UNIVERSITY OF CALIFORNIA SAN DIEGO

Channel Estimation and Data Detection Methods for 1-bit Massive MIMO Systems

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

David Kin Wai Ho

Committee in charge:

Professor Bhaskar D. Rao, Chair
Professor Patrick J. Fitzsimmons
Professor Kenneth Kreutz-Delgado
Professor Laurence B. Milstein
Professor Xinyu Zhang

2022

The dissertation of David Kin Wai Ho is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my wife Vivian

and my daughters Elise and Danielle

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First I would like to thank my advisor, Prof. Bhaskar D. Rao, for accepting me into his lab, for his patience with my progress as a part time student and for his guidance throughout the doctorate program. I thank him deeply for introducing me to the field of Bayesian analysis and encouraging me to explore the exciting field of deep learning, both of which are extensively applied throughout this dissertation.

I like to thank Prof. Laurence B. Milstein for his advise during my master's program and for his encouragement when I inquired about pursing a doctorate degree.

I am honored to have Prof. Patrick J. Fitzsimmons, Prof. Kenneth Kreutz-Delgado, and Prof. Xinyu Zhang serve on my committee.

I am very thankful to have worked with Prof. Max Welling who introduced me to his research on graph neural networks. Our many discussions on model based approach and mean field inference was instrumental to the successful development of a structured deep learning detector.

I like to thank PhuongBang Nguyen for the numerous discussions we had. It was he who inspired me to take on this journey, as few would have done. He was always available when I needed someone to share my thoughts. Thanks to all the wonderful people in the DSP lab, I had numerous fruitful discussions and valued the ideas they shared with me. They made my extended stay enjoyable.

Finally, I would like to thank my wife Vivian for her love and unwavering support, without whom I will not have succeeded in this audacious journey. I am also blessed to have two lovely daughters Elise and Danielle during this period. I dedicate this work to them.

Chapter 2, in part, is a reprint of the material as it appears in the paper: D. K. W. Ho and B. D. Rao, "Antithetic dithered 1-bit massive MIMO architecture: Efficient channel estimation via parameter expansion and PML", *IEEE Transactions on Signal*

Processing, vol. 67, no. 9, pp.2291-2303, May 2019. The dissertation author was the primary investigator and author of this paper.

Chapter 3, in part, is a reprint of unpublished material as it appears in the technical report: "Performance limit of feed-forward type neural 1-bit receivers". The dissertation author was the primary investigator and author of this report.

Chapter 4, in part, is a reprint of material that is currently being prepared for submission for publication to be identified as: D. K. W. Ho, M. Welling, B. D. Rao, "Structured Neural 1-bit Massive MIMO Detector Based on Stochastic Variational Inference". The dissertation author was the primary investigator and author of this material.

VITA

2001	Bachelor of Applied Science, University of Waterloo, Canada
2010	Master of Science, University of California San Diego
2022	Doctor of Philosophy, University of California San Diego

PUBLICATIONS

D. K. W. Ho, M. Welling and B. D. Rao, "Structured Neural 1-bit Massive MIMO Detector Based on Stochastic Variational Inference," in preparation for submission.

D. K. W. Ho and B. D. Rao, "One-bit Massive MIMO Detector Based on Variational Methods," submitted to *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

D. K. W. Ho and B. D. Rao, "Antithetic dithered 1-bit massive MIMO architecture: Efficient channel estimation via parameter expansion and PML", *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp.2291-2303, May 2019.

ABSTRACT OF THE DISSERTATION

Channel Estimation and Data Detection Methods for 1-bit Massive MIMO Systems

by

David Kin Wai Ho

Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California San Diego, 2022

Professor Bhaskar D. Rao, Chair

Massive multiple-input multiple-output (MIMO) is a promising technology for next generation communication systems. In massive MIMO, a base station (BS) is equipped with a large antenna with potentially hundreds of antennas elements, allowing many users to be served simultaneously. Unfortunately, the hardware complexity and power consumption will scale with the number of antennas. The use of one-bit analog-to-digital converters (ADCs) provides an attractive solution to solve the above issues, since a one-bit ADC consumes negligible power and complex automatic gain control (AGC) can be removed.

However, the signal distortion from the severe quantization poses significant challenges to the system designer. One bit quantization effectively removes all amplitude information, which is not recoverable by an increase in signal strength. This places a bound on channel estimation performance. Since the channel model is highly nonlinear, linear detector is suboptimal compared to more sophisticated nonlinear techniques.

To reduce the impairment caused by one-bit quantization, a novel antithetic dithering scheme is developed. Antithetic dither is introduced into the system to generate negative correlated noise. Efficient channel estimation algorithms are developed to exploit the induced negative correlated noise in the system. A statistical framework is developed to validate the noise reduction from negative correlated quantized output.

To improve the performance of data detection, feed forward neural network based detectors are developed, performance of these detectors are analyzed, architectural modification and training techniques are employed to partially resolve issues that prevent the networks from reaching ideal maximum likelihood performance.

Next, model based approaches are evaluated and the shortcomings of iterative methods that rely on the exact likelihood are identified. Iterative methods based on the exact likelihood is shown to diverge due to the increasingly large gradient at high SNR. The constant gradient induced by the sigmoid approximation is shown to increase the robustness of these methods. A structured deep learning detector based on stochastic variational inference is proposed. Stochastic estimate of the gradient is introduced to reduce complexity of the algorithm. Damping is added to improve the performance of mean field inference. Parallel processing is proposed to reduce inference time. The proposed detector is shown to outperform existing methods that do not employ a second candidate search step.

Chapter 1

Introduction

Massive multiple-input multiple-out (MIMO) is a breakthrough concept that promises to deliver the throughput, spectral and energy efficiency required in next generation wireless systems. In massive MIMO, a base station (BS) is equipped with a large antenna array with potentially hundreds of antenna elements. A massive MIMO system is depicted in Fig. 1.1. The large array provides an excess degrees of freedom that allows many user equipments (UEs) to be served in the same time-frequency resource. The signal can be narrowly focused through the use of multi-user beamforming, providing significant improvement in energy efficiency. The narrower beams also translate to less inter-user interference, thus the system is robust against both unintended interference and intentional jamming. In a massive MIMO system, the channel responses from the UEs are nearly orthogonal, maximum ratio combining (MRC) can be used in place of other computationally demanding techniques. Other benefits include the ability to use inexpensive low-power components, reduced signaling and latency by leveraging channel reciprocity in a time-division duplex (TDD) configuration and the removal of fast fading from channel hardening [1, 2].

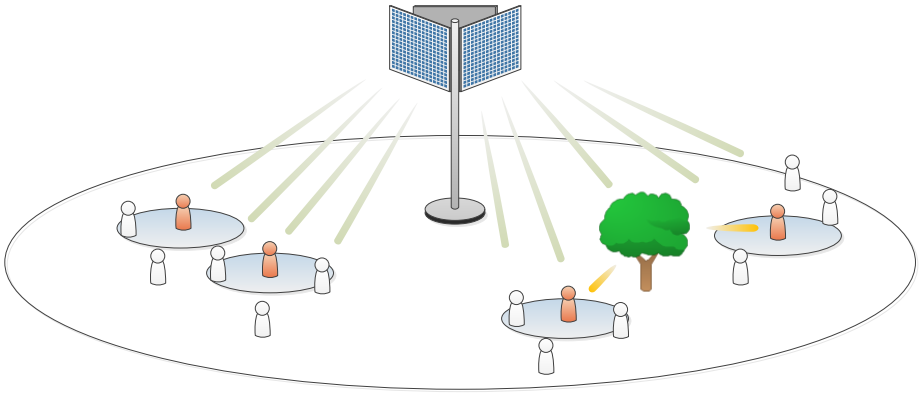


Figure 1.1: A Massive MIMO system

1.1 One-bit massive MIMO

In a conventional MIMO system, a fully digital architecture is employed where each antenna is connected to the digital interface via a dedicated RF chain. A high-resolution analog-to-digital converter (ADC) is used to provide a high-precision re-construction of the signal from analog domain for digital baseband processing. In massive MIMO system, the hardware complexity and power consumption will scale with the number of antennas. In millimeter wave (mmWave) systems with large bandwidth, the sampling rate of the ADCs must also scale with it. Unfortunately, high-speed, high-resolution ADCs are expensive and power hungry [3–5]. As an example, a flash ADC with b-bit resolution implements $2^b - 1$ comparators in parallel, the power consumption grows exponentially with resolution. The need to use a large number of these ADCs means power consumption can be excessive.

An alternative is to consider reducing sampling rate. This can be accomplished by employing multiple lower speed ADCs in a time-interleaved (TI) configuration. However, mismatch among the ADCs in gain, timing and voltage offset can cause an error floor in receiver performance. The increase in hardware complexity and the lack of clear power advantage make this choice unappealing. Another alternative is to reduce the resolution of the ADCs to ultra low resolutions, down to one bit.

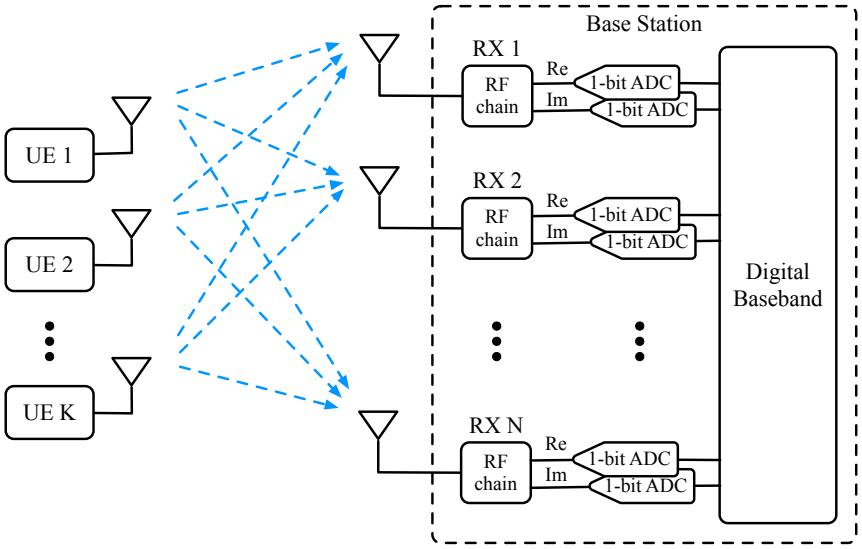


Figure 1.2: Model of a one-bit massive MIMO system.

