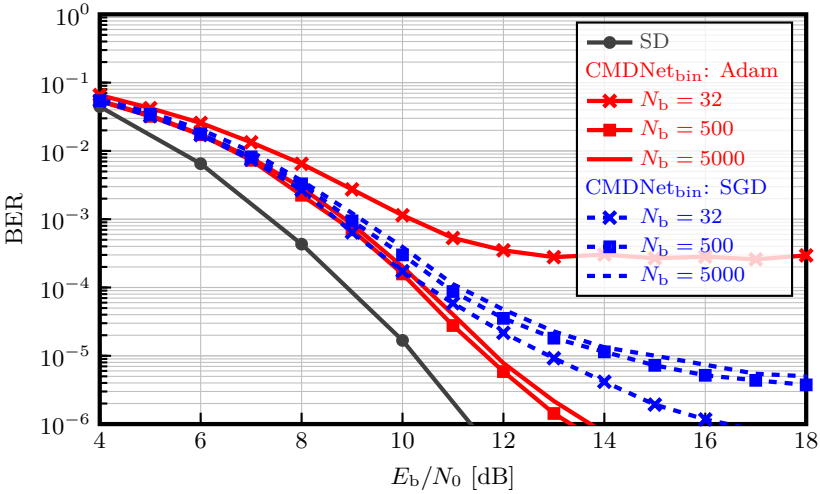

Figure A.6: BER curves of CMDNet trained with different batch sizes N_b and optimizers Adam and SGD in a 32×32-MIMO system.

32) than default ($N_b = 500$) leads to degradation in accuracy. The reason may lie in the poor generalization of Adam already mentioned necessitating larger batches. In contrast, choosing a larger batch size $N_b = 5000$ results in negligible performance loss. It remains the question if fine-tuning around the default value of $N_b = 500$ may lead to higher accuracy. At this point, we refer to the huge number of training hyperparameters making it nearly impossible to find the correct parametrization in a reasonable amount of time.

When optimizing CMDNet with default initialization $\boldsymbol{\theta}_0$ by SGD, we noticed that training does not converge. Hence, we applied starting point $\boldsymbol{\theta}_{0,\text{splin}}$ from (3.36). Using SGD, rather a low training batch size $N_b = 32$ commonly used in the ML domain leads to satisfying results. However, the BER is worse over the whole SNR region compared to that of CMDNet with both $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_{0,\text{splin}}$ trained with Adam. We conjecture that the pre-settings of Adam make it easy to use and leads to good convergence if the generalization issue is negligible or taken into account. In contrast, SGD may simply require more hyperparameter fine-tuning to achieve the same results.

We note that using Nesterov's accelerated gradient method speeds up training and improves convergence by incorporating momentum, which helps the optimizer anticipate the direction of the gradient, leading to faster

updates and reducing oscillations [SLA+19]. Numerical experiments (with momentum of 0.9) suggest that larger batch sizes are required or possible with Nesterov momentum similar to Adam with built-in momentum. This result is in accordance with observations from literature [SLA+19]. In our case, training with $N_\mathrm{b} = 500$ leads to similar BER compared to basic SGD with $N_\mathrm{b} = 32$, whereas training with $N_\mathrm{b} = 32$ does not converge. Furthermore, we note that we have not used a schedule with decreasing learning rate in this thesis which may be beneficial for both optimization with Adam and SGD as noted in Chapter 2.

Finally, we notice that it is difficult to draw specific conclusions with such a large number of hyperparameters and that finding a good parametrization seems like pure game of chance. It necessitates spending a lot of time on validation since general exact guidelines do not exist (see, e.g., [WRS+17; SLA+19]). One promising idea to overcome this problem could be to exploit random hyperparameter search known to be more efficient than grid search [BB12].

A.3.5 Model Architecture: Number of Layers

Similar to optimization, also selection of a DNN model or architecture requires a trial and error validation procedure. When it comes to CMDNet, there is fortunately only one hyperparameter besides parameters initialization to tune the model: the number of iterations or layers $N_\mathrm{it} = N_\mathrm{L}$. In Fig. A.7, we present results of CMDNet with different numbers of layers N_L in our default example of a 32×32 MIMO system expanding on results from Sec. 3.5.3 and Fig. 3.6. As a benchmark and for comparison, it is shown how the performance of the AMP algorithm changes with the number of iterations. We observe that the BER of CMDNet decreases with an increase in layer number as expected. Further, we note significant accuracy gains from $N_\mathrm{L} = 4$ to 16. Above $N_\mathrm{L} = 16$, the accuracy gains are minor or ambiguous. For example, CMDNet with $N_\mathrm{L} = 128$ performs worse than with default $N_\mathrm{L} = 64$. Maybe, the unadapted number of training iterations $N_\mathrm{e} = 100000$ is too low and does not allow for convergence. However, we note again that this is speculative, and that other layer numbers may require other hyperparameter settings besides N_e. The almost maximum performance with just $N_\mathrm{L} = 16$ layers compared to $N_\mathrm{L} = \{32, 64, 128\}$ points into this direction.

A.3.6 Online vs. Offline Training

In the review process of the article [BBD21], a misconception of our approach became apparent. In [BBD21], we focus explicitly on offline learning, making

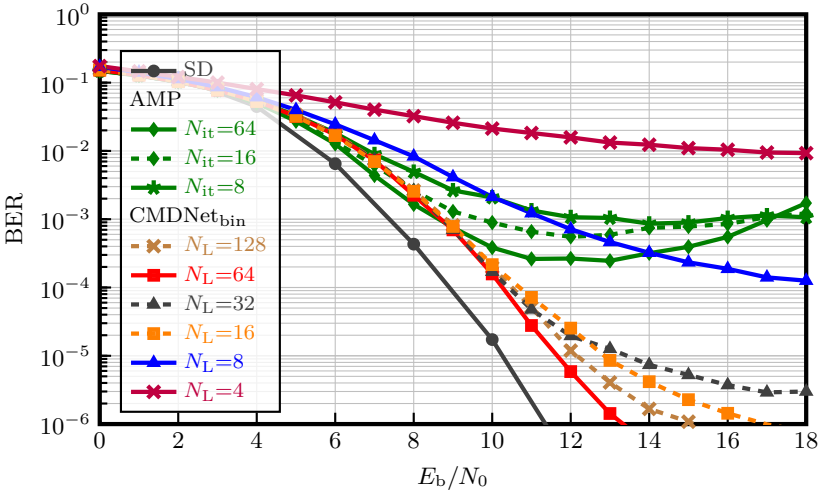

Figure A.7: BER curves for AMP and CMDNet with a varying number of iterations N_{it} or layer N_L in a 32×32 MIMO system.

online training aspects such as dataset size and number of training iterations less critical. The reasons for following our strategy are:

1. **Amortized Inference.** Offline learning means that we amortized our unfolded CMD algorithm CMDNet across multiple realizations of our training data to which $\mathbf{y}$, $\mathbf{H}$ and σ_n^2 belong to (see Chapter 2). Hence, we optimized the parameters $\boldsymbol{\theta}$ of CMDNet only once for the whole statistics of these RVs and used it accordingly for those statistics.

   In theory, this concept is known as Amortized Inference (see Chapter 2) and allows lowering complexity at the potential cost of accuracy since our DNN-like structure is not adapted to every realization anymore. In fact, at least amortization across $\mathbf{y}$ is usually assumed in most publications that deal with DNNs.

   In practice, this means that numerical optimization (training) w.r.t. all input statistics requires approximation of the expected value in equation (3.24) and (3.33) by an empirical sum (average) in (3.26) and (3.34), respectively. To make this approximation tight, we used a huge number of batches of training data, in this case $N = N_e \cdot N_b = 10^5 \cdot 500$. From a supervised learning perspective, allowing for tight approximation with a huge amount of training data is equivalent to enabling generalization to unseen data points, see Chapter 2.

To summarize the pros and cons of offline learning, we lower complexity at the cost of accuracy since we avoid the potentially wasteful online learning procedure. Since we only train once with great effort, we are even able to compensate a bit of the accuracy loss. After training, we only use CMDNet with its parameters $\boldsymbol{\theta}$ optimized for the whole input statistics for inference at run time.

2. **Starting point initialization.** To evaluate the effectiveness of the CMDNet compared to other approaches, there is another reason, besides Reason 1, why investigating the training progress/convergence is not very conclusive. It is the so-called starting point initialization (see Chapter 2) relevant in ML practice: When using variants of SGD for numerical optimization of, e.g., DNNs, we need to choose a starting point for parameters $\boldsymbol{\theta}$. This weight initialization in ML is not trivial. Usually, for DNNs, the weights are assumed to follow a certain probability distribution, e.g., Glorot Uniform, and randomly selected according to it. To investigate the relationship between accuracy and the number of training iterations, a single exemplary training run would be insufficient to capture the full range of statistical deviations arising from different initial starting points. This means that curves showing the accuracy vs. the number of training iterations should always be averaged w.r.t. these starting weight distributions.

   Since in our case we use a model-based approach instead of DNNs, we cannot rely on these default DNN weight distributions. Further, the choice of starting points heavily impacts training stability and convergence, and is hence crucial. In [BBD21], we thus sketched reasonable heuristics for a choice of the step size δ and the softmax temperature τ, both summarized in $\boldsymbol{\theta}$. These choices itself without any training allow for high detection accuracy which is not necessarily true for other initializations as shown in Fig. 3.7. Hence, we used starting points computed according to (3.35) and (3.36).

   Now, using always the same starting point implies training would proceed similarly and tend towards the same optimum from run to run in contrast to usual DNN training. We could observe this behavior in our simulations making the results reproducible [Bec23]. This means that the accuracy after the training process highly depends on the chosen starting point. In the extreme case, we could directly choose parameters being an optimum and no training would be required. Training progress could illustrate in a flat (noisy) curve not revealing further insights about CMDNet. Thus, we avoided illustration attempts of the training progress in [BBD21]. Interestingly, such progress is

Table A.2: DNN architecture for online MIMO detection.

Component	Layer	Dimension
Input	Received signal	N_{Rx}
ResNetBlock 1	Rectified Linear Unit (ReLU)	$N_{\mathrm{h}} = 512$
	Residual Connection	$N_{\mathrm{h}} = 512$
ResNetBlock 2	ReLU	$N_{\mathrm{h}} = 512$
	Residual Connection	$N_{\mathrm{h}} = 512$
	$\vdots$	$N_{\mathrm{h}} = 512$
ResNetBlock N_{L}	$\vdots$	$N_{\mathrm{h}} = 512$
$N_{\mathrm{T}}\times$ Classifier	Softmax	$M = 2$

already contained in rudiments in [BBD21] in Fig. 3.7: There, the BER performance before and after training is shown. Note that, in Appendix A.3.2, in contrast, we deal with the practical value of the training progress in depth, i.e., we show that visualization of the validation loss serves as a quick tool to assess whether training converges for the selected hyperparameters.

Following these two reasons, we opted against adding the online training separately as a new topic into [BBD21], and elaborate more deeply on the differences of offline training compared to online training.

First Online Learning Analysis

In the aftermath of the publication, however, we make a simple analysis of online learning or training to illuminate this blind spot of [BBD21]. We assume both $\mathrm{CMDNet}_{\mathrm{bin}}$ with $N_{\mathrm{L}} = 64$ and a specific DNN to be trained for a channel matrix realization $\mathbf{H}$ and then to be used within the coherence time. Afterwards, the current state of parameters $\boldsymbol{\theta}$ is used as a starting point for training according to the next realization $\mathbf{H}$. The SNR level is also fixed with σ_{n}^2 leaving only an amortization in (3.27) across received signals $\mathbf{y}$. As a starting point of CMDNet's parameters, we exploit the optimized parameters $\boldsymbol{\theta}_{10^5}$ from Fig. 3.4 and Fig. 3.7, and trained for $N_{\mathrm{e}} = 10^2$ iterations per realization with batch size $N_{\mathrm{b}} = 500$.

For design of the specific online-learning DNN, we use a simple Residual Network (ResNet) DNN detector [HZRS16b] shown in Tab. A.2 with N_{L} He-uniform-initialized ReLU layers of width N_{h} with a final softmax layer

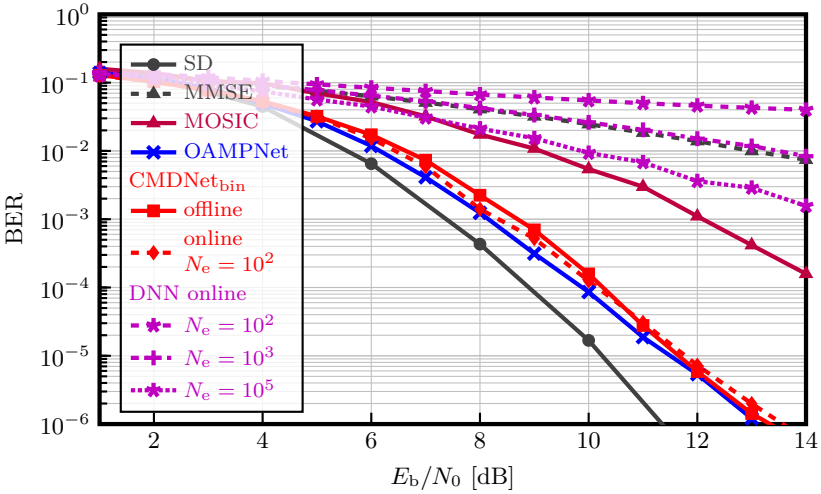

Figure A.8: BER curves of online-trained CMDNet and ResNet DNN in a 32×32 MIMO system with QPSK modulation. Effective system dimension is 64×64.

of width $M = 2$ for each of N_T transmit signals for classification. A residual connection between each current and previous layer improves the expressive power. If the input and layer dimensions do not match, we introduce a linear layer between input and first residual connection of width N_h. We choose width $N_h = 512$ and depth N_L such that performance is maximum for the respective number of training iterations N_e, while limiting DNN size to speed up training time. For $N_e = 10^2$ and 10^3, we thus select $N_L = 2$. For $N_e = 10^5$, we have $N_L = 6$. The number of $\mathbf{H}$ training realizations is 10^3, 10^2, and 10, respectively.

The results of the considered online training scenario are shown in Fig. A.8. When it comes to CMDNet, online training results in only slightly better average BER performance with this configuration. The ResNet DNN detector is able to match Minimum Mean Square Error (MMSE) detector performance after $N_e = 10^3$ iterations. However, the gap between SD and DNN BER curves remains significant even with online training for high values of $N_e = 10^5$. In this case, training on an Nvidia Titan RTX takes 700 s per realization, also significantly higher by a factor of 70 than the 10 s of CMDNet. This hints towards that it is a considerably greater effort to train a small DNN to high accuracy compared to a deep CMDNet.

In conclusion, the offline- and model-based design of ML algorithms like

CMDNet seems to be most promising, highlighting our contribution and justifying our restriction to offline learning in [BBD21]. However, a deeper scientific analysis of online training would require further investigations with novel scope. For example, evaluating CMDNet online requires reasonable assumptions for the online scenario like a statistical model for the change of the channel similar to [KAHF20]. Further, an evaluation adds only little value without comparing it to other light-weight approaches like MMNet. Therefore, we think that it remains to further research to find out about CMDNet's true online learning capability.

A.3.7 Robustness Against Various Mismatches

In the review process, the question arose whether CMDNet is robust against mismatches. *Mismatch — which can also be considered as overfitting (see Chapter 2) — can only occur if CMDNet is trained for specific realizations or statistics different from those encountered in use.* Since we follow the offline learning strategy (3.27) with amortization, we can choose a broad range of, e.g., channel, realizations for training that reflect all deployment statistics and thus partly avoid mismatch.

Now, we explain in detail what this means for different kinds of mismatches:

1. **SNR mismatch:** First, we point to Sec. 3.5.1 where we state that we train our algorithm CMDNet offline uniformly over a broad SNR range of $E_\mathrm{b}/N_0 \in [4, 27]$ dB. This means that we avoid mismatch by design using offline learning also known as amortized inference. Since we use a huge interval being much larger than the actual working point SNR region, mismatch outside of it is not really of relevance in practice. In Fig. 3.4, mismatch is indeed already illustrated: We see that CMDNet also performs well below 4 dB outside the training interval.

2. **Modulation mismatch:** From (3.14) and (3.32), we note that the modulation defines the architecture of CMDNet by construction of $\mathbf{m}$ or output $q(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$, respectively. Hence, it is not possible to analyze mismatch w.r.t. modulation since the number of classes/the architecture changes. Indeed, we could use parameters $\boldsymbol{\theta}$ optimized for, e.g., BPSK, and apply those to the architecture for, e.g., 16-Quadrature Amplitude Modulation (QAM), as long as the number of layers N_L is the same. For this example, we observe a slightly worse accuracy compared to DetNet in simulations, and it indeed works to some extent. In general, it makes more sense to optimize a low number of parameters to an intended scenario and to use it there.

3. **MIMO configuration mismatch:** The same reasoning as for modulation mismatch also applies to investigation of MIMO configuration mismatch. This means the dimensions of the MIMO system change and thus the architecture (and maybe also the number of layers N_{L} or parameters $\boldsymbol{\theta}$) of the algorithm.