The use of one-bit ADCs is particularly attractive because it can substantially reduce power consumption and lower hardware complexity. A one-bit ADC can implemented as a simple comparator which consumes negligible power [6]. With one-bit ADCs the overall receiver architecture can also be simplified, because automatic gain control (AGC) is not needed [7]. A one-bit massive MIMO system is illustrated in Fig. 1.2

The theoretical bound of one-bit massive MIMO has been studied in [8–12]. One bit quantization reduced the low SNR channel capacity by only a factor of $2/\pi$ (-1.96 dB) [8]. In one bit massive MIMO systems, high-order constellations can be supported [11] and it can attain a remarkable 70% of the spectral efficiency of a full-precision system [12].

However, the use of one-bit ADCs poses significant challenges to the system designer. One major drawback is the signal distortion from taking only the sign of the real and imaginary components of the received signal. The severe quantization effectively removes all amplitude information, which is not recoverable by an increase in signal strength. This places a bound on channel estimation performance.

One can reduce the quantization effect through the use of dithering. However, dither introduces noise into the system and leads to a reduction in signal to noise ratio

(SNR). In chapter 2, negatively correlated dither is exploited to partially cancel the dither noise. As a result the system can operate in a region with reduce compression effect from quantization.

In data detection, since the channel model is highly nonlinear, linear filtering is suboptimal compared to more sophisticated nonlinear techniques. The recent development in deep learning creates a new avenue for exploration of novel solutions. First, a neural network is known to be a universal approximator [13, 14]. A deep neural network can be used to learn the nonlinear relationship completely from data. This technique imposes no additional bias to the network and assumes all pertinent channel model details can be learned from data. At the other extreme, a neural network can be built with full model knowledge, it is then trained to optimize parameters within the model. The next section will introduce deep learning and elaborate on the two options.

1.2 Deep learning

Deep neural networks (DNNs) aim to remove many model assumptions inherit in traditional approaches by learning the model directly from observations. Since a DNN can in theory approximate any nonlinear function asymptotically [14–16], even at a finite limit, we expect the class of functions that a DNN can represent to be significantly larger than many tractable models, such as the exponential model typically employed in variational inference methods. Recent efforts had removed many of the problems associated with neural network training. Practitioners can routinely train deep nets hundreds of layers deep with relative ease [17–22]; the well-known problem of gradient vanishing and explosion in training deep feed-forward networks has largely been settled with modern training techniques [23–28]. The significant success of deep nets is also a result of favorable local minima found in these models [29–33]. Another truly remarkable property of

deep neural networks and SGD training, and more generally over-parameterized models, is implicit generalization; which has been recently elucidated in [34–38]. As a result of the DNN model's expressiveness, we can expand the scope of the inference problem to include all nonlinearity experienced by a receiver, such as amplifier nonlinearity, component mismatch, high-order statistical coupling (beyond cross-correlation), and the effect of quantization.

DNNs are not immune to practical challenges. Large amount of data is required to train a DNN because of it flexibility and its lack of bias. Large DNN may have hundreds of millions of parameters. We note that many published results lean heavily towards over-parameterized models, chiefly because it eliminates dependence on initialization that has plagued early efforts in the field [39]. Because the architectures of DNNs are generic, it is not straightforward to incorporate domain knowledge. See the following introductory texts for more information [40, 41].

Examples of DNN-based approach applied to communications include: autoencoder based end-to-end communication systems [42–45], channel estimation [46], channel estimation and DOA estimation [47], and soft-demapper [48].

In model-based approach, domain knowledge is incorporated at the outset, this leads to an optimization problem. In many interesting cases, direct optimization is intractable, and one must resort to an iterative inference algorithm. The inference iterations are unfolded to form layers in a deep network. The model parameters are trained via supervised learning. There are far fewer parameters in these models and training can be performed using fewer data samples. Because of the strong bias inherited in these model, they are also less likely to converge to a poor local minimum. Effectively, an inference algorithm is optimized to produce the best result in a fixed number of layers.

We provide a general formulation of algorithm unfolding that is applied to the detection problem. Consider a model which is fully specified with parameters $\boldsymbol{\theta}$, these

parameters determine the relationship between hidden variables $\mathbf{z}_i$ and observable values $\mathbf{x}_i$, additionally they may include the parameters of the inference algorithm. For a MIMO system, the parameters may include the channel $\mathbf{H}$ and noise variance σ^2. If gradient descent is used, the parameters may include the step size $\boldsymbol{\alpha}$. The hidden variable $\mathbf{z}_i$ might be the transmitted symbol in a hard detection problem, or it might be the log likelihood ratio of the transmitted bits in a soft detection problem. At test time, the hidden variable are determined by optimizing an objective function $\mathcal{F}(\mathbf{x}_i, \boldsymbol{\phi}_i)$,

$$\widehat{\boldsymbol{\phi}}_i = \arg\min_{\boldsymbol{\phi}_i} \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{x}_i, \boldsymbol{\phi}_i), \quad \mathbf{z}_i = g(\mathbf{x}_i, \widehat{\boldsymbol{\phi}}_i) \tag{1.1}$$

where $\boldsymbol{\phi}_i$ are intermediate quantities from which $\mathbf{z}$ are computed and g is the estimator of $\mathbf{z}_i$. In variational inference $\boldsymbol{\phi}_i$ are the variational parameters of the approximate distribution. The minimization problem is assumed to be solved using an iterative algorithm, each iteration is determined by

$$\boldsymbol{\phi}_i^{(t)} = f_{\boldsymbol{\theta}}(\mathbf{x}_i, \boldsymbol{\phi}_i^{(t-1)}) \tag{1.2}$$

where $t \in 1, \ldots, T$ and $\boldsymbol{\phi}_i^{(0)}$ is the initial estimate. We note that the function $f_{\boldsymbol{\theta}}$ may be composed of many smaller updates. For example, in variational inference, variational parameters are updated separately. At training time, the parameters $\boldsymbol{\theta}$ are learned using the objective function

$$\mathcal{E} = \sum_i \mathcal{L}(\mathbf{z}_i^*, \widehat{\mathbf{z}}_i(\mathbf{x}_i|\boldsymbol{\theta})) \tag{1.3}$$

where $\mathcal{L}$ is a loss function and $\mathbf{z}_i^*$ is the ground truth. The algorithm is then unfolded where each iteration forms a layer in an architecture that resembles a deep neural network. The output of each layer corresponds to the output of each iteration. The output of the final layer is determined by $\mathbf{z}_i^{(T)} = g(\mathbf{x}_i, \boldsymbol{\phi}_i^{(T)})$. In some instances, the model can be made more

expressive by untying the parameters between iterations. In other cases, the inference parameters that ordinarily are determined via cross validation can now be trained using standard deep learning techniques. In chapter 4, the model-based approach is applied specifically to mean field inference. See [49] for application of the model-based approach to belief propagation and non-negative matrix factorization.

Of all the deep learning detection scheme reported in the literature, the majority are model-based methods. Examples of model-based detection schemes are: DetNet [50], OAMPNet [51], MMNet [52] and RE-MIMO [53] and OBMNet [54].

1.3 Dissertation overview

The dissertation is organized as follows:

In chapter 2, we propose a novel antithetic dithered 1-bit receiver architecture to reduce signal distortion. Efficient channel estimation algorithms are developed to exploit the induced negative correlated noise for improved estimation performance. We illustrate that both linear and nonlinear estimators can benefit from negative correlation. We provide a rigorous analysis of a low complexity nonlinear estimator for channel estimation. In the process, we develop a generalized statistical framework to analyze correlated quantized output arising from this generalized linear model. We formalize the approximation technique used in this work as a special case of the more general pseudo maximum likelihood method. A parameter expanded EM (PX-EM) algorithm applied to such a system is shown to exhibit fast convergence, possessing an upper bounded convergence guarantee and a graceful monotonic estimation performance over a large SNR range. Stochastic Gibbs sampling algorithms are constructed to evaluate truncated multivariate normal distributions and to implement an asymptotically exact data augmentation algorithm for comparison.

In chapter 3, we study the feasibility of using feed forward networks to learn the nonlinear relationship of the 1-bit MIMO model purely from data. We develop feed forward neural network (FFNN) based soft-detectors and provide an analysis of the performance these detectors, address and partially resolve the issues that prevent the networks from reaching the ideal maximum likelihood (ML) performance limit. Next we turn to the simpler symbol detection problem and assess the performance issues by analyzing the performance of several state of the art FFNN architectures. We demonstrate that the performance limit cannot be overcome by modern deep learning architectures and training techniques, and provide possible explanations to this behavior. We conjecture that the extreme nonlinearity of the likelihood function makes the task of learning difficult, sufficient inductive bias must be given to the neural network to reduce the difficulty of learning, and direct mappings from data to approximate posterior estimates may be inferior to architectures based on iterative methods.

In chapter 4, we evaluate existing model based approaches, identify the shortcomings of gradient methods that rely on the exact likelihood and determine the attributes that make the sigmoid approximation robust in the high SNR regime. We focus on a set of algorithms based on variational methods and propose a structured deep learning detector based on stochastic variational inference. Stochastic estimate of the mean field update is introduced to reduce complexity of the algorithm. Damping is added to further improve the performance of mean field inference (MFI). Parallel processing is proposed to reduce inference time. The proposed PSMFNet contains few parameters and can be trained efficiently using standard deep learning techniques. In numerical experiments, the proposed detector is shown to outperform existing methods that do not employ a second candidate search step.