   - If we only extract parameters of a 32×32 system and apply them to a 8×8 system, both with $N_{\mathrm{L}} = 64$, we see slightly improved accuracy w.r.t. original $\boldsymbol{\theta}$ with $N_{\mathrm{L}} = 16$ from Fig. 3.5. We explain this behavior by the larger layer and parameter number: CMDNet with $N_{\mathrm{L}} = 64$ is more expressive. Although the layer number is different and does not allow for drawing a precise conclusion, it seems we are able to apply $\boldsymbol{\theta}$ evaluated for one system to another with different dimensions.
   - In contrast to modulation mismatch, we could optimize our algorithm for a 32×32 system and apply the exactly same architecture to a 8×8 system by setting all not needed entries in $\mathbf{y}$, $\mathbf{H}$ and $\mathbf{x}$ to 0. But the implicit presumption in algorithm design that every user is active would be violated.
   - At this point, multi-user detection approaches where also the number of active users is determined suggests itself, and we enter the new scope of compressive sensing theory. As one interesting idea for future research to introduce multi-user detection, we could set one class to 0 (which means inactive) and optimize CMDNet for one or different prior probabilities of being inactive.
   - The idea raises new questions going beyond the scope of this thesis: For now, we assumed equal prior symbol probabilities $\boldsymbol{\alpha}$ and optimized for these statistics of $\mathbf{x}$. What happens if the $\boldsymbol{\alpha}$ are not equal but differ according to certain statistics? Is then amortization of CMDNet's parameters $\boldsymbol{\theta}$ across the statistics of $\boldsymbol{\alpha}$ a valid strategy? Or does it make more sense to determine one different parameter set for each $\boldsymbol{\alpha}$ online or offline? Furthermore, does it make sense to add the $\boldsymbol{\alpha}$ (maybe untied across multiple layers) to the set of trainable parameters for performance improvements after optimization? From a conceptual point of view, there is no reason to us why weighting of symbols different from the true $\boldsymbol{\alpha}$ should be beneficial. Further, first experiments showed that the latter idea does not come with any performance benefits and makes training even instable.

4. **Channel correlation mismatch:** In contrast to the latter two scenarios, the architecture does not change but only the channel input

statistics. For example, we could hence simply use CMDNet with parameters $\boldsymbol{\theta}$ trained for Gaussian i.i.d. channels $p(\mathbf{H})$ and apply it to the One-Ring model from Sec. 3.5.5 described in Appendix A.2.2. After simulation of this scenario, we observe better detection accuracy with these mismatched $\boldsymbol{\theta}_{\text{mismatch}}$ over the whole SNR region from Fig. A.9. At first glance, this comes to us as a big surprise since training according to the correct channel model should enable tight adaptation. Taking a closer look, we are able to explain this behavior:

- First, we note that we use Gaussian statistics in both the i.i.d. and the One-Ring model. The difference lies in adding correlation according to the One-Ring model between the columns in $\mathbf{H}$. In fact, this mostly changes condition number of the matrices $\mathbf{H}$ and seems only to require little retraining of parameters.
- Second, we note that we optimize soft outputs w.r.t. cross-entropy also shown in Fig. A.9. Although their quality correlates with detection accuracy, these could be worse compared to CMDNet with correctly trained $\boldsymbol{\theta}$. Below $E_\text{b}/N_0 = 6$ dB, the measure of cross-entropy is indeed higher in contrast to accuracy indicating a worse soft output quality compared to correctly trained CMDNet (and lower above 6 dB). As errors in the low SNR region occur more often, the cross-entropy loss is usually dominated by this region and hence parameters are optimized to perform well there. This causes biasing of CMDNet's soft output and detection accuracy towards performing well at low SNRs. In correlated channels, errors are more frequent in the low SNR region than in i.i.d. channels which explains the observed behavior. In fact, these observations explain why we used a wide training SNR interval of $E_\text{b}/N_0 \in [4, 27]$: to improve accuracy also for high SNR. Further, this means we could use original parameters $\boldsymbol{\theta}$ for low and mismatched $\boldsymbol{\theta}_{\text{mismatch}}$ for high SNR.
- *Basically, all explanations boil down to different weighting of SNR regions according to error frequency which results in respective soft output measures. Note that we can achieve this different weighting also using the MSE instead of the Cross-Entropy (CE) loss as described in Appendix A.3.3.*

A.4 Extensions to CMDNet

Besides the surprisingly good performance of CMDNet in MIMO systems considering its low complexity, we also became aware of some limitations in

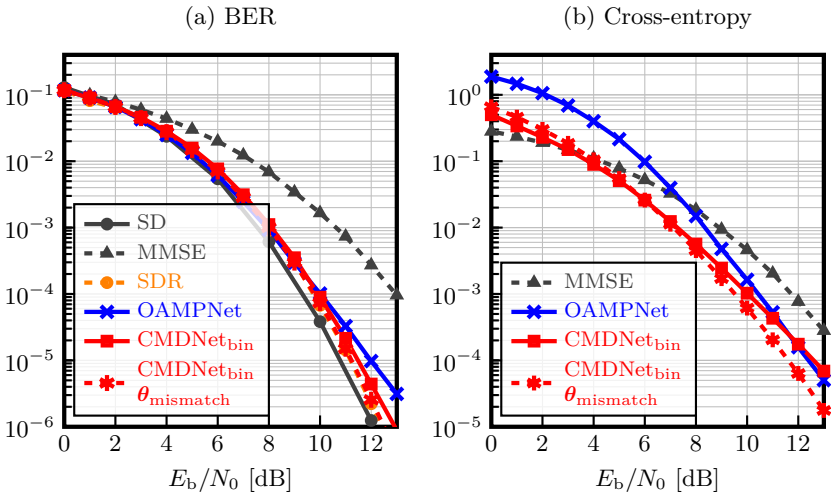

Figure A.9: BER curves and cross-entropy for CMDNet with mismatched parameters optimized for a i.i.d. Gaussian channel model when applied to a correlated 64×32 MIMO system with QPSK modulation. The correlation matrices were generated according to a One-Ring model with 20° angular spread and 120° cell sector. Effective system dimension is 128×64 and for iterative algorithms $N_{\text{it}} = N_{\text{L}} = 64$.

the numerical evaluation. This has stimulated multiple ideas to improve the default structure. In this section, we briefly present these ideas and show first numerical results.

Furthermore, we note that CMDNet has been so far applied to the example of a linear Gaussian MIMO channel. But thanks to its generic nature, CMDNet may also be applied to non-linear detection problems of other model-based research domains. We note that a deeper analysis falls outside the scope of this thesis.

A.4.1 Parallel CMD

One drawback of CMD and CMDNet is that it requires solving a non-convex optimization problem, i.e., the relaxed MAP problem (3.12c), being Nondeterministic Polynomial time (NP) hard. Therefore, we followed a steepest gradient descent approach to compute a computationally tractable solution. This means only convergence to a local solution is guaranteed. Since the Gaussian approximation of the interference terms becomes more accurate with increasing massive MIMO system dimensions, we conjecture

that the original problem can be replaced by a convex one with high fidelity. Hence, CMD's local solution may often coincide with the global one. Indeed, the numerical evaluation in Chapter 3 shows that performance is excellent in high dimensional systems, e.g., 32×32, but deteriorates in conventional small MIMO systems, e.g., 8×8. The latter is also true for higher-order modulation, e.g., 16-QAM, where a larger number of classes has to be detected.

To overcome trapping in a local optimum, we follow an idea from optimization theory: the combination of grid search with local optimization into multi-start strategies [Rao19]. Hence, our basic idea is to evaluate CMD for multiple starting points of gradient descent to increase the probability of finding the global optimum. In Fig. A.10, we show this approach in detail. Owing to its parallel structure, we call it Parallel Concrete MAP Detection (CMDpar). It can be summarized as follows:

1. We use multiple (N_{P}) branches of CMDNet (each with different parameters $\boldsymbol{\theta}_i$ and starting point $\mathbf{G}_i^{(0)}$).
2. We choose the result which minimizes the objective function $L\left(\mathbf{G}^{(N_{\mathrm{L}})}, \tau_i^{(N_{\mathrm{L}})}\right)$ from (3.12c).
3. We include the starting points $\mathbf{G}_i^{(0)}$ of CMDNet in the trainable parameters $\boldsymbol{\theta}$. By default, we define $\mathbf{G}_0^{(0)} = \mathbf{0}$ to include default CMD as one branch.
4. We train CMDpar w.r.t. $\tilde{\mathbf{x}}$ or $\sigma_{\tau^{(N_{\mathrm{L}})}}\left(\mathbf{G}^{(N_{\mathrm{L}})}\right)$ according to the corresponding loss function MSE or cross-entropy, respectively.

In contrast to heuristic gradient descent starting point initialization of CMD with, e.g., randomization, note that we optimize the starting points $\mathbf{G}_i^{(0)}$ as part of the trainable parameters $\boldsymbol{\theta}$ to increase detection accuracy. In Fig. A.11, we show first numerical results of the BER performance and cross-entropy of *binary* CMDpar with $N_{\mathrm{L}} = 16$ layers and $N_{\mathrm{P}} = 3$ branches trained for $N_{\mathrm{e}} = 300,000$ iterations assuming the following starting point initialization including the default $\mathbf{G}_0^{(0)} = \mathbf{0}$. For better interpretability, we first heuristically choose the starting point in terms of $\tilde{\mathbf{x}}_i^{(0)}$ as

$$\left\{\tilde{\mathbf{x}}_0^{(0)}, \tilde{\mathbf{x}}_1^{(0)}, \tilde{\mathbf{x}}_2^{(0)}\right\} = \{\mathbf{0}, +0.5 \cdot \mathbf{1}, -0.5 \cdot \mathbf{1}\} \tag{A.43}$$

such that, by inverting (3.17e) or (A.10e), i.e.,

$$\mathbf{s}_i^{(0)} = 2\tau \cdot \tanh^{-1}\left(\tilde{\mathbf{x}}_i^{(0)}\right) - \ln\left(1/\boldsymbol{\alpha} - 1\right), \tag{A.44}$$

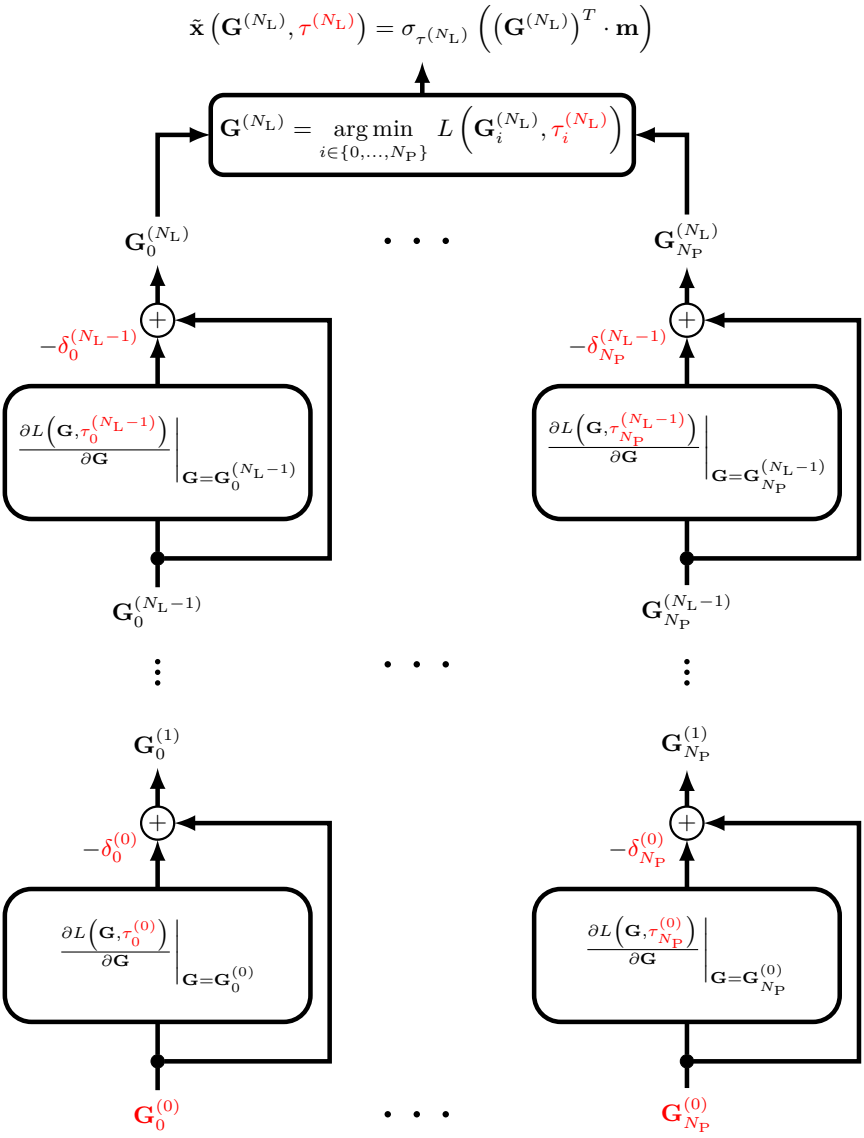

Figure A.10: Algorithmic structure of the parallel extension to CMDNet: CMD-par. Trainable parameters are shown in red.

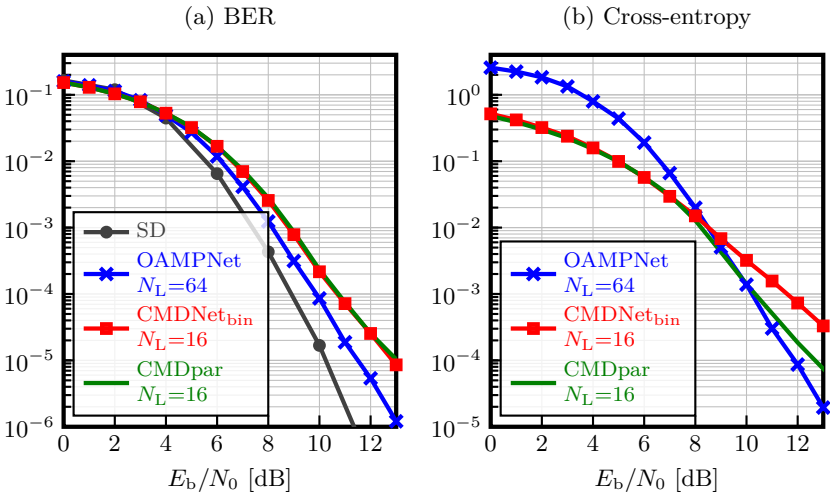

Figure A.11: BER curve and cross-entropy of CMDpar in a 32×32 MIMO system with modulation. Effective system dimension is 64×64.

and assuming default $\boldsymbol{\alpha} = [0.5, 0.5]$ and $\tau = 1$, we arrive at

$$\left\{\mathbf{s}_0^{(0)}, \mathbf{s}_1^{(0)}, \mathbf{s}_2^{(0)}\right\} = \{\mathbf{0}, \ln(3) \cdot \mathbf{1}, -\ln(3) \cdot \mathbf{1}\} \ . \tag{A.45}$$

In fact, we observe that accuracy of CMDpar is actually similar to one branch, i.e., to default CMDNet trained for $N_e = 100,000$. The reason may lie in the learned parameters $\mathbf{s}_i^{(0)}$: Their values are quite small in the order of 10^{-4}-10^{-2} and symmetric around 0. This means numerical optimization ends in the non-diverse but equal point $\mathbf{s}_i^{(0)} = \mathbf{0}$. However, the other learned parameters step size δ and softmax temperature τ differ per branch which suggests each branch of CMDpar to compute a different output. Considering the cross-entropy, a measure of soft output quality (see Fig. A.11 (b)), this may be the reason for an improvement of soft outputs. In particular in the high SNR region, CMDpar beats default CMDNet since there interference removal is crucial and hence a parallel structure following different hypotheses similar to MAP detection may be beneficial. In conclusion, although being a promising idea, CMDpar does not allow for learning of different gradient descent starting points without fine-tuning of the training process. However, we note that further investigations are required to evaluate its full potential.

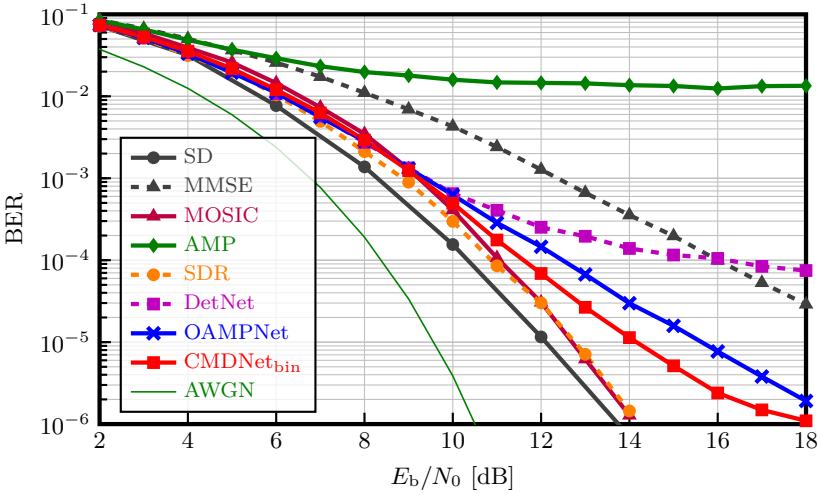

Figure A.12: BER curves of various detection methods in a correlated 64×32 MIMO system with QPSK modulation. The correlation matrices were generated according to a One-Ring model with 10° angular spread and 120° cell sector. Effective system dimension is 128×64 and for iterative algorithms $N_{\text{it}} = N_{\text{L}} = 64$.