Chapter 2

Antithetic Dithered 1-bit Massive MIMO Architecture

Massive multiple-input multiple-output (MIMO) systems with 1-bit analog-to-digital converters (ADCs) is a promising solution to control hardware complexity and lower power consumption. However, due to the severe distortion of received signal from the quantizer, it sets a limit to the accuracy of channel estimation. Drawing from recent work on negative noise correlation in quantization and statistics, a novel antithetic dithered 1-bit massive MIMO recievier architecture is proposed to reduce signal distortion. Antithetic dither is introduced to generate negative correlated noise across a pair of ADCs. Efficient channel estimation algorithms are developed to exploit the induced negative correlated noise in the system. We illustrate that both linear and nonlinear estimators can benefit from negative correlation. We provide a rigorous analysis of a low complexity nonlinear estimator for channel estimation. In the process, we developed a generalized statistical framework to analyze correlated quantized output arising from this generalized linear model. We formalized the approximation technique used in this work as a special case of the more general pseudo maximum likelihood method. A parameter

expanded EM (PX-EM) algorithm applied to such a system is shown to exhibit fast convergence, possessing an upper bounded convergence guarantee and a graceful monotonic estimation performance over a large SNR range. Stochastic Gibbs sampling algorithms are constructed to evaluate truncated multivariate normal distributions and to implement an asymptotically exact data augmentation algorithm for comparison.

2.1 Introduction

In a noise limited massive MIMO environment with fully connected digital baseband, 1-bit analog-to-digital converter (ADC) is especially appealing because of the simplicity of design and the low power consumption it offers. Theoretical analysis on achievable rates were studied as early as 2007 by [8, 55]. Since then various detection algorithms have been proposed, such as near ML detector in [56], GAMP estimator [57] and linear estimators [11, 12]. In the quantizer literature, dithering is a well known method used to condition the output of audio requantizers [58] and delta-sigma modulators/converters [59]. Recent works have noted that uniform independent quantization noise model [60] does not apply in general. They focused their analysis on error correlation to develop new methods to minimize the effect of correlated noise. For example, negative correlation is exploited to improve the performance of parallel ADCs [61] and time-interleaved ADCs [62]. In a parallel development in statistics, techniques have been developed to construct exchangeable pairs with the desired negative correlation to reduce variance over naive estimates [63].

It has been observed that for iid symmetrically distributed pilots $\{a_i\} \sim p(a)$, the asymptotic transformation of the unknown channel x_k via the passage of 1-bit ADCs [64] has a behavior reminiscent to the effect of compression due to a nonlinear power amplifier (PA); with worsening compression as $\sqrt{\text{SNR}}$ increases. Dithering leads to a reduction in SNR and shifts the signal towards the linear region, albeit at the expense of increasing the

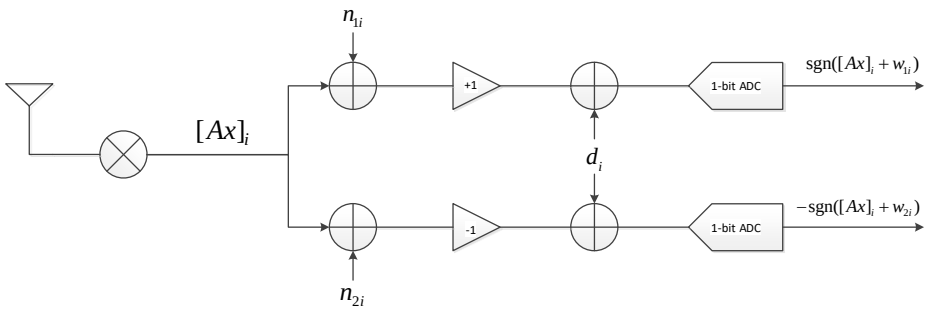


Figure 2.1: Topology of an antithetic dithered 1-bit receiver

overall noise power. One can perform subtractive dithering. If Schuchman's condition [65] is satisfied, subtractive dithering minimizes the noise contribution from any such dither. However the requirement to remove the dither signal exactly at the quantizer output makes it difficult to implement in practice.

Inspired by the development in dithering and the use of negative correlation, we propose a novel antithetic dithered architecture that has the dither canceling property of subtractive dithering without a need for exact knowledge of the dither signal. Unlike non-dithered 1-bit systems, whose symbol detection and channel estimation performance deteriorate at high SNR, such a system permits efficient and monotonically increasing NMSE performance over a wide SNR region to be achieved with little loss from dither induced noise. As a result the amplifier stages and automatic gain control (AGC) in a complex massive MIMO receiver can be substantially simplified by the flatter performance curve. A realizable topology is shown in Figure 2.1. We note here that a conventional 2-bit ADC increases the dynamic range of the receiver, although in practice complex AGC algorithms and gain stages are required to realize its full potential [66].

We highlight in this work that both linear and nonlinear estimators can benefit from variance reduction by exploiting negative correlation in the antithetic dither pair. In addition, we propose a fast parameter expanded EM (PX-EM) algorithm that is well

suited for the 1-bit ADC model with the ability to achieve the efficiency of a pseudo maximum likelihood estimator (PMLE). We briefly discuss the potential symbol estimation improvement a linear symbol estimator can realize at high SNR and proceed to focus our analysis on nonlinear estimators for channel estimation in this work. The contributions of this paper are summarized here:

- We proposed introducing negative correlated noise via antithetic dithering to provide monotonically increasing channel estimation performance and extend the usable range of 1-bit ADC systems beyond low SNRs.
- We illustrated that both linear and nonlinear estimators can benefit from negative correlation. For nonlinear estimators, we provided extensive theoretical and empirical analysis to support the claim.
- Efficient channel estimation algorithms are developed using pseudo-maximum likelihood method and the parameter expansion technique.

The rest of the paper is organized as follows. Section 2.2 introduces the system model under which the theoretical analysis and algorithm development are considered. Section 2.3 provides the motivation for using antithetic dithering and demonstrates the effect when the technique is used with linear estimators. Section 2.4 provides analysis of the performance improvement in nonlinear estimators via the classical statistical framework, introduces the concept of pseudo maximum likelihood method and includes a theoretical justification for its use. Section 2.5 analyzes the structure of the log-likelihood from an optimization standpoint and develops the various algorithms considered in this work including a thorough procedure for deriving the PX-EM. Section 2.6 compares the performance of various topologies and algorithms via numerical simulation. Section 2.7 ends this paper with concluding statements.

Here we provide a summary of uncommon notations. Let $\mathbb{1}_A(x)$ be the indicator function that takes on the value 1 if $x \in A$ and 0 if $x \notin A$. Let $(\cdot)^\mathsf{T}$ be the matrix transpose. $\mathbf{1}$ is a vector of all ones. $[a; b]$ denotes a column vector formed by stacking the elements vertically from top to bottom. Let $[\mathbf{A}]_{ij}$ be the (i, j)th-element of matrix $\mathbf{A}$. $\mathsf{D}^2_{\mathbf{x}}$ denotes the Hessian operator wrt $\mathbf{x}$ equivalent to $\frac{\partial^2}{\partial \mathbf{x} \partial \mathbf{x}^\mathsf{T}}$. Let $\varphi(x), \Phi(x)$ be the standard normal density and cumulative distribution function on the space of reals.

2.2 System model

In a receiver with 1-bit quantizers, the received signal for a narrowband MIMO system at antenna n_r, from K' single antenna transmitters and N' pilot symbols has the form

$$\mathbf{r}_{n_r} = \mathcal{Q}'(\mathbf{A}_c \mathbf{x}_{n_r} + \mathbf{w}_{n_r}) \tag{2.1}$$

where $\mathbf{A}_c \in \mathbb{C}^{N' \times K'}$ is the pilot symbol matrix, $\mathbf{x}_{n_r} \in \mathbb{C}^{K'}$ is the channel vector to be estimated corrupted by iid zero mean complex Gaussian noise vector $\mathbf{w}_{n_r}$, the quantized output $\mathbf{r}_{n_r} \in \mathcal{X}^{N'} \subset \mathbb{C}^{N'}$ consists of a vector of alphabets generated elementwise by the function

$$\mathcal{Q}'(z) = \operatorname{sgn}(\operatorname{Re} z) + j \operatorname{sgn}(\operatorname{Im} z), \quad z \in \mathbb{C} \tag{2.2}$$

where

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{if } x > 0 \\ -1, & \text{if } x \leq 0 \end{cases} \tag{2.3}$$

with $x \in \mathbb{R}$. Without loss of generality we transform the problem into a set of $N = 2N'$ real nonlinear system of equations

$$\mathbf{r} = \mathcal{Q}(\mathbf{A}\mathbf{x} + \mathbf{w}) \tag{2.4}$$

and study algorithms based on this equivalent form. Here $\mathcal{Q}(\cdot) = \text{sgn}(\cdot)$, with N, K denoting the corresponding dimensions of the real valued system. We comment here that the induced matrix $\mathbf{A}$ does not have the degrees of freedom of a fully independent matrix, but this does not pose any difficulty in the topic considered.