A.4.2 HyperCMD

Besides small MIMO systems and higher-order modulation, also detection in spatially correlated channels (see Appendix A.2.2) typical for the uplink of massive MIMO systems may be challenging. As an example, we give Fig. A.12: We observe that the BER performance of learning-based approaches degrades significantly in the high SNR region, assuming a local scattering model for channel correlation with a small angular spread of $\varphi_\Delta = 10^\circ$ typical for flat rural areas. In contrast, MMSE Ordered Successive Interference Cancellation (MOSIC) is robust for correlated channel matrices $\mathbf{H}$ with low condition numbers since it cancels symbol interference according to a SNR ordering of every row based on a Sorted QR Decomposition (SQRD). The reason for robustness of the SDR technique may lie in the close approximation of the original objective function. Nevertheless, CMDNet proves to be superior to other learning-based approaches DetNet and OAMPNet especially in the high SNR region. This is surprising since CMDNet has lower complexity and is optimized over a wide range of different user positions and hence correlation matrices. Unlike the i.i.d. Gaussian channel case, CMDNet is not amortized over a single but a wide range of channel

statistics.

It becomes clear from Fig. A.14 that the accuracy of learning-based approaches is limited in the high SNR region when it comes to massive MIMO models with channel correlation. As noted in [KAHF20], this effect is much more evident in practical correlated channels and detection requires a more sophisticated approach such as the well-suited heuristic MOSIC. Therefore, the authors of [KAHF20] proposed MMNet, a low complex model-based DNN, to make online training and thus fast and tight adaptation to new channel matrix realizations feasible, alleviating the need to consider channel statistics in algorithm design.

Since online training, e.g., by variants of SGD, is in general wasteful and relies on specific assumptions such as a long coherence time, the authors from [GAH20] introduce the concept of hypernetworks [BHV+16; HDL16; ZSBL19] to the overall design of MMNet and call this approach Hypernetwork-based MIMO Detection (HyperMIMO). Hypernetworks were first used in the context of image recognition [BHV+16]. It is an additional DNN usually trained for a limited number of input samples to compute optimized parameters of a DNN-based detector. Its aim is generalization, i.e., to predict the parameters of a DNN given a new input sample. By this means, it is able to recognize other objects of the same class, e.g., a dog, without the need for retraining.

Our idea is to apply the concept of hypernetworks to CMDNet. This means:

1. We use the default structure of CMDNet.
2. We introduce a hypernetwork with hyperparameters $\boldsymbol{\psi}$ to compute parameters $\boldsymbol{\theta} = f\big(\mathbf{H}, \sigma_{\mathrm{n}}^2 | \boldsymbol{\psi}\big)$ of CMDNet for each channel matrix and noise variance input.
3. We optimize the hyperparameters $\boldsymbol{\psi}$ of the hypernetwork w.r.t. $\tilde{\mathbf{x}}$ or $\sigma_{\tau^{(N_{\mathrm{L}})}}(\mathbf{G}^{(N_{\mathrm{L}})})$ according to the corresponding loss function MSE or cross-entropy, respectively.

We name this approach Hypernetwork-based Concrete MAP detection (HyperCMD) and depict its structure in Fig. A.13. Further, we note that the design and training of a hypernetwork is not trivial: The main bottleneck of the concept was identified in [BHV+16] to be the quadratic size of the output space since the weight matrices of a DNN is now parameterized by an input vector. In this special case, we further have a channel matrix $\mathbf{H}$ as input which is thus also of quadratic size. For example, HyperMIMO therefore uses a more elaborated design leveraging factorized linear layers, weight sharing and QR Decomposition (QRD) to lower the number of inputs as well as

outputs and consequently parameters. Hence, the size of the hypernetwork $f\big(\mathbf{H}, \sigma_\mathrm{n}^2|\boldsymbol{\psi}\big)$ and the number of its hyperparameters $\boldsymbol{\psi}$ is reduced. Further, it is trained for fixed channel statistic to ensure training convergence.

Numerical evaluation of HyperCMD in a 32×32 MIMO system for fixed i.i.d. Gaussian channel statistic supports these findings. In Fig. A.14, we show the results with two different hypernetwork architectures $\boldsymbol{\theta} = \left\{\tau^{(0)}, \ldots, \tau^{(N_\mathrm{it})}, \delta^{(0)}, \ldots, \delta^{(N_\mathrm{it}-1)}\right\} \in \mathbb{R}^{(2N_\mathrm{it}+1)\times 1} = f(\mathbf{H}, \sigma_\mathrm{n}^2|\boldsymbol{\psi})$ of HyperCMD:

1. Hypernetwork $f_1(\sigma_\mathrm{n}^2|\boldsymbol{\psi})$ has only noise standard deviation σ_n as input. In experiments, we observed that using the noise variance σ_n^2 instead leads only to minor differences. Thus, we use σ_n as in [GAH20] since it has smaller input range making DNN training easier. We define each element of $\boldsymbol{\theta}$ or the function f_1 to be a one-layer Neural Network (NN), with width N_h and ReLU function (see Appendix C) as non-linearity $\rho_\mathrm{relu}(\cdot)$, applied element-wise to allow for close function approximation of every mapping from noise variance to $\boldsymbol{\theta}$:

$$\tau^{(j)} = \left|\mathbf{w}_{2j}^T \cdot \rho_\mathrm{relu}\left(\mathbf{w}_{1j} \cdot \sigma_\mathrm{n} + \mathbf{b}_{1j}\right) + b_{2j}\right|^{-1} \tag{A.46a}$$

$$\delta^{(k)} = \tilde{\mathbf{w}}_{2k}^T \cdot \rho_\mathrm{relu}\left(\tilde{\mathbf{w}}_{1k} \cdot \sigma_\mathrm{n} + \tilde{\mathbf{b}}_{1k}\right) + \tilde{b}_{2k} \tag{A.46b}$$

$$\boldsymbol{\psi} = \left\{\mathbf{w}_{1j}, \tilde{\mathbf{w}}_{1k}, \mathbf{w}_{2j}, \tilde{\mathbf{w}}_{2k}, \mathbf{b}_{1j}, \tilde{\mathbf{b}}_{1k} \in \mathbb{R}^{N_\mathrm{h}\times 1}, b_{2j}, \tilde{b}_{2k} \in \mathbb{R}\right\} \quad \forall\, j \in 0 \ldots N_\mathrm{it}\,, k \in 0 \ldots N_\mathrm{it} - 1\,. \tag{A.46c}$$

   This means all parameters are computed *independently*. As the starting point, we choose weights as $\mathbf{w}_{1j} = \tilde{\mathbf{w}}_{1k} = \mathbf{w}_{2j} = \tilde{\mathbf{w}}_{2k} = 0.01 \cdot \mathbf{1}$ and biases as $\mathbf{b}_{1j} = \tilde{\mathbf{b}}_{1k} = 0$ with width $N_\mathrm{h} = 6$, whereas we set $b_{2j}^{(0)} = \tau_0^{(j)}$ and $\tilde{b}_{2j}^{(0)} = \delta_0^{(j)}$ to the default parameter starting point $\boldsymbol{\theta}_0$ from (3.35) in Sec. 3.5.1. In total, only a low number of $|\boldsymbol{\psi}| = (3N_\mathrm{h}+1)\cdot(2N_\mathrm{it}+1) = 627$ hyperparameters has to be optimized assuming a layer number $N_\mathrm{it} = N_\mathrm{L} = 16$.

2. Hypernetwork $f_2(\mathbf{H}, \sigma_\mathrm{n}^2|\boldsymbol{\psi})$ takes both noise standard deviation σ_n and vectorized channel matrix $\mathrm{vec}(\mathbf{H})$ as input. It consists of two dense, fully-connected layers with Exponential Linear Unit (ELU) activation function $\rho_\mathrm{elu}(\cdot)$ (see Appendix C) and a last linear layer:

$$\mathbf{v}_1 = \rho_\mathrm{elu}\left(\mathbf{W}_1 \cdot \begin{bmatrix} \sigma_\mathrm{n} \\ \mathrm{vec}(\mathbf{H}) \end{bmatrix} + \mathbf{b}_1\right) \tag{A.47a}$$

$$\mathbf{v}_2 = \rho_\mathrm{elu}\left(\mathbf{W}_2 \cdot \mathbf{v}_1 + \mathbf{b}_2\right) \tag{A.47b}$$

$$\mathbf{v}_3 = \mathbf{W}_3 \cdot \mathbf{v}_2 + \mathbf{b}_3 \tag{A.47c}$$

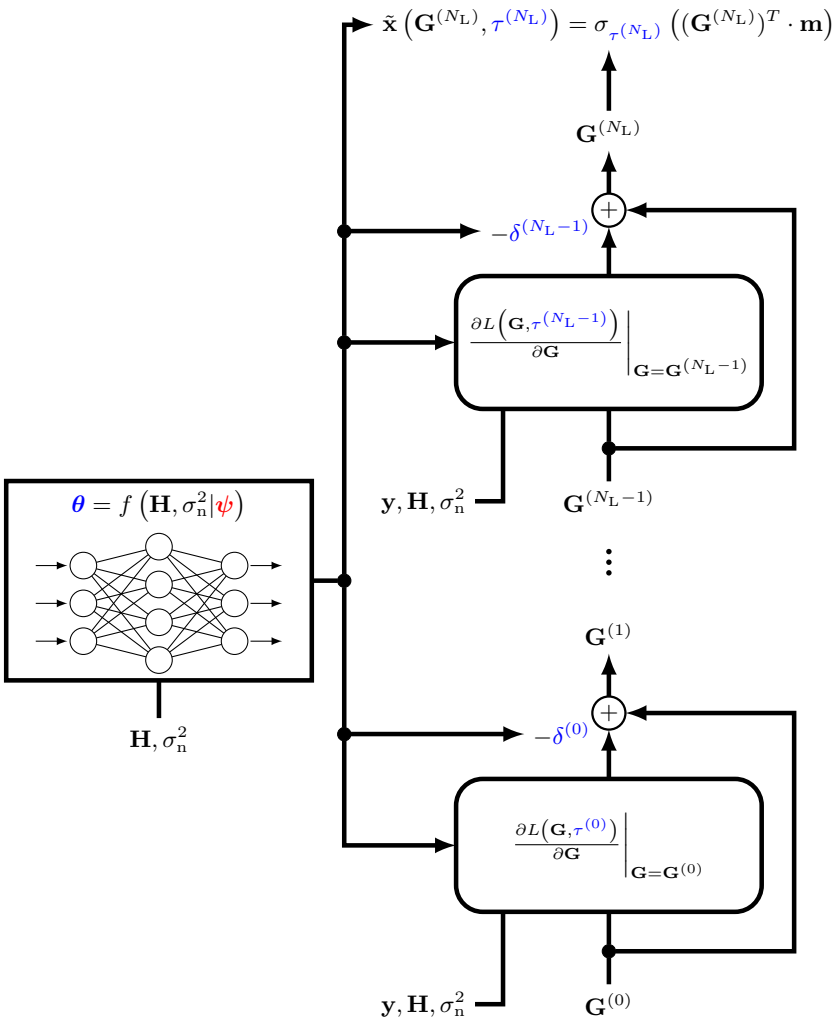

Figure A.13: Algorithmic structure of CMDNet with hypernetwork extension: HyperCMD. Parameters shown in blue are computed by the hypernetwork with trainable hyperparameters in red.

$$\left[1/\tau^{(0)} \dots 1/\tau^{(N_{\mathrm{it}})}\right]^T = \left[|v_{3,1}| \dots |v_{3,N_{\mathrm{it}}+1}|\right]^T \tag{A.47d}$$

$$\left[\delta^{(0)} \dots \delta^{(N_{\mathrm{it}}-1)}\right]^T = \left[v_{3,N_{\mathrm{it}}+2} \dots v_{3,2N_{\mathrm{it}}+1}\right]^T . \tag{A.47e}$$

In contrast to Hypernetwork $f_1(\sigma_{\mathrm{n}}^2|\boldsymbol{\psi})$, all inputs are processed jointly in one DNN. The hyperparameters have the following dimensions:

$$\begin{aligned}\boldsymbol{\psi} = \big\{ &\mathbf{W}_1 \in \mathbb{R}^{N_{\mathrm{h}}^{(1)} \times N_{\mathrm{T}} \cdot N_{\mathrm{R}}+1}, \mathbf{W}_2 \in \mathbb{R}^{N_{\mathrm{h}}^{(2)} \times N_{\mathrm{h}}^{(1)}}, \\ &\mathbf{W}_3 \in \mathbb{R}^{2N_{\mathrm{it}}+1 \times N_{\mathrm{h}}^{(2)}}, \mathbf{b}_1 \in \mathbb{R}^{N_{\mathrm{h}}^{(1)} \times 1}, \mathbf{b}_2 \in \mathbb{R}^{N_{\mathrm{h}}^{(2)} \times 1}, \\ &\mathbf{b}_3 \in \mathbb{R}^{2N_{\mathrm{it}}+1 \times 1} \big\} .\end{aligned} \tag{A.47f}$$

For $l = 1, \dots, 3$, we initialize biases to $\mathbf{b}_l = \mathbf{0}$ and weights $\mathbf{W}_l \in \mathbb{R}^{N_{\mathrm{h}}^{(l+1)} \times N_{\mathrm{h}}^{(l)}}$ by sampling according to the Glorot Uniform Initialization (2.80) from [GB10] known to speed up training convergence (see Chapter 2).

Note that this hypernetwork is dense and has a large input dimension resulting in a large number of hyperparameters. Consequently, efficient training of $f_2(\mathbf{H}, \sigma_{\mathrm{n}}^2|\boldsymbol{\psi})$ is difficult, and we restrict to investigation in a smaller 8×8 MIMO system. We choose the width of the two hidden layers of the hypernetwork to be $N_{\mathrm{h}}^{(1)} = N_{\mathrm{T}} \cdot N_{\mathrm{R}} + 1 = 16 \cdot 16 + 1 = 257$ and $N_{\mathrm{h}}^{(2)} = 75$ like in [GAH20]. Further, we use $N_{\mathrm{it}} = 16$ CMDNet layers so that the output dimension results to $2N_{\mathrm{it}} + 1 = 33$. In this configuration, Hypernetwork $f_2(\mathbf{H}, \sigma_{\mathrm{n}}^2|\boldsymbol{\psi})$ has already $|\boldsymbol{\psi}| = 88,164$ hyperparameters to be optimized decreasing convergence speed. We account for this fact by increasing the number of training iterations to $N_{\mathrm{e}} = 3 \cdot 10^6$, each with a batch size of $N_{\mathrm{b}} = 500$, and make training stable by reducing the learning rate of the optimizer Adam from $\epsilon = 10^{-3}$ to 10^{-4}.

Comparing CMDNet and HyperCMD 1 with small hypernetwork $f_1(\sigma_{\mathrm{n}}^2|\boldsymbol{\psi})$ for $N_{\mathrm{L}} = 16$ layers and a 32×32 MIMO system in Fig. A.14 (a), we observe a slight improvement (≤ 0.5 dB) in detection accuracy from $E_{\mathrm{b}}/N_0 = 8$ dB to 14 dB. This means HyperCMD is able to compute optimized parameters for each noise variance or SNR. The observation is in accordance with our findings from Sec. 3.5.3 and Fig. 3.7 that CMDNet benefits from an optimized parametrization $\boldsymbol{\theta}$ conditioned on σ_{n}^2 for the low and high SNR region, respectively.

If we now apply HyperCMD 2 with expressive hypernetwork $f_2(\mathbf{H}, \sigma_{\mathrm{n}}^2|\boldsymbol{\psi})$, whose input is extended by channel matrix realizations $\mathbf{H}$, to a 8×8 MIMO system (see Fig. A.14 (b)), the improvements are still minor and focused

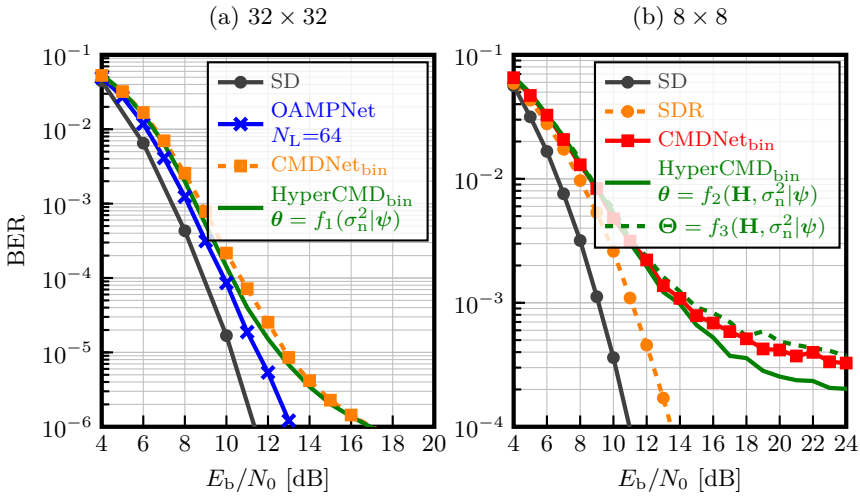

Figure A.14: BER curve of HyperCMD in a (a) 32×32 and (b) 8×8 MIMO system with QPSK modulation. Layer number is $N_L = 16$ and effective system dimension is (a) 64×64 and (b) 16×16.

on the high SNR region ≥ 12 dB where CMDNet's error floor begins. This means adding channel matrices as additional information for computation of CMDNet's parameters does not come with the gain in accuracy we expected.

The reason may lie in various factors:

1. In contrast to HyperMIMO, only 2 scalars, i.e., $\delta^{(j)}$ and $1/\tau^{(j)}$, are multiplied per layer for computation of all symbol entries in (3.14) and (3.17). Indeed, unfolding CMD with untying of scalars per layer and not into vectors or matrices makes sense if we optimize/amortize over a channel statistic $p(\mathbf{H})$, e.g., an i.i.d. Gaussian channel, and not a realization $\mathbf{H}$ since the channel acts the same on all transmit symbols being received with power 1 on average. But if we now compute optimized step sizes and softmax temperatures with and for each realization $\mathbf{H}$, we need a separate and/or combined weighting of symbol entries to gain any benefit. Thus, untying of the scalar parameters $\delta^{(j)}$ and $1/\tau^{(j)}$ to vectors (or even matrices) being multiplied element-wise in (3.14) or (3.17) may be necessary to make the concept work and allow for larger accuracy improvements. However, this comes at the drawback of an increase in the number of parameters $\boldsymbol{\theta}$ by a factor of at least N_T (for untying into vectors), a larger hypernetwork output and hence a larger number of hyperparameters $\boldsymbol{\psi}$.

For first numerical experiments (see Fig. A.14 (b)), we modified Hypernetwork 2 to compute untied element-wise multiplied vectors (collected in a matrix $\boldsymbol{\Theta} = f_3(\mathbf{H}, \sigma_\mathrm{n}^2|\boldsymbol{\psi})$) and adjusted the hidden layer size $N_\mathrm{h}^{(2)} = 392$ to half of the sum of input and output dimension. Unfortunately, we do not observe any accuracy gains after $N_\mathrm{e} = 10^6$ training iterations with standard parametrization w.r.t. both CMDNet and HyperCMD with scalars.

2. Furthermore, overfitting or space and time costs could make learning such a regressor (in particular with untied vectors) infeasible or require more training iterations. Employing QRD to lower the number of parameters by approximately a half to $N_\mathrm{T}(N_\mathrm{T}+1)/2$ did not yield any remarkable improvements in training convergence or speed in our experiments.

3. Maybe, success simply necessitates use of another training parametrization with different batch size or optimizer. The huge number of hyperparameters makes it difficult to pinpoint the exact reason for only minor improvements.

At this point, we note that numerical results indicate that HyperMIMO from [GAH20] achieves similar accuracy to an online trained MMNet. But it was only evaluated for a small 12×6 massive MIMO system and local scattering model with fixed user positions and Gaussian-distributed angles. If application of hypernetworks in large massive MIMO systems is possible and leads to a worthwhile accuracy complexity trade-off, remains an open research question. For example, online training of a small number of parameters of CMDNet could be of lower complexity at inference run time than HyperCMD.

A.5 Chapter Summary

In this chapter, we provided detailed explanations including AMP's non-optimality and an illustrative explanation of the non-convexity of the relaxed objective function. In particular, for the guiding example of the model-based approach CMDNet addressing an algorithm deficit instead of a model deficit, we discussed different training aspects such as the influence of hyperparameter choices in the context of established ML practices. The key results are manifold:

- We provided the complete derivation of binary CMD, and proved that CMD and binary CMD are different algorithms for BSPK symbols.

- We introduced the cross-entropy loss as a measure of soft information to enable deeper investigations.

- We contributed that the main benefit of validation loss, i.e., tracking the generalization to unseen data points, transforms into being a less noisy observation of the current training progress when using infinite model-based generated data.

- We found that using the MSE loss in CMDNet optimization instead — similarly to related data-driven MIMO detection approaches — leads to comparable performance, primarily due to a different SNR weighting that results in excellent BER in high-SNR regions. This outcome is somewhat surprising, as this practice is not well-grounded in theory, yet it helps explain the empirical success of these methods. The multi-loss does not seem to be suited for model-based approaches, as it tends to limit maximum performance rather than enhance it.

- We investigated the influence of training parameters such as the optimization algorithm, batch size and number of layers. In particular, we figured out that Adam — known to generalize worse than SGD — indeed performs comparable using an infinite model-generated data.

- We clarified why we exploit offline training from both theory and practice: It lowers complexity at the cost of accuracy by optimizing over a whole range of input statistics. This avoids the need for online retraining and enables extensive training to high accuracy before deployment.

- We revealed that CMDNet heavily reacts to starting weight initializations requiring heuristics different from standard DNN practice. In combination with the offline training philosophy, we showed that evaluation of CMDNet's training convergence compared to other approaches is not conclusive without further assumptions.

- A simple online learning analysis reveals that a default low-complex DNN does not lead to competitive performance with reasonable training complexity and that CMDNet performance only increases to a small extent. A deeper analysis is required.

- We explained that a mismatch of CMDNet parameters can be considered as overfitting. Surprisingly, in all considered scenarios, overall detection accuracy becomes comparable or even superior, indicating CMDNet's robustness against various mismatches.

- Finally, we proposed two extensions of CMDNet: CMDpar and HyperCMD. The former is motivated by the non-convexity of the optimization problem tackled by extending CMDNet by parallel processing branches with different starting points akin to a multi-start strategy. In the latter, we utilize the concept of hypernetworks to predict optimized parameters of CMDNet for each channel and noise variance realization. First simulation results show small improvements not justifying the extension's higher complexity. It remains to future research to clarify the full potential of CMDpar and HyperCMD.

In conclusion, the main contribution is that we shed new light on ML practices in the context of communications design and resulting changes with model-based approaches. A common problem remains the large hyperparameter space that may be tackled with a more efficient random search.

Appendix B

Semantic Communications Extensions

B.1 Overview

In this chapter, we extend our investigations on semantic communication systems from [BBD23; BBD24], corresponding to Chapter 4 and Chapter 5, respectively, by aspects that were cut or have fallen short. This includes philosophical extensions, a comparison of the two semantic system models from Chapter 4 and [BSW+23] in light of the literature, SINFONY design considerations and its competitiveness compared to classic digital communications designs. Furthermore, we incorporate ideas that have been published in [BBD23]'s former version [BBD22], including how semantics can be exploited in conventional digital designs and the example of floating-point transmission. Lastly, we reflect deeper on RL-SINFONY.

B.2 Extended Analysis on SINFONY

In this section, we expand our analysis of SINFONY by exploring philosophical aspects of semantics, comparing system models from Chapter 4 and [BSW+23], and presenting new simulation results that compare SINFONY with non-overall semantic communication system designs and assessing the design choices made for SINFONY.

B.2.1 Philosophical Extensions

If we consider human semantics, it becomes difficult to describe the actual meaning in terms of mathematical modeling. In case of image classification, we need to make use of labeled datasets, i.e., samples, as noted in Chapter 4, to include human-based semantics as usually done in the domain of ML.

If we go one step further, we realize that the human brain itself is an information processing system that recognizes these images and extracts the relevant information. The human being is part of the society (with its own goals), nature and ultimately the whole universe. These aspects are not included in Shannon's theory and philosophers still try to combine them in a unified theory of information [Flo09; Hof13].

Søren Brier states in [Bri08; Bri13]: *"Modern systems thinking views nature as containing multilevel, multidimensional hierarchies of interrelated clusters, which together form a heterogeneous general hierarchy of processual structures: a 'heterarchy'. Levels emerge through emergent processes when new holons appear through higher-level organization. [...] Meaning is generated through the entire heterarchy [...]"*. Holon is something that functions both as an independent, self-sufficient unit and as an integral component of a larger whole [Koe68]. For example, in biology, a cell is a holon, and in society, the human is a holon. This means the term meaning is closely related to the phenomenon of emergence and connects to organization in the universe.

Recently, a new interdisciplinary theory of emergence was proposed in [RGL+24], where Shannon's information theory plays a critical role in understanding how macroscopic processes emerge from microscopic dynamics. A key concept is informational closure, which quantifies the extent to which a macroscopic process can predict its future states without needing information from its underlying microscopic processes. This is formalized using Shannon's Mutual Information (MI), which measures how much information the macro-scale retains about its future, independent of microscale data. By integrating information-theoretic concepts like MI with principles from automata theory and computational mechanics, the theory establishes a unified framework for reasoning about the emergence of self-contained macroscopic behaviors in complex systems.

Given this more philosophical or interdisciplinary view, we conclude that Shannon's information theory can be seen as a key ingredient for also understanding and designing semantic communication systems from a broader perspective, especially when it comes to mathematical modeling or ML.

Semantic Hierarchical Levels

We emphasize that semantics can be incorporated at progressively increasing levels of complexity, forming a hierarchy of semantic layers. This concept is explained in Chapter 4 and illustrated in Sec. 4.5.2 with an example based on sensors and power plant control. Another technical example is as follows: At the first semantic layer, a RV $\mathbf{z}_1$ might represent the floating-point values of continuous variables processed by a digital processing unit, as shown in the numerical example of [BSW+23] or in Appendix B.4. Here, the bits $\mathbf{s}$ carry specific meanings, with some contributing more significantly to the target variable $\mathbf{z}_1$ through a function $\mathbf{z}_1 = f(\mathbf{s})$. Beyond this initial layer, a second semantic layer could interpret $\mathbf{z}_1$, for instance, by classifying images or sensor data $\mathbf{z}_2$. This layering process can continue up to the overall task $\mathbf{z}_3$, such as power plant control. This hierarchical structure enables the addition or removal of context as needed, making the system optimizable with respect to the variables $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \ldots, \mathbf{z}_i$ at different semantic levels. In the context of task-oriented communication, this implies that individual subtasks may contribute to a broader, overarching goal.

Additionally, these semantic layers can represent both microscopic and macroscopic processes, as in the phenomenon of emergence, bridging semantic communication with the findings in [RGL+24]. A deeper exploration of this connection is reserved for future work.

B.2.2 Alternative Semantic System Model

In this section, we shed light on why Chapter 4 and [BSW+23] include different views on semantic RVs.

In [BSW+23], the semantic RV $\mathbf{m} \in \mathcal{M}_m^{N_m \times 1}$ from domain $\mathcal{M}_m$ of dimension N_m represents messages of a factor node describing key distribution parameters of the exploration RV $\mathbf{s}$ to be exchanged between neighboring agents. We optimize communications via the InfoMax principle to preserve as much information as possible in the received signal about the factor node message $\mathbf{m}$. Notably, by doing so, we can provide a probabilistic estimate, i.e., the approximate posterior $q_{\boldsymbol{\varphi}}(\mathbf{m}|\mathbf{y})$. Since the message passing algorithm can make use of the probabilistic input including the uncertainty of the communication channel, this can be seen as a step towards a semantic design, i.e., communication-aware exploration and exploration-aware communication. As pointed out in [BSW+23], the design considerations can also be applied to semantics-agnostic settings. These can be seen as Joint Source-Channel Coding (JSCC) of $\mathbf{m}$ backed by semantic communication research [GQA+23].

In contrast, in Chapter 4, we use a different system model. We clearly

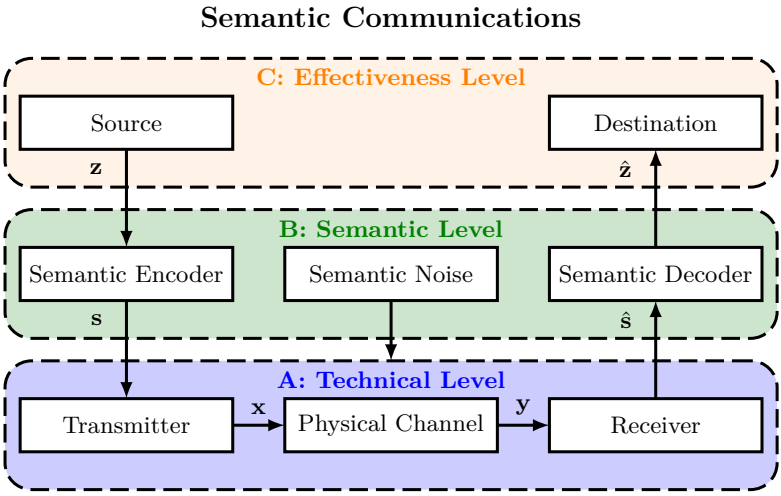

Figure B.1: Three levels of semantic communications according to Weaver.

differentiate between the semantic RV $\mathbf{z}$ and the observation or source $\mathbf{s}$, e.g., corresponding to the meaning of an image and the observation of that image, respectively. We aim to reconstruct the semantic RV $\mathbf{z}$ from the observation $\mathbf{s}$ with SINFONY based on the InfoMax principle. Transferred to the scenario in [BSW+23], this would mean that we see the semantic RV $\mathbf{m}$ now as an observation $\mathbf{s}$ and aim to extract its meaning $\mathbf{z}$ for the computations inside the message passing algorithm to enable more efficient communication. Differentiating between observation and semantics, we can clearly or explicitly define semantics leading to a more complete and consistent view.

Taking a look at a visualization of the three levels by Weaver depicted in Fig. B.1 from [BBD+11a; BBD+11b] and adopting the view from Chapter 4, we notice that the approach of [BSW+23] only operates after the semantic encoder on observation $\mathbf{s}$ and tailors the receiver output w.r.t. $\mathbf{s}$ to the semantic decoder. However, SINFONY from Chapter 4 also sees the semantically encoded message or observation $\mathbf{s}$ but then combines receiver and semantic decoder into one joint semantic receiver to reconstruct the semantic RV $\mathbf{z}$ directly.

Recalling that we propose multiple hierarchical semantic levels — beyond just Levels B and C — note that with SINFONY, communication does not necessarily terminate at the effectiveness level. Instead, it may operate at even higher semantic levels, which can themselves be followed by further

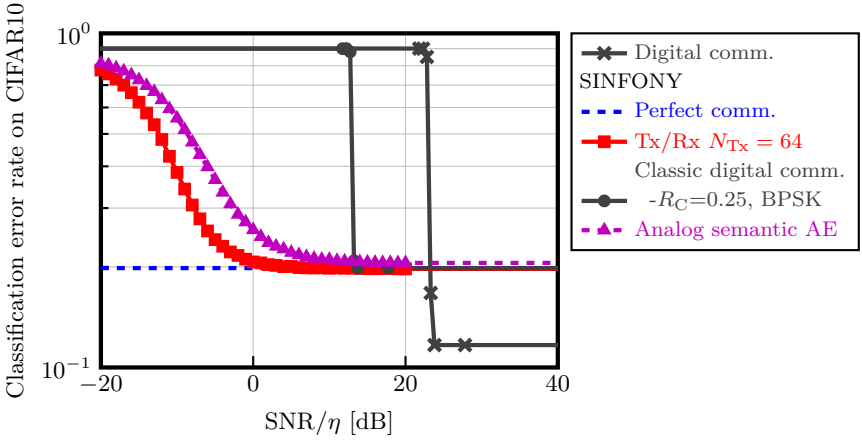

Figure B.2: Classification error rate of SINFONY with different kinds of optimized Tx/Rx modules and central image processing with digital image transmission on the CIFAR10 validation dataset as a function of normalized SNR.

abstractions.

As a concluding remark, in the scheme of Fig. B.1, one possible approach is to assume the technical level including transceiver and physical channel to be given. In [BBD+11a; BBD+11b], this is referred to as the semantic channel. Based on a standard communication system described by pdf $p(\hat{\mathbf{s}}|\mathbf{s})$, we could optimize the semantic encoder and decoder for a semantic transmission of $\mathbf{z}$. We elaborate further on this idea in Appendix B.3.

B.2.3 SINFONY vs. Classic Design on CIFAR10

In [BBD23] or Chapter 4, we compared the performance of semantic communication, specifically SINFONY, with that of a classic digital design. However, the evaluation was limited to the MNIST dataset. Now, we aim to provide a more detailed analysis by evaluating the comparison for the more complex CIFAR10 dataset.

The results on the CIFAR10 validation dataset in Fig. B.2 show that the performance gap between SINFONY and the digital image transmission (Digital comm.) becomes even larger, with a difference of roughly 25 dB compared to the MNIST scenario (see Fig. 4.7 in Sec. 4.6.4). Note that Digital comm. classifies the entire digitally transmitted image $\mathbf{s}$ at once,

allowing it to extract features along the image edges, unlike the distributed processing used in SINFONY. As a result, it outperforms other approaches at high SNR. However, this advantage may not apply to other scenarios: for example, joint feature processing of two images from different perspectives may not offer any performance benefits. Neglecting this technical aspect, even SINFONY with digital transmission of features $\mathbf{r}$ (SINFONY – Classic digital comm.) performs 10 dB better. This demonstrates the inefficiency of the classic design (Digital comm.).

Most notably, SINFONY outperforms SINFONY – Analog semantic AE by 5 dB. The latter approach refers to SINFONY trained for perfect communication links and paired with Transmitter (Tx) and Receiver (Rx) modules consisting of an analog semantic AutoEncoder (AE), optimized for feature transmission of $\mathbf{r}$ [BBD23]. The advantage of a semantic overall design, where all components are jointly optimized w.r.t. the semantic RV $\mathbf{z}$, becomes even more apparent compared to the MNIST scenario (see Fig. 4.7 in Sec. 4.6.4). We conclude that a semantic design is now also well-motivated from numerical experiments.