2.2.1 Antithetic dither model

We introduce at the input of two 1-bit ADCs, an antithetic dither pair $(\mathbf{d}, -\mathbf{d})$ to obtain two correlated observations $(\mathbf{r}_1, \mathbf{r}_2)$

$$\begin{aligned} \mathbf{r}_1 &= \mathcal{Q}(\mathbf{A}\mathbf{x} + \mathbf{n}_1 + \mathbf{d}) \\ \mathbf{r}_2 &= \mathcal{Q}(\mathbf{A}\mathbf{x} + \mathbf{n}_2 - \mathbf{d}) \end{aligned} \tag{2.5}$$

where $\mathbf{n}_i \sim \mathcal{N}(0, \sigma_n^2\mathbf{I})$, $\mathbf{d} \sim \mathcal{N}(0, \sigma_d^2\mathbf{I})$. The noise components are not necessarily independent i.e. the covariance of $(\mathbf{n}_1, \mathbf{n}_2)$ takes the form

$$\begin{bmatrix} \mathbf{\Gamma} & \mathbf{\Omega} \\ \mathbf{\Omega}^\mathsf{T} & \mathbf{\Gamma} \end{bmatrix}, \quad \mathbf{\Gamma} \triangleq \sigma_n^2\mathbf{I}. \tag{2.6}$$

Nonetheless, the noise $(\mathbf{n}_1, \mathbf{n}_2)$ and dither $\mathbf{d}$ are independent. The parameters $\mathbf{A}$, σ_d^2 are known.

In the theoretical analysis, we made a simplifying assumption that the noise along the two signal branches $\mathbf{n}_1, \mathbf{n}_2$ are independent to make the analysis tractable. We note that

in typical modem implementation, the noise vectors $\mathbf{n}_1, \mathbf{n}_2$ are highly positive correlated. Nonetheless, this does not impact the conditioning at high SNR and dither canceling property associated with antithetic dithering, which are the main contributions of this paper. We will however return to a more realistic case where $\mathbf{n}_1 = \mathbf{n}_2$ in the numerical experiments to demonstrate the proposed algorithms' effectiveness in this condition.

We can consider the Gaussian vectors $\mathbf{w}_i$ that combine the random components leading to

$$\begin{aligned} \mathbf{r}_1 &= \mathcal{Q}(\mathbf{A}\mathbf{x} + \mathbf{w}_1) \\ \mathbf{r}_2 &= \mathcal{Q}(\mathbf{A}\mathbf{x} + \mathbf{w}_2). \end{aligned} \tag{2.7}$$

All subsequent analysis will be based on this model.

2.3 Motivation and the effect on linear estimators

First we motivate the approach by presenting a linear example. Consider two observations of $\mathbf{x}$ via the Gaussian linear model

$$\mathbf{A}\mathbf{x} + \mathbf{w}_1 = \mathbf{b} \tag{2.8}$$

$$\mathbf{B}\mathbf{x} + \mathbf{w}_2 = \mathbf{c}. \tag{2.9}$$

Suppose $\mathbf{w}_1$ and $\mathbf{w}_2$ have zero mean, unit variance and negative correlation. In many situations the two observations can be highly correlated or the covariance information is simply not available such that an optimal estimator cannot be constructed. Given this constraint, one can dispense with the covariance information and apply the method of

least squares (LS), define

$$\mathbf{D} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix}. \tag{2.10}$$

The LS estimate has the form

$$\widehat{\mathbf{x}}_{\mathrm{LS}} = (\mathbf{D}^{\mathsf{T}}\mathbf{D})^{-1}\mathbf{D}^{\mathsf{T}}\mathbf{p} \tag{2.11}$$

$$= (\mathbf{A}^{\mathsf{T}}\mathbf{A} + \mathbf{B}^{\mathsf{T}}\mathbf{B})^{-1}\mathbf{A}^{\mathsf{T}}\mathbf{b} + (\mathbf{A}^{\mathsf{T}}\mathbf{A} + \mathbf{B}^{\mathsf{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathsf{T}}\mathbf{c} \tag{2.12}$$

$$= \mathbf{F}\mathbf{b} + \mathbf{G}\mathbf{c} \tag{2.13}$$

with variance

$$\mathrm{Cov}(\widehat{\mathbf{x}}_{\mathrm{LS}}) = \mathbf{F}\boldsymbol{\Sigma}_{bb}\mathbf{F}^{\mathsf{T}} + \mathbf{G}\boldsymbol{\Sigma}_{cc}\mathbf{G}^{\mathsf{T}} + \mathbf{F}\boldsymbol{\Sigma}_{bc}\mathbf{G}^{\mathsf{T}} + \mathbf{G}\boldsymbol{\Sigma}_{cb}\mathbf{F}^{\mathsf{T}} \tag{2.14}$$

$$= \mathbf{F}\boldsymbol{\Sigma}_{bb}\mathbf{F}^{\mathsf{T}} + \mathbf{G}\boldsymbol{\Sigma}_{cc}\mathbf{G}^{\mathsf{T}} + \mathbf{S}_{bc} \tag{2.15}$$

where $\boldsymbol{\Sigma}_{xy} = \mathrm{Cov}(\mathbf{x}, \mathbf{y})$, $\mathrm{Cov}(\mathbf{x}, \mathbf{y})$ is the cross covariance matrix of $\mathbf{x}, \mathbf{y}$ and $\mathrm{Cov}(\mathbf{x}) = \mathrm{Cov}(\mathbf{x}, \mathbf{x})$. For the special case where $\mathbf{A} = \mathbf{B}$, hence $\mathbf{F} = \mathbf{G}$, we have the condition, if $\boldsymbol{\Sigma}_{bc}$ is symmetric, negative semi-definite, then $\mathbf{S}_{bc}$ is also symmetric, negative semi-definite. Therefore

$$\mathrm{Cov}(\widehat{\mathbf{x}}_{\mathrm{LS}}) = \mathbf{F}\boldsymbol{\Sigma}_{bb}\mathbf{F}^{\mathsf{T}} + \mathbf{G}\boldsymbol{\Sigma}_{cc}\mathbf{G}^{\mathsf{T}} + \mathbf{S}_{bc} \tag{2.16}$$

$$\leq \mathbf{F}\boldsymbol{\Sigma}_{bb}\mathbf{F}^{\mathsf{T}} + \mathbf{G}\boldsymbol{\Sigma}_{cc}\mathbf{G}^{\mathsf{T}} \tag{2.17}$$

where (2.17) is the variance of the estimator with uncorrelated $\mathbf{b}, \mathbf{c}$. It can easily be seen that the inequality is reversed if $\boldsymbol{\Sigma}_{bc}$ is symmetric, positive semi-definite, i.e. when the noise vectors $\mathbf{w}_1, \mathbf{w}_2$ are positively correlated. We next consider one such scenario we call

the antithetic form

$$\mathbf{A}\mathbf{x} + \mathbf{w} = \mathbf{b} \tag{2.18}$$

$$\mathbf{A}\mathbf{x} - \mathbf{w} = \mathbf{c} \tag{2.19}$$

where $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I})$. The utility of this idea becomes apparent when the outputs of the system pass through a nonlinear monotonic function g, i.e.

$$\begin{aligned} \mathbf{r}_1 &= g(\mathbf{A}\mathbf{x} + \mathbf{w}) \\ \mathbf{r}_2 &= g(\mathbf{A}\mathbf{x} - \mathbf{w}). \end{aligned} \tag{2.20}$$

If the negative semi-definiteness of the covariance matrix is preserved i.e. $\mathrm{Cov}(\mathbf{r}_1, \mathbf{r}_2) \leq \mathbf{0}$, then it can be shown that a linear unbiased estimator of $\mathbf{x}$ will have reduced variance compared to the case where $\mathbf{r}_1, \mathbf{r}_2$ are uncorrelated. Note that monotonic functions satisfy this property as a consequence of Chebyshev's covariance inequality [67]. Hence this technique can be applied to a large class of nonlinear functions. Beyond linear estimators, this phenomenon applies more generally to nonlinear estimators, such as maximum likelihood estimators (MLEs). We provide a brief non-rigorous discussion on linear estimators to highlight the potential symbol estimation improvement that can be achieved by applying antithetic dithering in this scenario before focusing exclusively on nonlinear estimators in the subsequent sections.

2.3.1 Linear estimators

For any symmetrically distributed $\mathbf{w}$, we see from (2.20) that the marginal distribution of $\mathbf{r}_1$ and $\mathbf{r}_2$ are the same. Thus if an unbiased estimator $\mathbf{G}(\cdot)$ exists for one it is

an unbiased estimator for both, i.e. $\mathbb{E}[\mathbf{G}(\mathbf{r}_1)] = \mathbb{E}[\mathbf{G}(\mathbf{r}_2)] = \mathbf{x}$. Then

$$\mathbf{G}'(\mathbf{r}_1, \mathbf{r}_2) = \frac{1}{2}[\mathbf{G}(\mathbf{r}_1) + \mathbf{G}(\mathbf{r}_2)] \tag{2.21}$$

is also unbiased. If $\mathbf{G}$ is linear, the above reduces to

$$\mathbf{G}'(\mathbf{r}_1, \mathbf{r}_2) = \mathbf{G}\Big[\frac{1}{2}(\mathbf{r}_1 + \mathbf{r}_2)\Big]. \tag{2.22}$$

Suppose the cross covariance matrix $\mathbf{\Sigma}_{r_1 r_2}$ of $\mathbf{r}_1, \mathbf{r}_2$ is symmetric, denote the covariance matrix of $\mathbf{r}_i$, $\mathbf{\Sigma}_{r_i} \triangleq \mathbf{\Sigma}_r$, then the covariance of $\mathbf{G}'$ is

$$\mathbf{\Sigma}(\mathbf{G}') = \frac{1}{4}[2\mathbf{G}\mathbf{\Sigma}_r\mathbf{G}^{\mathsf{T}} + 2\mathbf{G}\mathbf{\Sigma}_{r_1 r_2}\mathbf{G}^{\mathsf{T}}]. \tag{2.23}$$

The degree of variance reduction depends on the relative negative correlation between $\mathbf{r}_1$ and $\mathbf{r}_2$. As with conventional dithered 1-bit systems, the same linear estimator can be used to realize significant variance reduction.