B.2.4 Alternative SINFONY Designs

In [BBD23] or Chapter 4, we investigated one particular SINFONY design: We assumed a uniformly distributed training SNR $\mathrm{SNR}_{\mathrm{train}} \in [-6, 4]$ dB and that all received signals are processed by the same Rx module. Further, Tx and Rx modules were assumed to consist of one intermediate ReLU layer.

Now, we aim to shed light regarding our design choice by comparing to different parametrizations shown in Fig. B.3. With $\mathrm{SNR}_{\mathrm{train}} \in [6, 16]$ dB, we can clearly observe that performance degrades at low SNR. This is no surprise considering that SINFONY cannot learn to protect the features during transmission effectively observing only little noise during training. However, at high SNR, accuracy improves slightly.

We also modify the Rx module such that all received signals are processed by independent ReLU layers — rather than shared ones — before being combined by the GlobalAvgPool2D layer. As an alternative, we also experiment with processing all signals jointly by one ReLU layer of width $N_{\mathrm{w}} = 4 \cdot N_{\mathrm{Tx}}$ equaling the total length of all received signals. Whereas the difference in performance of both approaches seems negligible, we can observe a 2 dB gap at low SNR compared to the standard SINFONY design with shared Rx modules. Making the receive layers more flexible could account for the difference in the four image patches and hence improve overall performance and efficiency. Overall, however, the benefit is minor and thus motivates the design choice of shared Rx modules based on image and channel assumptions

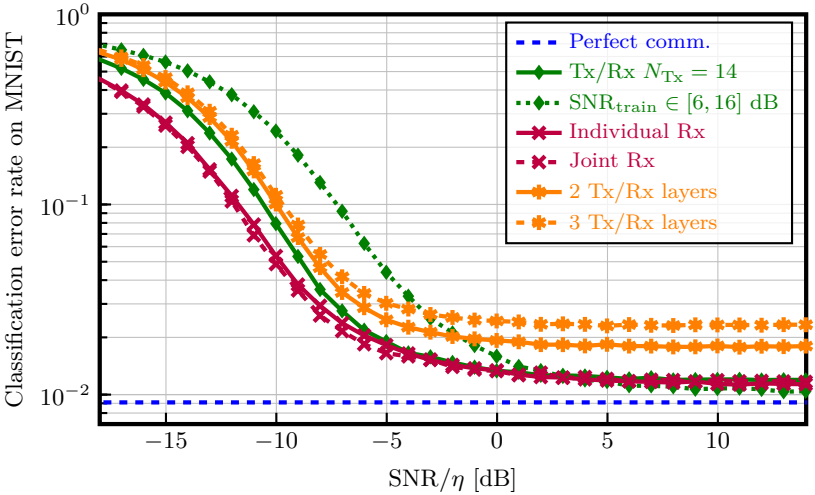

Figure B.3: Classification error rate of different SINFONY designs, i.e., different training SNR, Rx module design, and number of Tx/Rx layers, on MNIST as a function of SNR.

(see Chapter 4).

When using multiple, i.e., two or three, Tx/Rx layers, performance does not improve but degrades. At this point, we can only make the assumption that training for more than $N_{\mathrm{e}} = 20$ epochs is required which can be expected with a deeper architecture. Maybe, simply a different hyperparameter setup would suffice.

B.3 Semantic Communication in a Classic Design

Including various details of an application, i.e., the semantics, into the communication problem would challenge the conventional communication system design of, e.g., the most recent mobile communication standard 5G. Based on Fig. B.4, we will explain if it is possible to include the semantics in classic digital communications design and where the pitfalls lie.

In today's conventional systems, the semantics or task still plays a minor role since source encoding completely decouples the application context from the communication system. A first step towards a semantic design have been requirement profiles as they address the communications needs of an

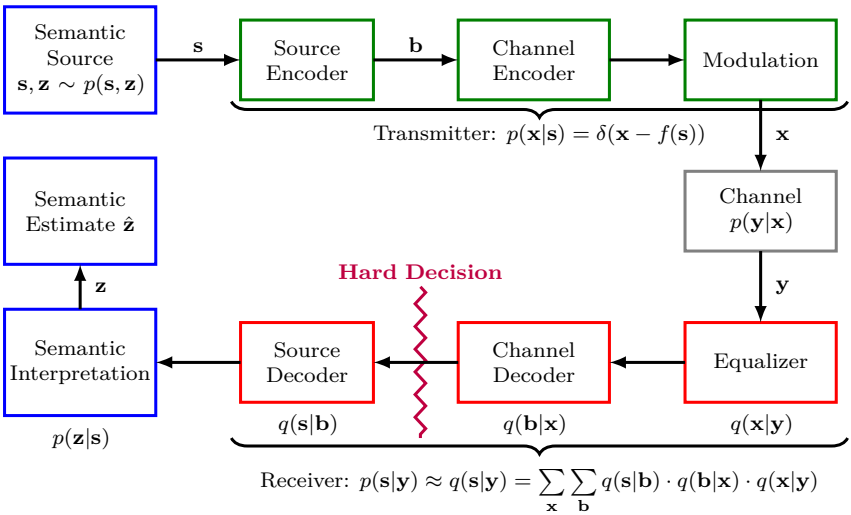

Figure B.4: Conventional digital communication system design and introduction of semantics.

application such as data rate, latency, and power in a more general sense. However, services or Quality of Service like in 5G seem to be a crude interface to reflect its requirements.

First, the source signal $\mathbf{s}$ is encoded by the source encoder for redundancy reduction, encoded with a channel code for error protection and finally modulated for transmission through a channel. All these steps are reversed with respective separated functional blocks at the receiver side. Separation into single blocks is usually preferred since optimization of all blocks together was too difficult/complex in the past. Assuming probabilistic models with factorization between these blocks at the receiver, we arrive at message passing schemes enabling the flow of soft information, e.g., between equalizer $q(\mathbf{x}|\mathbf{y})$ and channel decoder $q(\mathbf{b}|\mathbf{x})$. Message passing is indicated by the integral/summation operation to obtain $q(\mathbf{s}|\mathbf{y})$.

In particular, Shannon proved with the separation theorem that separate source and channel coding is optimal for large block-lengths and point-to-point transmission [GRV03]. As a result, source coding (also known as data compression) and channel coding mainly have been investigated independently in the last decades. However, the theorem does not hold for short block lengths or multi-point communication and optimal source-channel communication does not necessarily imply that coding must be used at all [GRV03].

Famous examples of generic source codes include Huffman codes and Lempel-Ziv compression minimizing redundancy for a sequence of i.i.d. source signals and ergodic sources, respectively. Besides, many application-specific compression techniques have been developed for image, video, audio, voice and language transmission. Human perception is exploited for compression including formats such as mpeg, mp3 and vorbis [PB15]. For channel coding, the recent main breakthroughs in the development of powerful codes reaching the Shannon limit have been Turbo, Low-Density Parity-Check (LDPC) and Polar codes.

Now, we explain the major drawback of the conventional communications design when it comes to semantics: As noted in Sec. 4.5.2, it only accounts for the entropy $\mathcal{H}(\mathbf{s})$ of the observation $\mathbf{s}$ but not the entropy $\mathcal{H}(\mathbf{z})$ of the semantic RV $\mathbf{z}$ behind. Further, we assume $\mathcal{H}(\mathbf{z}) \leq \mathcal{H}(\mathbf{s})$, meaning that the actual semantic uncertainty or information content is less than or equal to the source entropy $\mathcal{H}(\mathbf{s})$. For example, for lossless semantic transmission according to the source-channel separation theorem [CT06, Th. 7.13.1], the product of channel coding rate R_C and channel capacity C must exceed the product of source coding rate R_S and source entropy:

$$\mathcal{H}(\mathbf{z}) \leq \mathcal{H}(\mathbf{s}) \leq R_\mathrm{S} \cdot \mathcal{H}(\mathbf{s}) \leq R_\mathrm{C} \cdot C \leq C\,. \tag{B.1}$$

This implies that a channel code with a higher, more bandwidth-efficient rate R_C would be sufficient for transmitting $\mathbf{z}$, even though the reception of $\mathbf{s}$ might become lossy. However, reducing $R_\mathrm{C} \cdot C$ below $R_\mathrm{S} \cdot \mathcal{H}(\mathbf{s})$ is problematic because errors cannot be tolerated by design: Most source coding standards use VLC such as Huffman coding, which makes them highly sensitive to errors during decoding [ZPZ+12]. Specifically, decoding errors can result in incorrect bit sequence lengths after source decoding, rendering the communication system's output meaningless in terms of semantic output. Therefore, channel decoders are usually designed to achieve a low Frame Error Rate (FER), i.e., these only allow hard decisions to be propagated. The last point implies that there is usually an "information barrier" between channel and source decoders as indicated in Fig. B.4: Uncertainty, which is equivalent to Shannon information, cannot be propagated to higher layers for use by the application. This limitation makes designing a semantic receiver, given a standard transmitter and with or without standard receiver blocks (as discussed in Appendix B.4), very challenging in practice.

Further, powerful channel codes oftentimes have waterfall regions which amplifies the "cliff effect" [BKG19]: Either channel capacity is above the code rate and transmission is nearly deterministic or the link fails. This means that multiple codes with rates adapted for certain SNR regions are required and the complexity of the communication system grows.

A second weak point of the conventional design is that the required large block lengths for source and channel coding as well as the interleavers for statistical decoupling of the processed symbols or bits, e.g., between channel decoder and equalizer, add a huge amount of latency. We note that interleavers are required since standard decoders and equalizers are not designed for non-i.i.d. input data with memory.

To overcome these two major design flaws w.r.t. semantics, we conclude that it is crucial to remove the block-wise structure at transmitter and receiver. For example, we can achieve this by means of

1. Joint Source-Channel Coding (JSCC). Recent work considers AEs for this task and has shown performance improvements at low SNR for language, speech and image transmission [FRG18; BKG19; XQLJ21].

2. the SINFONY approach that we outline in Chapter 4.

With both approaches and exploiting analog transmit signals, a trade-off between source and channel coding is enabled at the transmitter. This translates into smooth transitions without waterfall regions as observed in the numerical results of Chapter 4. Further, we lower complexity since multiple codes with respective rates for specific SNR regions are not needed.

Another approach to enable standard-compatible semantic transmission is the use of a proxy network [HWG+25]. In this method, a proxy network is first trained to mimic conventional, non-differentiable communication blocks including source and channel coding — the technical level A in Fig. B.1. This differentiable proxy network is then used to train a semantic encoder and decoder for image transmission. The result is improved bandwidth efficiency while maintaining performance comparable to AE-based JSCC. A major drawback of this approach is that the latency of the conventional communication system is inherited and further increased by the semantic AE.

B.4 Floating-point Number Transmission

In [BSW+23], we demonstrated in a first investigation that by just adapting the receiver to account for semantics in a simple digital transmission scheme for multi-agent exploration, we can achieve a notable performance gain w.r.t. the semantic metric. From a general point of view, it is the example of floating-point number transmission with subsequent computations on a digital system. Note that it is rather a numerical toy example and introduces context on a very abstract level compared to the example of distributed image classification from [BBD23] or Chapter 4.

Here, we revisit this example and amend it by additional investigations. In particular, it was shown in [BSW+23] that also classic problems like unequal error protection that are very close to the technical level can be tackled in the proposed semantic framework. Since the world model is created for interpretation by machines, we deal with model-driven semantics. Referring to Fig. 4.1, an application generates a continuous data value $z \in \mathbb{R}$, which is processed as discrete floating-point number and represented as bits $\mathbf{s} \in \{0,1\}^{N_{\text{bit}} \times 1}$, on digital hardware.

Semantic Source: More precisely, one floating-point value consists of a signed bit, significand and exponent bits that contribute through a weighted sum of the function $z = f(\mathbf{s})$ defined in the standard [IEE19]. This means each bit of the float has a different meaning and is of different importance for our task of reconstructing z. However, relying on digital error-free transmission, each bit would be considered equally important. As a result, there is room for non-perfect and thus more efficient transmission of bit sequences as long as their meaning remains close, e.g., $z = 1.53$ and $\hat{z} = 1.54$. With semantic space in the real-valued domain and without any further detailed knowledge about the higher-level task, it is reasonable to assume that our receiver estimates $\hat{z}$ should be close to the true transmit value z in the MSE sense.

Here, we investigate a more general numerical example compared to the distributed full waveform inversion from [BSW+23]. We assume the semantic RV z to be Gaussian-distributed instead of using statistics of a given dataset. Further, we assume NaN as well as $\pm$inf values do not occur. For computational tractability, we consider 8-bit floating-point numbers (minifloats) with one signed bit, 4 exponent and 3 significand bits.

Transmission Model: Since we want to focus on the key aspect of introducing semantics into the communications design at the receiver side, we use a simple abstraction of the digital transmission system neglecting details, i.e., modern communication protocols with, e.g., strong LDPC or Polar codes: We assume an uncoded Binary Phase Shift Keying (BPSK) transmission of the bits $\mathbf{s}$ of each floating point number over an Additive White Gaussian Noise (AWGN) channel $p(\mathbf{y}|\mathbf{x})$ with noise variance σ_{n}^2 to a receiver.

Approaches: We examine the following approaches for the final decision or estimation of $\hat{z}$, based on either the posterior $p(z|\mathbf{y})$ computed via Bayes' theorem (2.3), or the variational posterior $q_{\boldsymbol{\varphi}}(z|\mathbf{y})$:

- **MAP detection:** Optimal for error-free transmission of bit sequences $\mathbf{s}$, since error rate is minimized. Most likely value z is selected based on $p(z|\mathbf{y})$.
- **Mean estimator:** Optimal for estimation of semantics z in the MSE sense based on $p(z|\mathbf{y})$.
- **Single-bit detector:** As usually assumed in classic digital communications, every bit is considered stochastically independent, i.e., $p(\mathbf{s}) \approx \prod_{i=1}^{N_{\text{bit}}} p(s_i)$, and detected separately. We assume that the prior probability $p(s_i)$ of every single-bit s_i is known. Subsequently, we estimate $\hat{\hat{z}} = f(\mathbf{s})$.
- **Analog transmission:** Analog transmission of z over the AWGN channel is used as a reference curve. We assume N_{bit} power-normalized channel uses with subsequent averaging for a fair comparison.
- **DNN estimator:** For approximate estimation, we set the mean of a Gaussian approximate posterior $q_{\boldsymbol{\varphi}}(z|\mathbf{y})$ to a small Rx DNN shown in Tab. B.1. We take the mean, i.e., the output of the DNN, as the estimate $\hat{z}$.
- **SINFONY:** Moving beyond receiver design of [BSW+23], we can also parametrize the encoder $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{s})$ by a DNN and optimize the resulting semantic transceiver, e.g., SINFONY, via (4.9). Our selected structure is shown in Tab. B.1. Note that normalization of the encoder output across the batch is required to constrain the output power to one.

Training Details: Following the InfoMax principle, we optimize the DNN-based approaches by minimizing the cross-entropy (2.60), which is equivalent to the MSE loss for Gaussian approximate posteriors, as shown in (2.62). Thus, we trained the DNN-based approaches with MSE loss for $N_{\text{e}} = 10,000$ iterations with the stochastic gradient descent variant Adam and a batch size of $N_{\text{b}} = 1000$, performing 10 steps per iteration. To optimize the receiver over a wider SNR range, we choose the SNR to be uniformly distributed within $\text{SNR}_{\text{train}} \in [6, 16]$ dB where $\text{SNR} = 1/\sigma_{\text{n}}^2$ with noise variance σ_{n}^2. We initialize ReLU layers with uniform distribution according to He and all other layers according to Glorot [HZRS15].

Numerical Results: In Fig. B.5, we show the Normalized MSE (NMSE) performance of the investigated approaches from [BSW+23] and SINFONY for Gaussian-distributed semantic RVs as a function of SNR.

Table B.1: DNN-based transmitter and receiver for semantic communications design with floating-point numbers.

Component	Layer	Dimension
Input	float8	N_{bit}
Tx	ReLU	$2N_{\text{bit}}$
	ReLU	$2N_{\text{bit}}$
	Linear	$2N_{\text{bit}}$
	Normalization (dim.)	N_{bit}
Channel	AWGN	N_{bit}
Rx	ReLU	$2N_{\text{bit}}$
	ReLU	$2N_{\text{bit}}$
	Linear	1

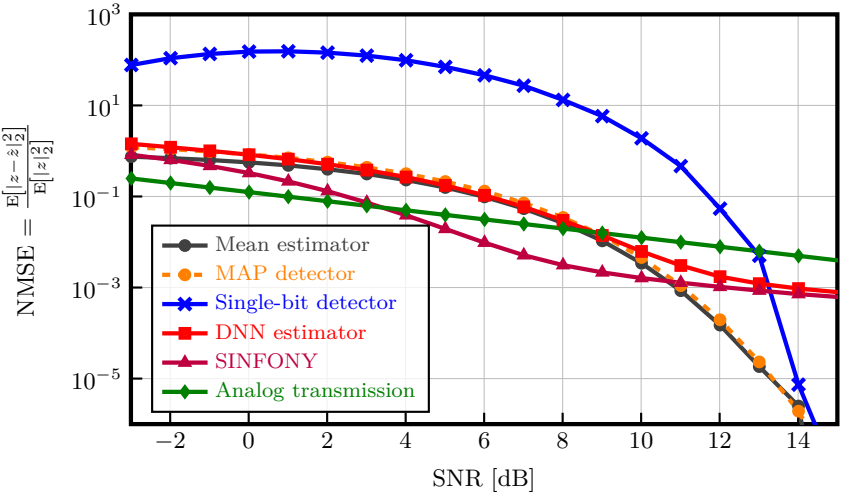

Figure B.5: NMSE as a function of SNR for different semantic and semantic-agnostic transceiver approaches and 8-bit floating-point resolution. We assume uncoded digital BPSK transmission over an AWGN channel.

We observe the same qualitative behavior as in the example of distributed full waveform inversion: The classic approach with subsequent $z = f(\mathbf{s})$ is again clearly inferior in the considered SNR range. Note that we correct NaN and $\pm$inf to the most probable bits based on the individual bit priors $p(s_i)$. Most notably, the semantic low-complex DNN estimator for receiver design performs very close to the optimal receiver, i.e., the mean estimator, at low SNR.

If we also optimize the encoder, we are able to surpass the NMSE of the classic receivers in the low SNR regime. The DNN transceiver can utilize the bandwidth, i.e., the N_{bit} channel uses, more efficiently by performing lossy compression, transmitting the more important bits with higher reliability while disregarding bits that contribute less to z. For high SNR, both DNN receiver estimator and transceiver are not able to increase the precision arbitrarily. We think that this drawback can be overcome by training at higher SNR or incorporating the noise variance as an additional input to the DNN design.

We conclude that even with semantic knowledge on a low hierarchical level (see Appendix B.2.1) about the floating-point structure, a semantic design yields tremendous gains and can be realized with manageable effort, e.g., by low-complex DNNs.

B.5 Reflections on RL-SINFONY

In [BBD24] or Chapter 5, we exploit the SPG to enable training of SINFONY for unknown or non-differentiable channels and iteratively optimize separated transmitter and receiver designs.

Non-differentiable Semantic Reward

Another challenge in semantic communication is the non-differentiability of the semantic objective function or reward $\mathcal{L}(\mathbf{s}, \hat{\mathbf{s}})$, as with BiLingual Evaluation Understudy (BLEU), or its computational intractability, as with Bidirectional Encoder Representations from Transformers (BERT). This challenge can be tackled by using the SPG, leading to the important insight that RL-SINFONY can also be used to train semantic communication systems with a non-differentiable objective $\mathcal{L}(\mathbf{s}, \hat{\mathbf{s}})$ that measures the semantic similarity [LLC+22]. In this case, the gradient of the objective function $\mathcal{L}(\mathbf{s}, \hat{\mathbf{s}})$ w.r.t. the decoder $q(\mathbf{s}|\mathbf{y})$ needs to be approximated in the same way as the channel gradient in [BBD24]. For example, if $\mathbf{s}$ is discrete, a softmax policy $q(\mathbf{s}|\mathbf{y})$ is introduced at the receiver side to enable exploration by sampling the objective function. In this setup, the decoder $q(\mathbf{s}|\mathbf{y})$ executes

the stochastic policy, whereas this task was handled by the encoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{s})$ in [BBD24]. It is also worth noting that both policies can be combined for greater flexibility.

Slow Training Convergence

Further, the problem of slow convergence becomes apparent from the numerical results on the CIFAR10 dataset from Chapter 5. We note that we use a higher exploration variance $\sigma^2_{\text{exp}} = 0.15$ compared to previous works [AH19a; LLC+22] with values of 0.0225 and 0.01, respectively, and larger batch sizes ($N_{\text{b}} = 512$ with SGD and $N_{\text{b}} = 500$ with Adam) to reduce estimator variance in our numerical experiments. However, further tuning of σ^2_{exp} is not expected to solve the slow convergence problem. Moreover, we used a relatively small number of channel uses per agent ($N_{\text{Tx}} = 16$), but among 4 agents, this number grows to 64, resulting in a large output space. Instead of tuning model parameters, we thus suggested exploring variance-reduction techniques in future work [GBB04; PBC+18; Sim18a; Li20].

Note that, analogous to the mean $\bar{\mathbf{x}} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$, also the exploration variance σ^2_{exp} can be parametrized by a DNN with parameters $\boldsymbol{\theta}_{\sigma_{\text{exp}}}$ to potentially accelerate convergence. There are two design options: Either one DNN with shared parameters $\boldsymbol{\theta}$ or two independent DNNs with independent parameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_{\bar{\mathbf{x}}}, \boldsymbol{\theta}_{\sigma_{\text{exp}}}]^T$ are used.

One specific idea to increase convergence with large output spaces is the DDPG [SLH+14]: Besides the deterministic policy or actor $\bar{\mathbf{x}} = \mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$, a DNN critic $Q(\mathbf{x})$ is introduced that approximates/estimates the cumulated reward function, e.g., the relation between source and target, since the true action-value function is not known/differentiable. In our scenario, this means we could estimate channel and receiver by training the critic $Q_{\boldsymbol{\varphi}}(\mathbf{x}) = -\ln q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{x})$ and optimize our actor or encoder $\mathbf{x} = \mu_{\boldsymbol{\theta}}(\mathbf{s})$ according to this estimate. This lowers variance similar to the reparametrization trick but requires that each time the receiver is updated, also the critic is updated [AH19a]. How much this scheme could increase or decrease convergence speed in our scenario is an open question also worth to be investigated in future work.

However, we note that the use of critics often implies a bias unless strict compatibility conditions are met [WRD+18]. These are rarely fulfilled in practice. In contrast, the SPG is unbiased but suffers from high variance. This translates into being less sample efficient but more stable.

B.6 Chapter Summary

In this chapter, we elaborated on key aspects of semantic communication more deeply. These include:

- We deepen the philosophical considerations about semantics and meaning by a more interdisciplinary view that highlights that meaning is closely related to organization, i.e., emergence, in the universe. Even here Shannon's information theory is applied for understanding. Further, we introduce semantic hierarchical levels as a replacement for Weaver's semantic and effectiveness level.

- We explain the difference between the system models of Chapter 4 and [BSW+23]. SINFONY transmits the semantics behind an observation explicitly defined. In [BSW+23], the messages in a Factor Graph (FG) are exchanged and to be transmitted one-to-one: With probabilistic models, we are able to include the uncertainty of the communication channel to provide a probabilistic estimate. This estimate closely connects to the message passing algorithm in a semantic sense making exploration communications-aware and communications exploration-aware.

- We extend the simulative comparison between semantic communication and a classic digital design based on the exemplary dataset of CIFAR10. The normalized SNR gap is even larger compared to the MNIST example. Most notably, SINFONY trained w.r.t. the communication channel can outperform SINFONY with an analog semantic AE for channel protection of its features. An overall semantic communication design proves itself to be more effective for more challenging datasets.

- We show performance results of alternative SINFONY designs justifying our default design choice. In particular, using a separate Rx module for each received signal, improves performance but only slightly.

- We reflect on why it is difficult to introduce semantics in a classic digital communication system and which efforts have already been done into this direction. Hard source decoding of VLC source codes increases latency and establishes an information barrier beyond which it is difficult to propagate soft information making a graceful degradation difficult. This leads to the key insight to remove the block-wise structure reflecting the semantics-tailored design, e.g., JSCC or SINFONY.

- We look at the numerical example from [BSW+23] from the different perspective of transmission of floating-point numbers. Based on a more general example, we show that the key insights of semantic performance improvements are valid.
- Lastly, we note that the SPG can also be used to train RL-SINFONY with a non-differentiable objective function measuring semantic similarity. Further, we elaborate on the problem of slow training convergence and present ideas to overcome this problem, e.g., parametrization of the exploration variance or introduction of the powerful DDPG approach [SLH+14].

Appendix C

Important Activation Functions

In this appendix, we complement Chapter 2 by the most important Deep Neural Network (DNN) activation functions and mathematical relations.

Softmax

For classification problems, the softmax function is a crucial final layer of a discriminative model since it provides probability outputs in the interval $[0, 1]$ [Sim18a]:

$$\sigma\left(\mathbf{x}\right) = \frac{e^{\mathbf{x}}}{\sum\limits_{k=1}^{M} e^{x_k}} \,. \tag{C.1}$$

Note that in the notation used in this thesis, the exponential function is applied element-wise. The softmax normalizes the sum of the outputs to 1 representing posterior probabilities for each class. In Chapter 3, it was used for a probabilistic continuous relaxation of discrete Random Variables (RVs) including the prior probabilities $\boldsymbol{\alpha}$ and the softmax temperature τ controlling the tightness of the relaxation:

$$\sigma_\tau\left(\mathbf{x}\right) = \frac{e^{(\ln(\boldsymbol{\alpha})+\mathbf{x})/\tau}}{\sum\limits_{k=1}^{M} e^{(\ln(\alpha_k)+x_k)/\tau}} \,. \tag{C.2}$$

Sigmoid

The sigmoid function is another activation function oftentimes used in Machine Learning (ML) literature. It converts the input to a real number within the interval $[0, 1]$ and can be seen as the special case of the softmax function when only two classes are present. Then, it outputs the probability of one of these classes:

$$\rho(x) = \frac{1}{1 + e^{-x}} \tag{C.3a}$$

$$= \frac{\tanh(x/2) + 1}{2} \tag{C.3b}$$

$$\rho'(x) = \rho(x) \cdot (1 - \rho(x)) \tag{C.3c}$$

$$= \rho(x) - \rho^2(x) \tag{C.3d}$$

$$= \frac{1}{4} \cdot (1 - \tanh^2(x/2)). \tag{C.3e}$$

The complimentary probability gives the probability for the other class. When driven into saturation in a DNN, it is known to decrease training speed [HZRS16a].

Tanh

A common alternative to the sigmoid function is the tanh function. Note that the output is nothing but a scaled and shifted sigmoid output:

$$f(x) = \tanh(x) \tag{C.4a}$$

$$= 2 \cdot \rho(2x) - 1 \tag{C.4b}$$

$$f'(x) = 1 - \tanh^2(x) \tag{C.4c}$$

$$= 2 \cdot \rho'(2x) \tag{C.4d}$$

$$f(-x)^2 = (-f(x))^2 \tag{C.4e}$$

$$= f(x)^2 . \tag{C.4f}$$

It is hence restricted to an output interval $[-1, 1]$ and basically shares the same properties as the sigmoid activation function.

ReLU

The Rectified Linear Unit (ReLU) is an activation function often encountered in ML literature and defined as:

$$\rho_{\text{relu}}(x) = \max(0, x) \tag{C.5a}$$

$$= \begin{cases} x, & x > 0 \\ 0, & x \leq 0\,. \end{cases} \tag{C.5b}$$

Its introduction is considered to be a breakthrough in optimization of DNNs: Before 2006, DNNs with, e.g., at that time standard sigmoid activation functions, were considered intractable to optimize until [HOT06] proposed an unsupervised pre-training procedure to make supervised training feasible. In 2011, it was shown in [GBB11] that using ReLU as a non-linearity enables fast supervised training of DNNs, avoiding the need for such unsupervised pre-training. Basically, ReLUs, combined with advances in dedicated training on Graphics Processing Units (GPUs), paved the way for recent breakthroughs in ML and lie at the core of nearly all DNN applications today. Benefits include sparse activations, better gradient propagation compared to the sigmoid function, efficient computation with addition and multiplication, and scale-invariance [HZRS16a]. Drawbacks include non-differentiability at zero, non-negativity, unboundedness and that some neural units may be pushed into inactive states.

ELU

Numerous advancements and modifications to the ReLU have been proposed since its introduction. One example is the Exponential Linear Unit (ELU) introduced in [CUH16], which allows negative values in contrast to the ReLU, and thus a small, positive gradient when the unit is not active. This modification allows pushing mean unit activations closer to zero and eventually for faster training and better generalization performance. It is defined as

$$\rho_{\text{elu}}(x) = \begin{cases} x, & x > 0 \\ a \cdot (e^x - 1), & x \leq 0\,, \end{cases} \tag{C.6}$$

where $a \geq 0$ is a hyperparameter to be tuned.

Acronyms

Adam . . . . . Adaptive Moment Estimation
ADF . . . . . Automatic Differentiation Framework
AE . . . . . AutoEncoder
AI . . . . . Artificial Intelligence
AMP . . . . . Approximate Message Passing
ASK . . . . . Amplitude Shift Keying
AWGN . . . . . Additive White Gaussian Noise

BER . . . . . Bit Error Rate
BERT . . . . . Bidirectional Encoder Representations from Transformers
BLEU . . . . . BiLingual Evaluation Understudy
BPSK . . . . . Binary Phase Shift Keying
BS . . . . . Base Station

CE . . . . . Cross-Entropy
CFER . . . . . Coded Frame Error Rate
ChatGPT . . . . . Chat Generative Pre-Trained Transformer
CMD . . . . . Concrete MAP Detection
CMDNet . . . . . Concrete MAP Detection Network
CMDpar . . . . . Parallel Concrete MAP Detection

CNN . . . . . Convolutional NN

DDPG . . . . . Deep Deterministic Policy Gradient
DetNet . . . . . Detection Network
dim. . . . . . dimension
DL . . . . . Deep Learning
DNN . . . . . Deep Neural Network
DSP . . . . . Digital Signal Processor

ELBO . . . . . Evidence Lower BOund
ELU . . . . . Exponential Linear Unit
EM . . . . . Expectation Maximization

FER . . . . . Frame Error Rate
FG . . . . . Factor Graph

GAN . . . . . Generative Adversarial Network
GCM . . . . . Generalized Context Model
GLM . . . . . Generalized Linear Model
GPU . . . . . Graphics Processing Unit

HDM . . . . . Human Decision-Making
HMI . . . . . Human-Machine Interface
HyperCMD . . . . Hypernetwork-based Concrete MAP detection
HyperMIMO . . . Hypernetwork-based MIMO Detection

I . . . . . Information
i.i.d. . . . . . independent and identically distributed
IB . . . . . Information Bottleneck
InfoMax . . . . . Information Maximization
IO . . . . . Individual Optimal

JSCC . . . . . Joint Source-Channel Coding

KL . . . . . Kullback–Leibler

LAMA . . . . . LArge MIMO Approximate message passing
LDPC . . . . . Low-Density Parity-Check

LLM . . . . . Large Language Model
LLR . . . . . Log-Likelihood Ratio
LOS . . . . . Line-Of-Sight
LS . . . . . Least Squares

M . . . . . Moment
MAC . . . . . Multiple Access Channel
MAP . . . . . Maximum A Posteriori
MaxL . . . . . Maximum Likelihood
MC . . . . . Monte Carlo
MF . . . . . Matched Filter
MFVI . . . . . Mean-Field Variational Inference
MI . . . . . Mutual Information
MILBO . . . . . MI Lower BOund
MIMO . . . . . Multiple Input Multiple Output
ML . . . . . Machine Learning
MM . . . . . Majorization Minimization
MMNet . . . . . Massive MIMO Network
MMSE . . . . . Minimum Mean Square Error
MOP . . . . . Multiplicative OPeration
MOSIC . . . . . MMSE Ordered Successive Interference Cancellation
MSE . . . . . Mean Square Error

NLP . . . . . Natural Language Processing
NMSE . . . . . Normalized MSE
NN . . . . . Neural Network
NP . . . . . Nondeterministic Polynomial time

OAMP . . . . . Orthogonal Approximate Message Passing
OAMPNet . . . . OAMP Network

pdf . . . . . probability density function
PIC . . . . . Parallel Interference Cancellation
pmf . . . . . probability mass function
PSK . . . . . Phase Shift Keying

QAM . . . . . Quadrature Amplitude Modulation
QPSK . . . . . Quadrature Phase Shift Keying
QRD . . . . . QR Decomposition

ReLU . . . . . Rectified Linear Unit
res. un. . . . . . residual unit
ResNet . . . . . Residual Network
RGB . . . . . Red Green Blue
RL . . . . . Reinforcement Learning
RL-SINFONY . . Reinforcement Learning-based SINFONY
RV . . . . . Random Variable
Rx . . . . . Receiver

s.t. . . . . . subject to
SD . . . . . Sphere Detector
SDR . . . . . SemiDefinite Relaxation
SDRadio . . . . Software Defined Radio
SER . . . . . Symbol Error Rate
SGD . . . . . Stochastic Gradient Descent
SIC . . . . . Successive Interference Cancellation
SINFONY . . . . Semantic INFOrmation TraNsmission and RecoverY
SNR . . . . . Signal-to-Noise Ratio
SotA . . . . . State of the Art
SPG . . . . . Stochastic Policy Gradient
SQRD . . . . . Sorted QR Decomposition

Tx . . . . . Transmitter

UE . . . . . User Equipment
ULA . . . . . Uniform Linear Array

VAE . . . . . Variational AutoEncoder
VI . . . . . Variational Inference
VLC . . . . . Variable-Length Codes

w.r.t. . . . . . with respect to

ZF . . . . . Zero Forcing

List of Symbols

Functions and Operators

$\lvert\cdot\rvert$	Absolute value
$[\cdot]$	Iverson bracket
$\lim_{\cdot\to\cdot} \cdot$	Limes
$\lVert\cdot\rVert$	Norm operator
$\prod_{\cdot}$	Product
$\sum_{\cdot}$	Summation
$\arg\max_{\cdot}$	Argument of the maximum
$\arg\min_{\cdot}$	Argument of the minimum
$D_f(\cdot \parallel \cdot)$	f-divergence
$\operatorname{diag}\{\cdot\}$	Diagonal matrix operator
$D_{\mathrm{KL}}(\cdot \parallel \cdot)$	KL divergence
$\mathrm{E}.[\cdot]$	Expectation of a random variable
$f(\cdot)$	General function
$I(\cdot)$	Mutual information
one-hot	One-hot function
$\operatorname{rank}(\cdot)$	Rank of a matrix
$\operatorname{sign}(\cdot)$	Sign function
$\operatorname{sim}(\cdot,\cdot)$	Similarity measure
$\operatorname{vec}(\cdot)$	Vectorization operator