We illustrate the improvement realizable in estimating the symbols of an uplink 1-bit massive MIMO receiver with linear estimator, full channel knowledge and antithetic dithering in Figure 2.2. We limit the scope of this work to non-linear estimators and provide this result only to illustrate the benefit of antithetic dithering for the linear case. We note that at 10dB SNR, the system performs poorly without dithering hence the non-dithered result is not shown. Observe that distortion to the constellation is minimized given sufficient richness of the channel in the frequency domain compared to a narrowband system; the detailed analysis of the behavior in wideband massive MIMO system can be found in [64]. Hence antithetic dithering is the dominant factor in reducing noise in the received symbols in the class of 1-bit systems.

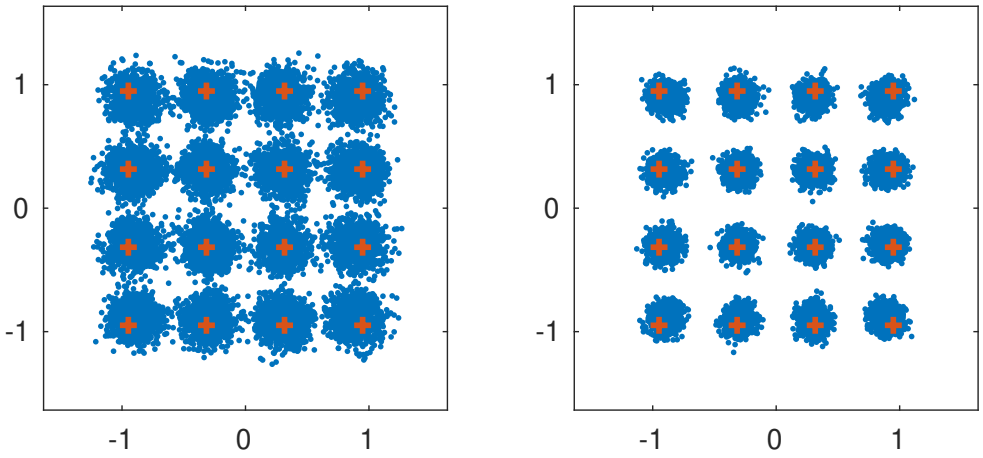


Figure 2.2: Qualitative comparison of linear MMSE equalized single carrier 16-QAM symbols over frequency selective channel using (left) normal dither and (right) antithetic dither with signal to dither power ratio = 1, $N_{rx} = 128$, $N_{tx} = 2$, SNR = 10dB

2.4 Analysis of ML estimators

We are interested in understanding whether the same behavior exists in the more general class of nonlinear estimators. In this work, we restrict our focus to ML estimators where the tools for analysis are quite mature. We begin this endeavor by presenting an alternate grouping such that an independent observation pair $\mathbf{r}_{*j}$ using identical design vector $\mathbf{a}_j$ has the form

$$\mathbf{r}_{*j} = \mathcal{Q}(\underbrace{\mathbf{A}_{*j}\mathbf{x} + \mathbf{w}_{*j}}_{\mathbf{y}_{*j}}), \quad j \in \{1, \ldots, N\} \tag{2.24}$$

where $\mathbf{r}_{*j} = [r_{1j}, r_{2j}]^\mathsf{T}$ and $\mathbf{y}_{*j} = [y_{1j}, y_{2j}]^\mathsf{T}$, $\mathbf{A}_{*j} = [\mathbf{a}_j \ \mathbf{a}_j]^\mathsf{T}$, $\mathbf{w}_{*j} = [w_{1j}, w_{2j}]^\mathsf{T} \sim \mathcal{N}(0, \mathbf{\Sigma})$. As we shall see this is a more natural representation for the subsequent analysis. The Fisher information matrix (FIM) is obtained by summing the FIM of the independent groups, i.e.

$$\boldsymbol{J}_N(\mathbf{x}) = \sum_j \boldsymbol{J}_j(\mathbf{x}). \tag{2.25}$$

We note that for the asymptotic result to hold the variables $(\mathbf{r}_{*j}, \mathbf{A}_{*j})$ are assumed to be sampled from a common joint distribution $p(\mathbf{r}, \mathbf{A})$ such that as $N \to \infty$ we have the asymptotic problem

$$\max_{\mathbf{x}} \frac{1}{N} \sum_{j} \Psi(\mathbf{r}_{*j}, \mathbf{A}_{*j}; \mathbf{x}) \xrightarrow{\text{a.s.}} \max_{\mathbf{x}} \mathbb{E}_{p(\mathbf{r},\mathbf{A})}[\Psi(\mathbf{r}, \mathbf{A}; \mathbf{x})] \tag{2.26}$$

for some function Ψ satisfying the usual regularity conditions.

2.4.1 Score function and Fisher information matrix for the correctly specified model

We provide the main results in this section. Detailed derivation can be found in Appendix 2.8.1. The subscript j will be dropped for the general derivation in the following discussion. Consider the complete data density

$$p(\mathbf{r}, \mathbf{y}|\mathbf{x}) = \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{r})}(\mathbf{y}) p(\mathbf{y}|\mathbf{x}) \tag{2.27}$$

where

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(\mathbf{y} - \mathbf{A}\mathbf{x})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \mathbf{A}\mathbf{x})\Big\}. \tag{2.28}$$

This is the scenario where the model is correctly specified. The score function becomes

$$\nabla_{\mathbf{x}} \log p(\mathbf{r}|\mathbf{x}) = \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\, \mathbb{E}_{p(\mathbf{y}|\mathbf{r},\mathbf{x})}[\mathbf{y} - \mathbf{A}\mathbf{x}] \tag{2.29}$$

$$= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\, \mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}] \tag{2.30}$$

$$= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\overline{\mathbf{w}}(\mathbf{r}) \tag{2.31}$$

where we use the abbreviation $\overline{\mathbf{w}}(\mathbf{r}) \triangleq \mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}]$. The expectation in (2.30) is taken over the interval $\mathbf{w} \in \mathcal{Q}^{-1}(\mathbf{r}) - \mathbf{A}\mathbf{x}$. The outer product form of the information matrix is

$$\boldsymbol{J}(\mathbf{x}) = \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\nabla_{\mathbf{x}} \log p(\mathbf{r}|\mathbf{x}) \nabla_{\mathbf{x}}^{\mathsf{T}} \log p(\mathbf{r}|\mathbf{x})] \tag{2.32}$$

$$= \mathbf{A}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \, \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\overline{\mathbf{w}}(\mathbf{r}) \overline{\mathbf{w}}(\mathbf{r})^{\mathsf{T}}] \boldsymbol{\Sigma}^{-1} \mathbf{A} \tag{2.33}$$

$$= \mathbf{A}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \mathbf{R}_{\overline{\mathbf{w}}(\mathbf{r})} \boldsymbol{\Sigma}^{-1} \mathbf{A}. \tag{2.34}$$

Here we define the shorthand $\mathbf{R}_{\overline{\mathbf{w}}(\mathbf{r})} \triangleq \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\overline{\mathbf{w}}(\mathbf{r}) \overline{\mathbf{w}}(\mathbf{r})^{\mathsf{T}}]$.

Remark 1. For the general case, we note that no known closed form of the expectation of a truncated multivariate normal exists. In the subsequent analysis we will evaluate this quantity using a Gibbs sampler described in Appendix 2.8.4.

We can readily obtain expression $\langle 26 \rangle$ in [57] when we specialize $\boldsymbol{\Sigma} = \sigma_\eta^2 \mathbf{I}$. Noting that $\overline{\mathbf{w}}(\mathbf{r})$ decouples and has the form

$$\overline{w}(r_i) \triangleq \mathbb{E}_{p(w|r_i,\mathbf{x})}[w] = r_i \sigma_\eta \frac{\varphi(u_i)}{\Phi(r_i u_i)} \tag{2.35}$$

for $r_i \in \{-1, 1\}$ and $u_i = \mathbf{a}_i^{\mathsf{T}} \mathbf{x} / \sigma_\eta$. For the independent case with N independent observations, we have

$$\boldsymbol{J}_{\text{indep}}(\mathbf{x}) = \mathbf{A}^{\mathsf{T}} \, \mathbb{E}_{p(\mathbf{r}|\mathbf{x})} \, \text{diag} \, \{\widehat{w}(r_i)\}_{i=1}^N \mathbf{A} \tag{2.36}$$

$$= \sum_{i=1}^N \mathbb{E}_{p(r_i|\mathbf{x})} \Big[\frac{\varphi^2(u_i)}{\Phi^2(r_i u_i)} \Big] \frac{1}{\sigma_\eta^2} \mathbf{a}_i \mathbf{a}_i^{\mathsf{T}} \tag{2.37}$$

$$= \sum_{i=1}^N \frac{1}{\sigma_\eta^2} \frac{\varphi^2(u_i)}{\Phi(u_i)\Phi(-u_i)} \, \mathbf{a}_i \mathbf{a}_i^{\mathsf{T}} \tag{2.38}$$

as required.

2.4.2 Hessian of the log-likelihood

We compute the Hessian to be used in the misspecified model. Detailed derivation can be found in Appendix 2.8.2. The Hessian is given as

$$\mathsf{D}_{\mathbf{x}}^2 \log p(\mathbf{r}|\mathbf{x}) = \mathbf{A}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\big[\mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}) - \overline{\mathbf{w}}(\mathbf{r})\overline{\mathbf{w}}(\mathbf{r})^\mathsf{T} - \boldsymbol{\Sigma}\big]\boldsymbol{\Sigma}^{-1}\mathbf{A} \tag{2.39}$$

where $\mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}) \triangleq \mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}\mathbf{w}^\mathsf{T}]$. Noting that

$$\begin{aligned}\mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r})] &= \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}\big\{\mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}\mathbf{w}^\mathsf{T}]\big\} &(2.40)\\ &= \mathbb{E}_{p(\mathbf{w},\mathbf{r}|\mathbf{x})}[\mathbf{w}\mathbf{w}^\mathsf{T}] = \boldsymbol{\Sigma}, &(2.41)\end{aligned}$$

we obtain the Hessian form of the information matrix

$$\begin{aligned}\boldsymbol{J}'(\mathbf{x}) &= \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[-\mathsf{D}_{\mathbf{x}}^2 \log p(\mathbf{r}|\mathbf{x})] &(2.42)\\ &= \mathbf{A}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\boldsymbol{\Sigma} - \mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}) + \overline{\mathbf{w}}(\mathbf{r})\overline{\mathbf{w}}(\mathbf{r})^\mathsf{T}]\boldsymbol{\Sigma}^{-1}\mathbf{A} &(2.43)\\ &= \mathbf{A}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\overline{\mathbf{w}}(\mathbf{r})\overline{\mathbf{w}}(\mathbf{r})^\mathsf{T}]\boldsymbol{\Sigma}^{-1}\mathbf{A} = \boldsymbol{J}(\mathbf{x}) &(2.44)\end{aligned}$$

which reduces to the outer product version of the Fisher information matrix as required for information matrix equivalence.