General and Calligraphic Symbols

$\mathbf{0}$	Zero matrix

$\mathbf{1}$ . . . . . All-ones matrix, matrix of ones
$\mathcal{A}$ . . . . . Accuracy measure / Placeholder set
$\mathbb{C}$ . . . . . Complex-valued domain
$\mathcal{D}$ . . . . . Dataset, training set
$\mathcal{D}_{\text{Batch}}$ . . . . Mini-batch set
$\mathcal{D}_{\text{HK}}$ . . . . . Exemplar dataset or GCM knowledge base
$\mathcal{D}_{\text{P}}$ . . . . . Pilot set
$\mathcal{D}_{\text{T}}$ . . . . . Training set
$\mathcal{D}_{\text{Test}}$ . . . . . Test set
$\mathcal{D}_{\text{Val}}$ . . . . . Validation set
$\hat{\mathcal{H}}_{\cdot}(\cdot)$ . . . . . Empirical entropy
$\mathcal{H}(\cdot)$ . . . . . Entropy, cross-entropy
$\mathcal{L}$ . . . . . Loss function
$\mathcal{L}_{\boldsymbol{\theta}}^{\text{SPG}}$ . . . . . SPG objective function
$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\varphi}}^{\text{CE}}$ . . . . . Cross-entropy loss
$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\varphi}}^{\text{ELBO}}$ . . . . ELBO loss
$\mathcal{M}$ . . . . . Domain/Set
$\mathcal{N}$ . . . . . Normal/Gaussian distribution
$\mathcal{N}_{\mathbb{C}}$ . . . . . Circularly-symmetric Gaussian distribution
$\mathcal{O}(\cdot)$ . . . . . Big-O Operator
$\mathcal{Q}_{\cdot}\{\cdot\}$ . . . . . Quantizer function
$\mathbb{R}$ . . . . . Real-valued domain
$\mathcal{U}$ . . . . . Uniform distribution
$\mathbb{Z}$ . . . . . Domain of integers

Greek Symbols

$\boldsymbol{\alpha}$ . . . . . Prior probabilities of symbol vector
β . . . . . Lagrange multiplier
γ . . . . . GCM similarity decay constant
δ . . . . . Gradient step size/Dirac delta function
ϵ . . . . . Learning rate/Random angle deviation
ζ . . . . . Total path gain
η . . . . . Spectral efficiency
$\boldsymbol{\Theta}$ . . . . . Parameter matrix

$\boldsymbol{\theta}$	Parameter vector, encoder parameters
$\boldsymbol{\theta}_{\mathrm{P}}$	Presentation parameters vector
$\boldsymbol{\vartheta}$	Variational parameters vector
λ	Wavelength
μ	Mean
$\boldsymbol{\nu}$	Semantics presentation
$\rho(\cdot)$	Activation/Sigmoid function
$\rho_{\mathrm{relu}}(\cdot)$	ReLU function
$\boldsymbol{\Sigma}$	Covariance matrix
σ	Standard deviation
σ_{exp}	Exploration/perturbation standard deviation
σ_{n}	Noise standard deviation
$\sigma_{\tau}(\cdot)$	Softmax function (with optional softmax temperature)
σ_{φ}	Angular standard deviation
τ	Softmax temperature
ϕ	Input features of a GLM
$\tilde{\varphi}$	Angle of arrival of a single path
φ	Angle of user equipment
$\boldsymbol{\varphi}$	Variational distribution parameter vector
φ_{cell}	Cell sector
φ_{Δ}	Angular spread
$\boldsymbol{\varphi}_{\mathrm{G}}$	GCM parameter vector
$\boldsymbol{\varphi}_{\mathrm{nat}}$	Exponential family natural parameters
$\boldsymbol{\psi}$	Hyperparameter vector

Roman Symbols

$\mathbf{A}$	Placeholder matrix/Spatial sampling matrix
a	Placeholder variable
a	Placeholder variable realization
$\mathbf{a}$	Placeholder vector
$\boldsymbol{\mathsf{a}}$	Placeholder vector realization
B	Bandwidth
$\mathbf{b}$	Bit vector/Bias vector of neural network
C	Channel capacity

D . . . . . Distance in multiples of wavelength
$\mathbf{D}$ . . . . . Matrix with zero diagonal elements
d . . . . . Distance in m
$\mathbf{G}$ . . . . . Gumbel variables matrix
$\mathbf{g}$ . . . . . Multivariate Gumbel variable
$\mathbf{H}$ . . . . . Channel matrix
$\mathbf{I}$ . . . . . Identity matrix
I_{C} . . . . . Mutual information constraint
L . . . . . Objective function
M . . . . . Cardinality of discrete set, number of classes
$\mathbf{m}$ . . . . . Discrete set vector/Semantic variables in swarm exploration
N . . . . . Number of samples
$\mathbf{n}$ . . . . . Noise vector
N_{b} . . . . . Number of batches, batch size
N_{bit} . . . . . Number of bits of a floating point number
N_{c} . . . . . Number of image color dimensions
N_{e} . . . . . Number of training iterations/epochs
$N_{\mathrm{e,rx}}$ . . . . . Number of fine-tuning training iterations / receiver epochs
N_{Feat} . . . . . Number of features
N_{h} . . . . . Number of neurons in one layer / layer width
N_{it} . . . . . Number of iterations
N_{L} . . . . . Number of layers, DNN depth
N_{P} . . . . . Number of parallel CMD branches
N_{path} . . . . . Number of multipath components
N_{pilot} . . . . . Number of training examples / dataset size
N_{R} . . . . . Number of receive antennas
N_{Rx} . . . . . Channel output dimension
N_{T} . . . . . Number of transmit antennas
N_{train} . . . . . Number of training examples / dataset size
N_{Tx} . . . . . Transmitter dimension
N_{w} . . . . . Receiver dimension/layer width
N_{x} . . . . . Number of image pixels in x-dimension
N_{y} . . . . . Number of image pixels in y-dimension
p . . . . . Probability distribution/mass function

Q . . . . . DDPG critic
q . . . . . Approximating distribution
$\mathbf{r}$ . . . . . Features
R_{C} . . . . . Channel coding rate
$\mathbf{R_h}$ . . . . . Correlation matrix of channel response
R_{S} . . . . . Source coding rate
$\mathbf{s}$ . . . . . Sensed observation/Source/Logistic/Exploration RVs
$\mathbf{W}$ . . . . . Weight matrix
$\mathbf{w}$ . . . . . Weights
$\hat{\mathbf{x}}$ . . . . . Estimated symbols
$\tilde{\mathbf{x}}$ . . . . . Symbol function
$\bar{\mathbf{x}}$ . . . . . Encoder/Transmit output with exploration noise
$\mathbf{x}$ . . . . . Transmit symbol/Target variable vector
$\tilde{\mathbf{y}}$ . . . . . Total observation vector
$\mathbf{y}$ . . . . . Observation/Receive vector
$\tilde{\mathbf{z}}$ . . . . . Semantic estimate (after semantic communication decision)
$\hat{\mathbf{z}}$ . . . . . Semantic estimate (after human decision)
$\tilde{\mathbf{z}}$ . . . . . Concrete variable vector
$\mathbf{z}$ . . . . . Semantic random variable/One-hot vector

Bibliography

[AAB+15] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from `https://www.tensorflow.org/`, Nov. 2015. DOI: `10.5281/zenodo.4724125`.

[AH19a] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019. DOI: `10.1109/JSAC.2019.2933891`.

[AH19b] F. A. Aoudia and J. Hoydis, "Towards Hardware Implementation of Neural Network-based Communication Algorithms," in *20th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2019)*, Cannes, France, Jul. 2019, pp. 1–5. DOI: `10.1109/SPAWC.2019.8815398`.

[APF+18] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy, "Fixing a Broken ELBO," in *International Conference on Machine Learning (ICML 2018)*, vol. 35, Stockholm, Sweden, Jul. 2018, pp. 159–168.

[AZ21] I. E. Aguerri and A. Zaidi, "Distributed Variational Representation Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 1, pp. 120–138, Jan. 2021. DOI: `10.1109/TPAMI.2019.2928806`.

[BB12] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *Journal of Machine Learning Research*, vol. 13, no. 10, pp. 281–305, Feb. 2012.

[BB23] C. M. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*. Cham, Switzerland: Springer, Nov. 2023. DOI: 10.1007/978-3-031-45468-4.

[BBD+11a] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a theory of semantic communication," in *IEEE Network Science Workshop (NSW 2011)*, West Point, NY, USA, Jun. 2011, pp. 110–117. DOI: 10.1109/NSW.2011.6004632.

[BBD+11b] J. Bao, P. Basu, M. Dean, C. Partridge, A. Swami, W. Leland, and J. A. Hendler, "Towards a Theory of Semantic Communication (Extended Technical Report)," Rensselaer Polytechnic Institute, Troy, NY, USA, Tech. Rep. ADA544137, Mar. 2011, Available at: https://apps.dtic.mil/sti/citations/ADA544137.

[BBD17] C. Bockelmann, E. Beck, and A. Dekorsy, "One- and Two-dimensional Compressive Edge Spectrum Sensing," in *8th Jahreskolloquium Kommunikation in der Automation (KommA 2017)*, Magdeburg, Germany, Nov. 2017.

[BBD18] E. Beck, C. Bockelmann, and A. Dekorsy, "Compressed Edge Spectrum Sensing for Wideband Cognitive Radios," in *26th European Signal Processing Conference (EUSIPCO 2018)*, Rome, Italy, Sep. 2018, pp. 1705–1709. DOI: 10.23919/EUSIPCO.2018.8553617.

[BBD19] E. Beck, C. Bockelmann, and A. Dekorsy, "Compressed Edge Spectrum Sensing: Extensions and Practical Considerations," *at - Automatisierungstechnik*, Methods for a reliable wireless communication in the industry, vol. 67, no. 1, pp. 51–59, Jan. 2019. DOI: 10.1515/auto-2018-0059.

[BBD20] E. Beck, C. Bockelmann, and A. Dekorsy, "Concrete MAP Detection: A Machine Learning Inspired Relaxation," in *24th International ITG Workshop on Smart Antennas (WSA 2020)*, Hamburg, Germany: VDE VERLAG, Feb. 2020, pp. 1–5.

[BBD21] E. Beck, C. Bockelmann, and A. Dekorsy, "CMDNet: Learning a Probabilistic Relaxation of Discrete Variables for Soft Detection With Low Complexity," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8214–8227, Dec. 2021. DOI: 10.1109/TCOMM.2021.3114682.

[BBD22] E. Beck, C. Bockelmann, and A. Dekorsy, *Semantic Communication: An Information Bottleneck View*, arXiv preprint: 2204.13366v1 (Version 1), 2022. DOI: 10.48550/arXiv.2204.13366.

[BBD23] E. Beck, C. Bockelmann, and A. Dekorsy, "Semantic Information Recovery in Wireless Networks," *Sensors*, vol. 23, no. 14, p. 6347, Jul. 2023. DOI: 10.3390/s23146347.

[BBD24] E. Beck, C. Bockelmann, and A. Dekorsy, "Model-free Reinforcement Learning of Semantic Communication by Stochastic Policy Gradient," in *1st IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN 2024)*, Stockholm, Sweden, May 2024, pp. 367–373. DOI: 10.1109/ICMLCN59089.2024.10625190.

[BBDH14] P. Basu, J. Bao, M. Dean, and J. Hendler, "Preserving Quality of Information by Using Semantic Relationships," *Pervasive and Mobile Computing*, vol. 11, pp. 188–202, Apr. 2014. DOI: 10.1016/j.pmcj.2013.07.013.

[Bec17] E. Beck, "Compressed Spectrum Sensing for Cognitive Radio in Time and Space," de, M.S. thesis, University of Bremen, Bremen, Germany, May 2017.

[Bec23] E. Beck, *Concrete MAP Detection Network (CMDNet) Software*, Zenodo, version v1.0.2, Oct. 2023. DOI: 10.5281/zenodo.8416507.

[Bec24] E. Beck, *Semantic Information Transmission and Recovery (SINFONY) Software*, Zenodo, version v1.2.2, Dec. 2024. DOI: 10.5281/zenodo.8006567.

[BHS17] E. Björnson, J. Hoydis, and L. Sanguinetti, "Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, Nov. 2017. DOI: 10.1561/2000000093.

[BHV+16] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *30th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, Dec. 2016, pp. 523–531.

[Bis06] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Springer, Jan. 2006.

[BKG19] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep Joint Source-Channel Coding for Wireless Image Transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, Sep. 2019. DOI: 10.1109/TCCN.2019.2919300.

[BLR+25] E. Beck, H.-Y. Lin, P. Rückert, Y. Bao, B. von Helversen, S. Fehrler, K. Tracht, and A. Dekorsy, *Integrating Semantic Communication and Human Decision-Making into an End-to-End Sensing-Decision Framework*, arXiv preprint: 2412.05103, Mar. 2025. DOI: 10.48550/arXiv.2412.05103.

[Bri08] S. Brier, *Cybersemiotics: Why Information is Not Enough!* University of Toronto Press, Jan. 2008. DOI: 10.3138/9781442687813.

[Bri13] S. Brier, "Cybersemiotics: A new foundation for transdisciplinary theory of information, cognition, meaningful communication and the interaction between nature and culture," *Integral Review*, vol. 9, no. 2, pp. 220–263, Jun. 2013.

[BS19] A. Balatsoukas-Stimming and C. Studer, "Deep Unfolding for Communications Systems: A Survey and Some New Directions," in *IEEE International Workshop on Signal Processing Systems (SiPS 2019)*, Nanjing, China, Oct. 2019, pp. 266–271. DOI: 10.1109/SiPS47522.2019.9020494.

[BSW+23] E. Beck, B.-S. Shin, S. Wang, T. Wiedemann, D. Shutin, and A. Dekorsy, "Swarm Exploration and Communications: A First Step towards Mutually-Aware Integration by Probabilistic Learning," *Electronics*, Swarm Communication, Localization and Navigation, vol. 12, no. 8, p. 1908, Apr. 2023. DOI: 10.3390/electronics12081908.

[BV04] S. P. Boyd and L. Vandenberghe, *Convex optimization*, 29th ed. Cambridge, United Kingdom: Cambridge University Press, Mar. 8, 2004, 716 pp. DOI: 10.1017/CBO9780511804441.

[CAD+20] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. t. Brink, "Trainable Communication Systems: Concepts and Prototype," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5489–5503, Sep. 2020. DOI: 10.1109/TCOMM.2020.3002915.

[CB20] A. Caciularu and D. Burshtein, "Unsupervised Linear and Nonlinear Channel Equalization and Decoding Using Variational Autoencoders," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 1003–1018, Sep. 2020. DOI: 10.1109/TCCN.2020.2990773.

[CB52] R. Carnap and Y. Bar-Hillel, "An Outline of a Theory of Semantic Information," Research Laboratory of Electronics, Massachusetts Institute of Technology, Technical Report 247, Oct. 1952, p. 54.

[CGGS13] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks," in *16th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI 2013)*, Nagoya, Japan: Springer, Sep. 2013, pp. 411–418. DOI: 10.1007/978-3-642-40763-5_51.

[CGWW09] M. A. Chappell, A. R. Groves, B. Whitcher, and M. W. Woolrich, "Variational Bayesian Inference for a Nonlinear Forward Model," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 223–236, Jan. 2009. DOI: 10.1109/TSP.2008.2005752.

[Cho+15] F. Chollet *et al.*, *Keras*, https://keras.io, Mar. 2015.

[CMS12] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in *25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, Providence, RI, USA, Jun. 2012, pp. 3642–3649. DOI: 10.1109/CVPR.2012.6248110.

[CRBD18] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," in *31st Conference on Advances in Neural Information Processing Systems (NIPS 2018)*, Montreal, Canada, Dec. 2018, pp. 6571–6583.

[CT06] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd. Hoboken, NJ, USA: Wiley-Interscience, Jul. 2006. DOI: 10.1002/047174882X.

[CUH16] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations (ICLR 2016)*, vol. 4, San Juan, Puerto Rico, May 2016, pp. 1–14.

[FG18] N. Farsad and A. Goldsmith, "Neural Network Detection of Data Sequences in Communication Systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018. DOI: 10.1109/TSP.2018.2868322.

[Flo09] L. Floridi, "Philosophical Conceptions of Information," in *Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information*, ser. Lecture Notes in Computer Science, Apr. 2009, pp. 13–53. DOI: 10.1007/978-3-642-00659-3_2.

[FRG18] N. Farsad, M. Rao, and A. Goldsmith, "Deep Learning for Joint Source-Channel Coding of Text," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, Calgary, Canada, Apr. 2018, pp. 2326–2330. DOI: 10.1109/ICASSP.2018.8461983.

[GAH20] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep HyperNetwork-Based MIMO Detection," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2020)*, vol. 21, Atlanta, GA, USA, May 2020, pp. 1–5. DOI: 10.1109/SPAWC48557.2020.9154283.

[GB10] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, vol. 13, Chia (Sardinia), Italy: JMLR Workshop and Conference Proceedings, May 2010, pp. 249–256.

[GBB04] E. Greensmith, P. L. Bartlett, and J. Baxter, "Variance Reduction Techniques for Gradient Estimates in Reinforcement Learning," *Journal on Machine Learning Research*, vol. 5, pp. 1471–1530, Dec. 2004.

[GBB11] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, Ft. Lauderdale, FL, USA: JMLR Workshop and Conference Proceedings, Jun. 2011, pp. 315–323.

[GBBD22a] S. Gracla, E. Beck, C. Bockelmann, and A. Dekorsy, "Deep Reinforcement Model Selection for Communications Resource Allocation in On-Site Medical Care," in *IEEE Wireless Communications and Networking Conference (WCNC 2022)*, Austin, TX, USA, Apr. 2022, pp. 1485–1490. DOI: 10.1109/WCNC51071.2022.9771679.

[GBBD22b] S. Gracla, E. Beck, C. Bockelmann, and A. Dekorsy, "Learning Resource Scheduling with High Priority Users using Deep Deterministic Policy Gradients," in *IEEE International Conference on Communications (ICC 2022)*, Seoul, South Korea, May 2022, pp. 4480–4485. DOI: 10.1109/ICC45855.2022.9838349.

[GBBD23] S. Gracla, E. Beck, C. Bockelmann, and A. Dekorsy, "Robust Deep Reinforcement Learning Scheduling via Weight Anchoring," *IEEE Communications Letters*, vol. 27, no. 1, pp. 210–213, Jan. 2023. DOI: 10.1109/LCOMM.2022.3214574.

[GP20] Z. Goldfeld and Y. Polyanskiy, "The Information Bottleneck Problem and its Applications in Machine Learning," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 19–38, May 2020. DOI: 10.1109/JSAIT.2020.2991561.

[GQA+23] D. Gündüz, Z. Qin, I. E. Aguerri, H. S. Dhillon, Z. Yang, A. Yener, K. K. Wong, and C.-B. Chae, "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 5–41, Jan. 2023. DOI: 10.1109/JSAC.2022.3223408.

[Gra20] S. Gracla, "Resource Allocation using Deep Learning," en, M.S. thesis, University of Bremen, Bremen, Germany, Oct. 2020.

[GRV03] M. Gastpar, B. Rimoldi, and M. Vetterli, "To Code, or Not to Code: Lossy Source-Channel Communication Revisited," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1147–1158, May 2003. DOI: 10.1109/TIT.2003.810631.

[Hae21] S. Haesloop, "Quantization of Deep MIMO Detectors," en, M.S. thesis, University of Bremen, Bremen, Germany, Jul. 2021.

[Has22] S. Hassanpour, "Source & Joint Source-Channel Coding Schemes Based on the Information Bottleneck Framework," Ph.D. dissertation, University of Bremen, Bremen, Germany, Aug. 2022, p. 196.

[HB23] M. Hummert and E. Beck. "KI in der Kommunikationstechnik." de-DE. *Industrial Radio Lab Germany Blog*. (Feb. 2023), [Online]. Available: https://industrial-radio-lab.eu/2023/02/23/ki-in-der-kommunikationstechnik/.

[HBD24] A. Halimi Razlighi, C. Bockelmann, and A. Dekorsy, "Semantic Communication for Cooperative Multi-Task Processing Over Wireless Networks," *IEEE Wireless Communications Letters*, vol. 13, no. 10, pp. 2867–2871, Oct. 2024. DOI: 10.1109/LWC.2024.3451139.

[HDL16] D. Ha, A. Dai, and Q. V. Le, *HyperNetworks*, arXiv preprint: 1609.09106, Dec. 2016. DOI: 10.48550/arXiv.1609.09106.

[Hes25] T. Heskes, *Bias-variance decompositions: The exclusive privilege of bregman divergences*, arXiv preprint: 2501.18581, Jan. 2025. DOI: 10.48550/arXiv.2501.18581.

[Hof13] W. Hofkirchner, *Emergent Information: A Unified Theory of Information Framework*. World Scientific: Singapore, Dec. 2013, vol. 3. DOI: 10.1142/7805.

[HOT06] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006. DOI: 10.1162/neco.2006.18.7.1527.

[HRW14] J. R. Hershey, J. L. Roux, and F. Weninger, *Deep Unfolding: Model-based Inspiration of Novel Deep Architectures*, arXiv preprint: 1409.2574, Sep. 2014. DOI: 10.48550/arXiv.1409.2574.

[HSW89] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, Jan. 1989. DOI: 10.1016/0893-6080(89)90020-8.

[HTB+25] A. Halimi Razlighi, M. H. V. Tillmann, E. Beck, C. Bockelmann, and A. Dekorsy, "Cooperative and Collaborative Multi-Task Semantic Communication for Distributed Sources," in *IEEE International Conference on Communications (ICC 2025)*, Montreal, Canada, Jun. 2025, pp. 1–6. DOI: 10.48550/arXiv.2411.02150.

[HWG+25] X. Han, Y. Wu, Z. Gao, B. Feng, Y. Shi, D. Gündüz, and W. Zhang, "SCSC: A Novel Standards-Compatible Semantic Communication Framework for Image Transmission," *IEEE Transactions on Communications*, pp. 1–17, Jan. 2025. DOI: 10.1109/TCOMM.2025.3529221.

[HZRS15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *IEEE International Conference on Computer Vision (ICCV 2015)*, vol. 14, Santiago, Chile, Dec. 2015, pp. 1026–1034. DOI: 10.1109/ICCV.2015.123.

[HZRS16a] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[HZRS16b] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *14th European Conference on Computer Vision (ECCV 2016)*, ser. Lecture Notes in Computer Science, Amsterdam, Netherlands, Oct. 2016, pp. 630–645. DOI: 10.1007/978-3-319-46493-0_38.

[IEE19] IEEE, "IEEE Standard for Floating-Point Arithmetic," *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, pp. 1–84, Jul. 2019. DOI: 10.1109/IEEESTD.2019.8766229.

[JGMS15] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of Large MIMO Detection via Approximate Message Passing," in *IEEE International Symposium on Information Theory (ISIT 2015)*, Hong Kong, Jun. 2015, pp. 1227–1231. DOI: 10.1109/ISIT.2015.7282651.

[JGMS18] C. Jeon, R. Ghods, A. Maleki, and C. Studer, *Optimal Data Detection in Large MIMO*, arXiv preprint: 1811.01917, Nov. 2018. DOI: 10.48550/arXiv.1811.01917.

[JGP17] E. Jang, S. Gu, and B. Poole, "Categorical Reparameterization with Gumbel-Softmax," in *5th International Conference on Learning Representations (ICLR 2017)*, Toulan, France, Apr. 2017, pp. 1–13. DOI: 10.48550/arXiv.1611.01144.

[KAHF20] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive Neural Signal Detection for Massive MIMO," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020. DOI: 10.1109/TWC.2020.2996144.

[KCT+18] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-End Deep Learning of Optical Fiber Communications," *IEEE/OSA Journal of Lightwave Technology*, vol. 36, no. 20, pp. 4843–4855, Oct. 2018. DOI: 10.1109/JLT.2018.2865109.

[Koe68] A. Koestler, *The Ghost in the Machine*. New York, NY, USA: Macmillan, 1968, p. 384.

[KSH12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *25th Conference on Advances in Neural Information Processing Systems (NIPS 2012)*, Lake Tahoe, Nevada, USA, Dec. 2012, pp. 1097–1105. DOI: 10.1145/3065386.

[Li20] Y. Li, *Topics in Approximate Inference*, Manuscript available at http://yingzhenli.net/home/pdf/topics_approx_infer.pdf, Dec. 2020.

[LL09] D. D. Lin and T. J. Lim, "A Variational Inference Framework for Soft-In Soft-Out Detection in Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2345–2364, May 2009. DOI: 10.1109/TIT.2009.2016054.

[LLC+22] K. Lu, R. Li, X. Chen, Z. Zhao, and H. Zhang, *Reinforcement Learning-powered Semantic Communication via Semantic Similarity*, arXiv preprint: 2108.12121, Apr. 2022. DOI: 10.48550/arXiv.2108.12121.

[LWZ+21] Q. Lan, D. Wen, Z. Zhang, Q. Zeng, X. Chen, P. Popovski, and K. Huang, "What is Semantic Communication? A View on Conveying Meaning in the Era of Machine Intelligence," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, Dec. 2021. DOI: 10.23919/JCIN.2021.9663101.

[MLE21] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, Mar. 2021. DOI: 10.1109/MSP.2020.3016905.

[MMT17] C. J. Maddison, A. Mnih, and Y. W. Teh, "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables," in *5th International Conference on Learning Representations (ICLR 2017)*, Toulan, France, Apr. 2017, pp. 1–20. DOI: 10.48550/arXiv.1611.00712.

[Nos84] R. M. Nosofsky, "Choice, similarity, and the context theory of classification," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, no. 1, p. 104, Jan. 1984. DOI: 10.1037/0278-7393.10.1.104.

[OH17] 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, Dec. 2017. DOI: 10.1109/TCCN.2017.2758370.

[PB15] P. Počta and J. G. Beerends, "Subjective and Objective Assessment of Perceived Audio Quality of Current Digital Audio Broadcasting Systems and Web-Casting Applications," *IEEE Transactions on Broadcasting*, vol. 61, no. 3, pp. 407–415, Sep. 2015. DOI: 10.1109/TBC.2015.2424373.

[PBC+18] M. Papini, D. Binaghi, G. Canonaco, M. Pirotta, and M. Restelli, "Stochastic Variance-Reduced Policy Gradient," in *International Conference on Machine Learning (ICML 2018)*, vol. 35, Stockholm, Sweden, Jul. 2018, pp. 4026–4035.

[PGM+19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *32nd Conference on Advances in Neural Information Processing Systems (NIPS 2019)*, Vancouver, Canada, Dec. 2019, pp. 8024–8035.

[PSB+20] P. Popovski, O. Simeone, F. Boccardi, D. Gündüz, and O. Sahin, "Semantic-Effectiveness Filtering and Control for Post-5G Wireless Connectivity," *Journal of the Indian Institute of Science*, vol. 100, no. 2, pp. 435–443, Apr. 2020. DOI: 10.1007/s41745-020-00165-6.

[Rao19] S. S. Rao, *Engineering Optimization: Theory and Practice*. Hoboken, NJ, USA: John Wiley & Sons, Nov. 2019. DOI: 10.1002/9781119454816.

[RGL+24] F. E. Rosas, B. C. Geiger, A. I. Luppi, A. K. Seth, D. Polani, M. Gastpar, and P. A. M. Mediano, *Software in the natural world: A computational approach to hierarchical emergence*, arXiv preprint: 2402.09090, Jun. 2024. DOI: 10.48550/arXiv.2402.09090.

[SAH19] M. Stark, F. A. Aoudia, and J. Hoydis, "Joint Learning of Geometric and Probabilistic Constellation Shaping," in *IEEE GLOBECOM Workshops (GC Wkshps 2019)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6. DOI: 10.1109/GCWkshps45667.2019.9024567.

[SB21] E. C. Strinati and S. Barbarossa, "6G networks: Beyond Shannon towards semantic and goal-oriented communications," *Computer Networks*, vol. 190, p. 107 930, May 2021. DOI: 10.1016/j.comnet.2021.107930.

[SBD+24] A. Suresh, E. Beck, A. Dekorsy, P. Rückert, and K. Tracht, "Human-integrated Multi-agent Exploration using Semantic Communication and Extended Reality Simulation," in *10th IEEE International Conference on Automation, Robotics and Applications (ICARA 2024)*, Athens, Greece, May 2024, pp. 419–426. DOI: 10.1109/ICARA60736.2024.10553050.

[SDW17] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO Detection," in *18th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2017)*, Sapporo, Japan, Jul. 2017, pp. 1–5. DOI: 10.1109/SPAWC.2017.8227772.

[SDW19] N. Samuel, T. Diskin, and A. Wiesel, "Learning to Detect," *IEEE Transactions on Signal Processing*, vol. 67, no. 10, pp. 2554–2564, May 2019. DOI: 10.1109/TSP.2019.2899805.

[Sha48] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[SHM+16] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. DOI: 10.1038/nature16961.

[Sim18a] O. Simeone, "A Brief Introduction to Machine Learning for Engineers," *Foundations and Trends® in Signal Processing*, vol. 12, no. 3-4, pp. 200–431, Aug. 2018. DOI: 10.1561/2000000102.

[Sim18b] O. Simeone, "A Very Brief Introduction to Machine Learning with Applications to Communication Systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, Dec. 2018. DOI: 10.1109/TCCN.2018.2881442.

[Sim22] O. Simeone, *Machine Learning for Engineers*. Cambridge, United Kingdom: Cambridge University Press, Nov. 3, 2022, 450 pp. DOI: 10.1017/9781009072205.

[SJRvH24] F. I. Seitz, J. B. Jarecki, J. Rieskamp, and B. von Helversen, "Disentangling Perceptual and Process-Related Sources of Behavioral Variability in Categorization," *PsyArXiv*, Sep. 2024. DOI: 10.31234/osf.io/g3bpa.

[SLA+19] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Dahl, "Measuring the Effects of Data Parallelism on Neural Network Training," *Journal of Machine Learning Research*, vol. 20, no. 112, pp. 1–49, Jul. 2019.

[SLH+14] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *31st International Conference on Machine Learning (ICML 2014)*, Beijing, China, Jan. 2014, pp. 387–395.

[SMZ22] J. Shao, Y. Mao, and J. Zhang, "Learning Task-Oriented Communication for Edge Inference: An Information Bottleneck Approach," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 197–211, Jan. 2022. DOI: 10.1109/JSAC.2021.3126087.

[SMZ23] J. Shao, Y. Mao, and J. Zhang, "Task-Oriented Communication for Multidevice Cooperative Edge Inference," *IEEE Transactions on Wireless Communications*, vol. 22, no. 1, pp. 73–87, Jan. 2023. DOI: 10.1109/TWC.2022.3191118.

[ST17] R. Shwartz-Ziv and N. Tishby, *Opening the Black Box of Deep Neural Networks via Information*, arXiv preprint: 1703.00810, Mar. 2017. DOI: 10.48550/arXiv.1703.00810.

[SW49] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, 16th ed. Urbana, IL, USA: The University of Illinois Press, Sep. 1949.

[TPB99] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *37th Annual Allerton Conference on Communication, Control and Computing (Allerton 1999)*, Sep. 1999, pp. 368–377. DOI: 10.48550/arXiv.physics/0004057.

[TZ15] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *IEEE Information Theory Workshop (ITW 2015)*, Jerusalem, Israel, Apr. 2015, pp. 1–5. DOI: 10.1109/ITW.2015.7133169.

[UKE+22] E. Uysal, O. Kaya, A. Ephremides, J. Gross, M. Codreanu, P. Popovski, M. Assaad, G. Liva, A. Munari, B. Soret, T. Soleymani, and K. H. Johansson, "Semantic Communications in Networked Systems: A Data Significance Perspective," *IEEE/ACM Transactions on Networking*, vol. 36, no. 4, pp. 233–240, Jul. 2022. DOI: 10.1109/MNET.106.2100636.

[VSP+17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *31st Conference on Advances in Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, Dec. 2017, pp. 6000–6010.

[Wea49] W. Weaver, "Recent Contributions to the Mathematical Theory of Communication," in *The Mathematical Theory of Communication*, Urbana, IL, USA: The University of Illinois Press, 1949, pp. 261–281.

[WRD+18] C. Wu, A. Rajeswaran, Y. Duan, V. Kumar, A. M. Bayen, S. Kakade, I. Mordatch, and P. Abbeel, "Variance Reduction for Policy Gradient with Action-Dependent Factorized Baselines," in *International Conference on Learning Representations (ICLR 2018)*, vol. 6, Vancouver, Canada, Apr. 2018.

[WRS+17] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The Marginal Value of Adaptive Gradient Methods in Machine Learning," in *31st Conference on Advances in Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, Dec. 2017, pp. 4148–4158.

[WWW+17] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep Learning for Wireless Physical Layer: Opportunities and Challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, Nov. 2017. DOI: 10.1109/CC.2017.8233654.

[XQLJ21] 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, Apr. 2021. DOI: 10.1109/TSP.2021.3071210.

[XYN+23] W. Xu, Z. Yang, D. W. K. Ng, M. Levorato, Y. C. Eldar, and M. Debbah, "Edge Learning for B5G Networks With Distributed Signal Processing: Semantic Communication, Edge Computing, and Wireless Sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 17, no. 1, pp. 9–39, Jan. 2023. DOI: 10.1109/JSTSP.2023.3239189.

[ZES20] A. Zaidi, I. Estella-Aguerri, and S. Shamai Shitz, "On the Information Bottleneck Problems: Models, Connections, Applications and Information Theoretic Views," *Entropy*, vol. 22, no. 2, p. 151, Feb. 2020. DOI: 10.3390/e22020151.

[ZPZ+12] A. Zribi, R. Pyndiah, S. Zaibi, F. Guilloud, and A. Bouallegue, "Low-Complexity Soft Decoding of Huffman Codes and Iterative Joint Source Channel Decoding," *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1669–1679, Jun. 2012. DOI: 10.1109/TCOMM.2012.041212.100330.

[ZSBL19] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-Shot Adversarial Learning of Realistic Neural Talking Head Models," in *IEEE/CVF International Conference on Computer Vision (ICCV 2019)*, vol. 17, Seoul, South Korea, Oct. 2019, pp. 9458–9467. DOI: 10.1109/ICCV.2019.00955.

Index