Remark 2. The above derivation is quite general in the sense that it can be applied to arbitrary quantizing functions $\mathcal{Q}$, i.e. functions that induce a finite partition on the input space Y, and arbitrary number of correlated observations and design matrix.

2.4.3 Estimator performance as a function of correlation

We have already seen in Section 2.3 that the antithetic form leads to linear estimates with reduced variance for negative correlated observations and conversely increased

variance when observations are positively correlated. The same intuition can be extended to nonlinear estimates of the generalized linear model with quantized outputs, provided that we have a large sample size and that sufficient noise is present to decorrelate among the observation pairs.

To gain insights into the relative performance as the correlation of the noise pair is varied, we proceed to study a representative model and provide a full characterization of the asymptotic behavior of the corresponding ML estimator. Using the result derived in (2.34), we analyze the trace of the Fisher information matrix for a model with

$$\mathbf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \tag{2.45}$$

and x_i drawn independently from the standard normal distribution. This model, while simple, contains the necessary features of the problem considered, namely, a full rank matrix $\mathbf{A}$ and noise pair of varying correlation ρ.

In Figure 2.3, the trace of the Fisher information matrix is plotted against differing noise correlation. Observe that the sensitivity of the ML estimator as correlation is varied is apparent. Negative correlation induced by antithetic dithering improves the performance of the estimator as measured by $\operatorname{tr} \boldsymbol{J}_{\text{rep}}$, where $\boldsymbol{J}_{\text{rep}}$ is the FIM of the representative model. Note also in the opposite extreme as the two observations become highly positive correlated, the estimator performance approaches that of an estimator given only one observation per design direction.

It is instructive to study the Fisher information expression to gain additional insights. Suppose we have a 2×2 case (2 correlated observations, 2 unknowns, 1 design vector)

$$\mathbf{A}_* = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}. \tag{2.46}$$

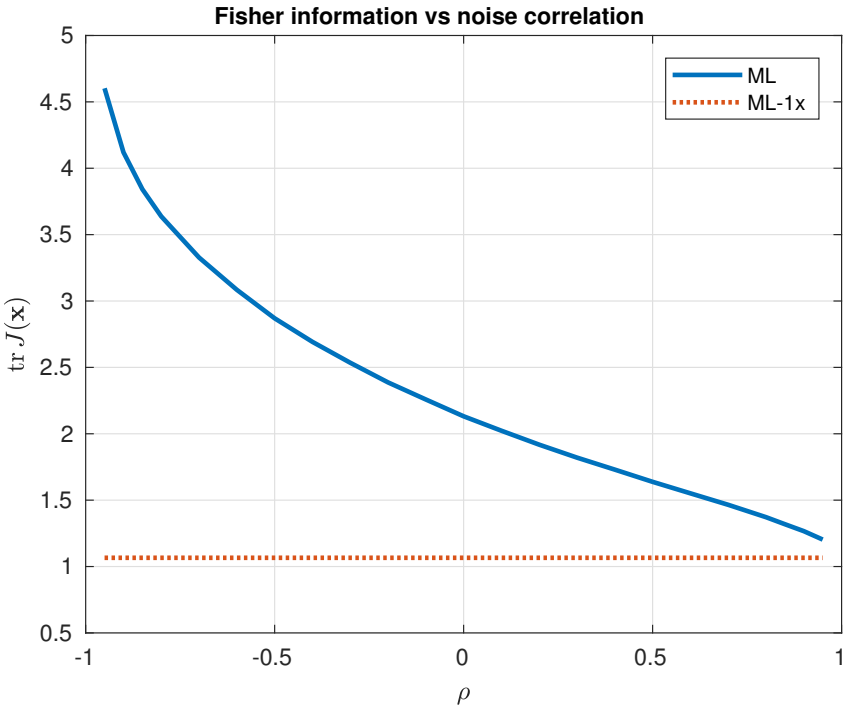


Figure 2.3: The trace of the Fisher information matrix as a function of noise correlation is labeled ML. The plot shows the convergence of the ML estimator with positively correlated noise (ML) to that of an ML estimator with single unpaired observations (ML-1x) as $\rho \to 1$.

Rewrite the fisher information matrix of the representative model $\boldsymbol{J}_{\text{rep}}(\mathbf{x})$ as

$$\boldsymbol{J}_{\text{rep}}(\mathbf{x}) = \mathbf{A}_*^{\mathsf{T}} \boldsymbol{\Gamma} \mathbf{A}_* \tag{2.47}$$

where $\boldsymbol{\Gamma} = \boldsymbol{\Sigma}^{-1} \mathbf{R}_{\overline{\mathbf{w}}(\mathbf{r})} \boldsymbol{\Sigma}^{-1}$, so

$$\boldsymbol{J}_{\text{rep}}(\mathbf{x}) = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} \gamma_{11} & \gamma_{12} \\ \gamma_{21} & \gamma_{22} \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}. \tag{2.48}$$

It can be seen that $\boldsymbol{\Gamma}$ is related to $\boldsymbol{\Sigma}^{-1}$. Observe that the $(1,1)$-entry of $\boldsymbol{J}_{\text{rep}}$ effectively sums together all elements in $\boldsymbol{\Gamma}$. In the negative correlated case, all elements of $\boldsymbol{\Gamma}$ are positive. On the other hand, the off-diagonal elements are zero and negative

respectively for uncorrelated and positively correlated scenarios. Antithetic dithering is most effective if $\mathbf{\Gamma}$ is positive. Even if negative correlation cannot be achieved mutually in all variables, as long as the sum of all elements in $\mathbf{\Gamma}$ is greater, we still have improvement over the independent case. We state this result formally:

Theorem 1. Suppose we have two generalized linear models with multiple correlated observations as in (2.24). Express the Fisher information matrices as $\boldsymbol{J}_1(\mathbf{x}) = \mathbf{A}_*^\top \mathbf{\Gamma}_1 \mathbf{A}_*$ and $\boldsymbol{J}_2(\mathbf{x}) = \mathbf{A}_*^\top \mathbf{\Gamma}_2 \mathbf{A}_*$. Denote the element sum of a matrix $\mathbf{G}$ as $\mathrm{sum}(\mathbf{G}) = \mathbf{1}^\top \mathbf{G} \mathbf{1}$. If $\mathrm{sum}(\mathbf{\Gamma}_1) \geq \mathrm{sum}(\mathbf{\Gamma}_2)$ then the Fisher information measure $\mathrm{tr}\, \boldsymbol{J}(\mathbf{x})$ satisfies

$$\mathrm{tr}\, \boldsymbol{J}_1(\mathbf{x}) \geq \mathrm{tr}\, \boldsymbol{J}_2(\mathbf{x}). \tag{2.49}$$

Proof. The result follows immediately from inspection of the matrices $\boldsymbol{J}_1, \boldsymbol{J}_2$ in the stated form. □

2.4.4 Pseudo maximum likelihood

In order to construct a simple and more robust estimator while preserving the optimality of an ML estimator, we explore one of the commonly techniques used in statistics. We precede the analysis with a brief introduction of the relevant results. It is a known fact that when the true distribution is unknown, the maximum likelihood estimator is an estimator of parameters that minimizes the Kullback-Leibler (KL) distance between the true distribution and the specified family of distributions [68, 69]. In addition, as long as the true distribution is symmetric, maximum likelihood performed under the assumption of normality (e.g. Least squares) yields consistent estimate of the mean. This technique and many others that perform inference using a modified likelihood function are special cases of the more general *pseudo likelihood method* [70]. It has been used in statistics in situations where the likelihood is complicated by high-dimensional dependencies, or

in a pragmatic light, where a simplified likelihood offers more robustness. Under mild conditions, the pseudo maximum likelihood can be shown to preserve consistency and asymptotic normality with only a small loss of efficiency.

Precisely, the pseudo log-likelihood of N i.i.d observations $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$ is defined as

$$\mathcal{L}(\mathbf{Y}; \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \log f(\mathbf{y}_i; \boldsymbol{\theta}) \tag{2.50}$$

where the true distribution $p(\mathbf{Y})$ is not an element of the parameterized family $\{f(\mathbf{Y}; \boldsymbol{\theta}) : \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$, with $f(\mathbf{Y}; \boldsymbol{\theta}) \equiv \prod_{i=1}^{N} f(\mathbf{y}_i; \boldsymbol{\theta})$. The corresponding pseudo maximum likelihood estimator is defined as the random function

$$\widehat{\boldsymbol{\theta}}_{\mathrm{PML}}(\mathbf{Y}) = \arg\max_{\boldsymbol{\theta} \in \boldsymbol{\Theta}} \mathcal{L}(\mathbf{Y}; \boldsymbol{\theta}). \tag{2.51}$$

For the antithetic dithered model, we provide the following arguments that make estimating on an approximate likelihood appealing. First, the absolute covariance is unidentifiable in this setup thus it is desirable to perform estimation without this knowledge. Second, the sole intention of the induced correlation is to improve performance hence it is effectively a nuisance parameter. Additionally, we will show in the sequel that an approximate likelihood consisting of only marginal densities leads to a low complexity EM algorithm.

We begin the theoretical analysis with an important asymptotic normality result from [68] to evaluate asymptotic estimator covariance, and we restate it here for convenience.

Theorem 2. (Asymptotic normality of pseudo MLE [68, Theorem 3.2]) Define the quali-

ties

$$\boldsymbol{A}(\boldsymbol{\theta}) = \mathbb{E}\big\{\nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^{\mathsf{T}} \log f(\mathbf{y}; \boldsymbol{\theta})\big\}, \tag{2.52}$$

$$\boldsymbol{B}(\boldsymbol{\theta}) = \mathbb{E}\big\{\nabla_{\boldsymbol{\theta}} \log f(\mathbf{y}; \boldsymbol{\theta}) \cdot \nabla_{\boldsymbol{\theta}}^{\mathsf{T}} \log f(\mathbf{y}; \boldsymbol{\theta})\big\}, \tag{2.53}$$

$$\boldsymbol{C}(\boldsymbol{\theta}) = \boldsymbol{A}(\boldsymbol{\theta})^{-1} \boldsymbol{B}(\boldsymbol{\theta}) \boldsymbol{A}(\boldsymbol{\theta})^{-1} \tag{2.54}$$

where the expectation is taken with respect to the true distribution. Then the n-sample pseudo MLE estimate $\widehat{\boldsymbol{\theta}}_n = \arg\max_{\boldsymbol{\theta}} \log f(\mathbf{y}; \boldsymbol{\theta})$ satisfies

$$\sqrt{n}(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \boldsymbol{C}(\boldsymbol{\theta}^*)) \tag{2.55}$$

where $\boldsymbol{\theta}^*$ is the unique identifiable parameter that minimizes the KL distance between the true density g and the parameterized function $f(\,\cdot\,; \boldsymbol{\theta}^*)$.

In order to characterize the efficiency of the pseudo MLE, we develop the asymptotic matrices for the misspecified model, which corresponds to the parameterized *pseudo* likelihood function we seek to maximize.

2.4.5 Score, Hessian and the asymptotic covariance matrix for the misspecified model

For the misspecified model, consider using the model with distribution

$$q(\mathbf{r}, \mathbf{y}|\mathbf{x}) = \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{r})}(\mathbf{y}) q(\mathbf{y}|\mathbf{x}) \tag{2.56}$$

where

$$q(\mathbf{y}|\mathbf{x}) = \frac{1}{|2\pi\mathbf{I}|^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(\mathbf{y} - \mathbf{A}\mathbf{x})^{\mathsf{T}}(\mathbf{y} - \mathbf{A}\mathbf{x})\Big\}. \tag{2.57}$$

The significance of this model hinges on its complete ignorance of the correlation between observations. Maximizing this likelihood reduces to estimation via their marginals. We will see the fruitfulness of employing this likelihood in section 2.5.5. Computation of the score function and the Hessian gives us

$$\nabla_{\mathbf{x}} \log q(\mathbf{r}|\mathbf{x}) = \mathbf{A}^{\mathsf{T}} \overline{\mathbf{w}}(\mathbf{r}; q), \tag{2.58}$$

$$\mathsf{D}^2_{\mathbf{x}} \log q(\mathbf{r}|\mathbf{x}) = -\mathbf{A}^{\mathsf{T}} \big[\mathbf{I} - \mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}; q) + \overline{\mathbf{w}}(\mathbf{r}; q)\overline{\mathbf{w}}(\mathbf{r}; q)^{\mathsf{T}}\big]\mathbf{A} \tag{2.59}$$

$$= -\mathbf{A}^{\mathsf{T}} \big[\mathbf{I} - \mathbf{\Sigma}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}; q)\big]\mathbf{A} \tag{2.60}$$

where $\mathbf{\Sigma}_{\mathbf{w}|\mathbf{r}} = \mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}) - \overline{\mathbf{w}}(\mathbf{r})\overline{\mathbf{w}}(\mathbf{r})^{\mathsf{T}}$ is the covariance matrix of $\mathbf{w}|\mathbf{r}$. The asymptotic covariance matrix is

$$\boldsymbol{C}(\mathbf{x}) = \boldsymbol{A}(\mathbf{x})^{-1} \boldsymbol{B}(\mathbf{x}) \boldsymbol{A}(\mathbf{x})^{-1} \tag{2.61}$$

with $\boldsymbol{A}$ and $\boldsymbol{B}$ taking the form

$$\boldsymbol{A}(\mathbf{x}) = \mathbf{A}^{\mathsf{T}} \mathbb{E}_{p(\mathbf{r}|\mathbf{x})} \big[\mathbf{I} - \mathbf{\Sigma}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}; q)\big]\mathbf{A}, \tag{2.62}$$

$$\boldsymbol{B}(\mathbf{x}) = \mathbf{A}^{\mathsf{T}} \mathbb{E}_{p(\mathbf{r}|\mathbf{x})} [\overline{\mathbf{w}}(\mathbf{r}; q)\overline{\mathbf{w}}(\mathbf{r}; q)^{\mathsf{T}}]\mathbf{A}. \tag{2.63}$$

Noting that the conditional variance of $w_i|r_i, \mathbf{x}$ [71] for independent w_i is

$$\mathrm{var}_{p(w_i|r_i,\mathbf{x})}[w^2] = \sigma_\eta^2 \Big[1 - \frac{s_i \varphi(s_i)}{\Phi(s_i)}\Big] - \sigma_\eta^2 \Big[\frac{\varphi(s_i)}{\Phi(s_i)}\Big]^2 \tag{2.64}$$

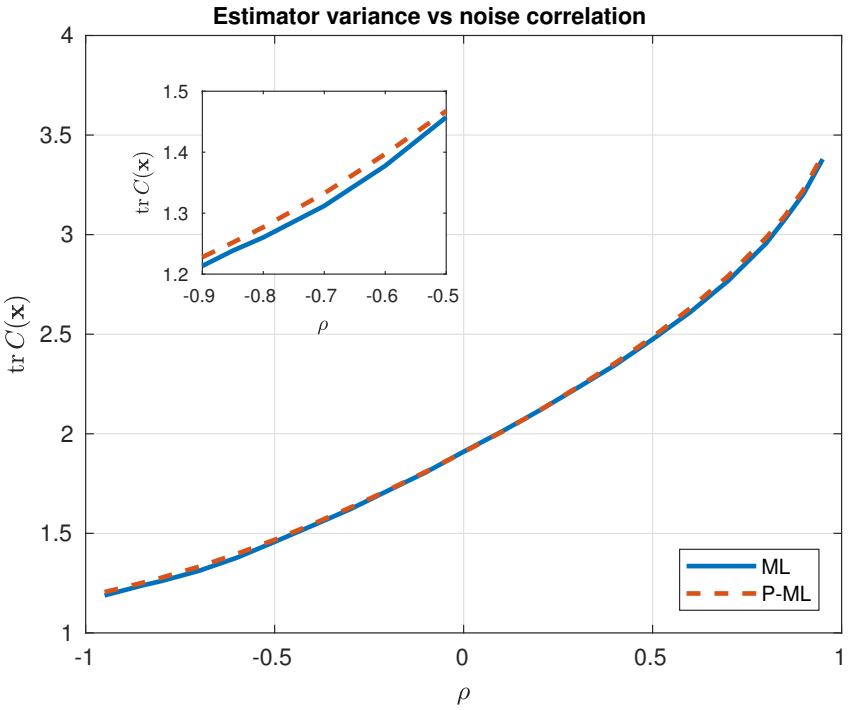


Figure 2.4: The trace of the asymptotic estimator covariance matrix for the pseudo-ML estimator (P-ML) using the misspecified model and the true ML estimator (ML) using the true likelihood function are compared.

where $s_i = r_i \mathbf{a}_i^\mathsf{T} \mathbf{x}/\sigma_\eta$. When $\sigma_\eta = 1$ and N observations are available, these matrices become

$$\boldsymbol{A}(\mathbf{x}) = \mathbf{A}^\mathsf{T} \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\mathbf{D}(\mathbf{r}; q)]\mathbf{A} \tag{2.65}$$

$$\boldsymbol{B}(\mathbf{x}) = \mathbf{A}^\mathsf{T} \mathbb{E}_{p(\mathbf{r}|\mathbf{x})}[\mathbf{T}(\mathbf{r}; q)]\mathbf{A}. \tag{2.66}$$

Here $\mathbf{D}$ and $\mathbf{T}$ are respectively

$$\mathbf{D}(\mathbf{r}; q) = \operatorname{diag}\left\{ \frac{s_i \varphi(s_i)}{\Phi(s_i)} + \frac{\varphi^2(s_i)}{\Phi^2(s_i)} \right\}_{i=1}^{N}, \tag{2.67}$$

$$[\mathbf{T}(\mathbf{r}; q)]_{ij} = r_i r_j \frac{\varphi(s_i)}{\Phi(s_i)} \frac{\varphi(s_j)}{\Phi(s_j)}. \tag{2.68}$$

We note that s_i in (2.67) and (2.68) is reduced to $s_i = r_i \, \mathbf{a}_i^\mathsf{T} \mathbf{x}$.

2.4.6 Loss of efficiency relative to true likelihood

Using the results in (2.34), (2.65) and (2.66), we compare the trace of the asymptotic covariance of the true ML and pseudo ML estimate as a function of noise correlation. The resultant plot is shown in Figure 2.4. The pseudo ML estimate has slightly increased variance when noise is correlated. As expected, the performance converges to that of the true ML estimator as $\rho \to 0$ since the misspecified likelihood tends to the true likelihood. This provides a rigorous theoretical justification for the use of simplified algorithms that maximize a suitably defined pseudo likelihood.

2.5 Likelihood function and EM algorithms for 1-bit ADC

In this section, we characterize the structural properties of the likelihood function from the point of view of optimization and algorithm construction, derive the EM algorithm for the general correlated case and present a PX-EM algorithm that can readily exploit the structure of the problem to provide a rate of convergence improvement over the standard EM technique. We introduce a Markov Chain Monte Carlo (MCMC) algorithm as an empirical lower bound to compare the performance of the proposed scheme.

2.5.1 Log-likelihood function

Returning to the original model (2.4), let $\mathbf{a}_i^\mathsf{T}, i = 1, \ldots, N$ be the ith row of $\mathbf{A}$, $\mathbf{r} = [r_1, \ldots, r_N]$. The log-likelihood has the form

$$\mathcal{L}(\mathbf{x}, \sigma^2; \mathbf{r}) = \sum_{r_i=1} \log\Big[\Phi\Big(\mathbf{a}_i^\mathsf{T}\frac{\mathbf{x}}{\sigma}\Big)\Big] + \sum_{r_i=-1} \log\Big[1 - \Phi\Big(\mathbf{a}_i^\mathsf{T}\frac{\mathbf{x}}{\sigma}\Big)\Big] \tag{2.69}$$

$$\mathcal{L}(\mathbf{x}'; \mathbf{r}) \triangleq \sum_{r_i=1} \log\Big[\Phi\Big(\mathbf{a}_i^\mathsf{T}\mathbf{x}'\Big)\Big] + \sum_{r_i=-1} \log\Big[\Phi\Big(-\mathbf{a}_i^\mathsf{T}\mathbf{x}'\Big)\Big]. \tag{2.70}$$

We remark here that only the ratio $\mathbf{x}/\sigma$ can be identified in this model. Note that identifiability of $\mathbf{x}' \triangleq \mathbf{x}/\sigma$ and consistency of the estimate for this model is satisfied when $\mathbf{A}$ has full column rank. The log-likelihood function is globally concave in $\mathbf{x}'$ so any iterative algorithm that monotonically increase the value of the log-likelihood will converge to the global maximum. This follows from the fact that φ is log-concave and the cdf of a log-concave density is log-concave. We highlight here that this model is equivalent to the probit model in the statistics literature. The log-concavity property applies more generally to the multiple correlated observations scenario since the multivariate normal density $p(\mathbf{w})$ is log concave and the likelihood can be constructed as product of functions having the form [72]

$$f_i(\mathbf{x}) = \mathbb{P}(\mathbf{A}\mathbf{x} + \mathbf{w} \in R_i) \tag{2.71}$$

$$= \int p(\mathbf{w})\, \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{r}_i)}(\mathbf{A}\mathbf{x} + \mathbf{w}) d\mathbf{w} \tag{2.72}$$

where $\mathcal{Q}^{-1}(\mathbf{r}_i)$ are convex sets. We continue our discussions using the normalized form (2.70).

Remark 3. Since the parameters $\boldsymbol{\theta} = (\mathbf{x}, \sigma^2)$ are not identifiable, joint estimation algorithm such as ECME [73] cannot be used. Hence we resort to other methods to decouple

the parameters. The introduction of a known dither provides a simple solution using only 1-bit ADCs. We restrict our attention to the algorithm development and leave this discussion in Appendix 2.8.3. In the sequel we assume σ_w is available and it will be used to adjust the final estimate in the algorithms.

2.5.2 General EM derivation

We derive the E-step of the EM algorithm for the correlated case, assuming $\mathbf{\Sigma}$ is known:

$$
\begin{aligned}
& Q(\mathbf{x}, \widehat{\mathbf{x}}^{(k)}) && (2.73)\\
& = \mathbb{E}_{\mathbf{y}|\mathbf{r},\widehat{\mathbf{x}}^{(k)}}[\log p(\mathbf{r}, \mathbf{y}|\mathbf{x})] && (2.74)\\
& = \mathbb{E}_{\mathbf{y}|\mathbf{r},\widehat{\mathbf{x}}^{(k)}}[\log p(\mathbf{y}|\mathbf{x}) + \log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{r})}(\mathbf{y})] && (2.75)\\
& \doteq \mathbb{E}_{\mathbf{y}|\mathbf{r},\widehat{\mathbf{x}}^{(k)}}[-\tfrac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|^2_{\mathbf{\Sigma}^{-1}}] && (2.76)\\
& \doteq 2\mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{\Sigma}^{-1}\,\mathbb{E}_{\mathbf{y}|\mathbf{r},\widehat{\mathbf{x}}^{(k)}}[\mathbf{y}] - \mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{A}\mathbf{x} && (2.77)\\
& = 2\mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{\Sigma}^{-1}(\mathbf{A}\widehat{\mathbf{x}}^{(k)} + \mathbb{E}_{\mathbf{w}|\mathbf{r},\widehat{\mathbf{x}}^{(k)}}[\mathbf{w}]) - \mathbf{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{A}\mathbf{x}, && (2.78)
\end{aligned}
$$

where we use the symbol $\doteq$ to mean equivalence for the purpose of ML inference. This amounts to finding the conditional mean of $\mathbf{w}|\mathbf{r}, \widehat{\mathbf{x}}^{(k)}$. However, evaluation of this quantity is challenging in general because the expectation of truncated multivariate Gaussian admits no close form. The M-step is equivalent to maximizing the likelihood of a Gaussian linear model with expected quantizer input

$$
\widehat{\mathbf{y}}^{(k)} \triangleq \mathbf{A}\widehat{\mathbf{x}}^{(k)} + \mathbb{E}_{\mathbf{w}|\mathbf{r},\widehat{\mathbf{x}}^{(k)}}[\mathbf{w}]. \tag{2.79}
$$

Algorithm 1: PX-EM algorithm

Input: 1-bit outputs $\mathbf{r} = [r_1, \ldots, r_N]^\mathsf{T}$, design matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]^\mathsf{T}$, σ_w, $\epsilon > 0$

States: Latent variables $\mathbf{y} = [y_1, \ldots, y_N]^\mathsf{T}$

Initialize $\mathbf{x}^{(1)} = \mathbf{0}$, $\alpha = 1$

for $n := 1$ *to* N_{max} **do**

  E-step: For each i, impute y_i

$$y_i^{(n+1)} = \mathbf{a}_i^\mathsf{T}\mathbf{x}^{(n)} + \frac{r_i \varphi(s_i^{(n)})}{\Phi(r_i s_i^{(n)})},\ s_i^{(n)} = \frac{\mathbf{a}_i^\mathsf{T}\mathbf{x}^{(n)}}{\alpha^{(n)}} \tag{2.80}$$

  M-step: Estimate working parameter α^2

$$(\alpha^2)^{(n+1)} = \frac{1}{N}\sum_{i=1}^{N}(y_i^{(n+1)} - \mathbf{a}_i^\mathsf{T}\mathbf{x}^{(n)})^2 \tag{2.81}$$

  Compute the LS solution

$$\mathbf{x}^{(n+1)} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{y}^{(n+1)} \tag{2.82}$$

  Terminate loop when $\|\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}\| < \epsilon$

end

Set $\mathbf{x}^* = \sigma_w(\mathbf{x}^{(n+1)}/\alpha^{(n+1)})$

2.5.3 PX-EM algorithm

The EM algorithm is well known and has been extensively used in the community, we simply refer the interested readers to classic results in [74, 75]. The utility of EM algorithm notwithstanding, one often cited shortcoming is its slow convergence. Various EM extensions have been successful in speeding up the algorithm, we propose one variant called parameter expanded EM (PX-EM) [76, 77] that is well suited for the 1-bit ADC model. We explain the concept using the idea introduced in [77]. Let $\mathbf{r}$ be the observed data, $\mathbf{y}$ the missing data and θ the parameter of interest. Traditional EM sidesteps the need to compute a complex observed likelihood $f(\mathbf{r}|\theta)$ by augmenting the data space with missing data such that working with complete data likelihood $f(\mathbf{r}, \mathbf{y}|\theta)$ makes the problem more tractable. Like other EM extensions, PX-EM preserves the monotone convergence of EM. PX-EM additionally introduces a hidden identifiable parameter α to the complete-data model with the condition that the expanded model $f(\mathbf{r}, \mathbf{y}_\alpha|\theta, \alpha)$ preserves the observed data model $f(\mathbf{r}|\theta)$. Precisely, the probability distribution $f(\mathbf{r}, \mathbf{y}_\alpha|\theta, \alpha)$ must satisfy

$$\int f(\mathbf{r}, \mathbf{y}_\alpha|\theta, \alpha) d\mathbf{y}_\alpha = f(\mathbf{r}|\theta). \tag{2.83}$$

The key intuition is that by expanding the parameter space from θ to (θ, α), the PX-EM algorithm is allowed to move more freely along a family of orbits parameterized by α. C. Liu et al. [76] showed that the rate of convergence of PX-EM is at least as fast as the original EM and the rate improvement can be dramatic in some cases.

2.5.4 PX-EM for 1-bit ADC

We now introduce the steps in the derivation of PX-EM for the 1-bit ADC model using the methodology developed in [76]. We note that the complete data model for

standard EM is of the form

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{A}\mathbf{x}, \mathbf{I}) \tag{2.84}$$

$$\mathbf{r}|\mathbf{y} = \mathcal{Q}(\mathbf{y}). \tag{2.85}$$

This model can be expanded by allowing the algorithm to estimate the residual variance α^2 so that we have the following expanded model

$$\mathbf{y}|\widetilde{\mathbf{x}}, \alpha \sim \mathcal{N}(\mathbf{A}\widetilde{\mathbf{x}}, \alpha^2\mathbf{I}) \tag{2.86}$$

$$\mathbf{r}|\mathbf{y} = \mathcal{Q}(\mathbf{y}). \tag{2.87}$$

In the original model, the parameter α can be viewed as a hidden parameter fixed at 1, however this parameter becomes identifiable in the expanded model. In order for this model to be a valid expansion, we require the following conditions to be met:

1. **(Preservation of the observed data model)** There is a many-to-one reduction function $\boldsymbol{h}$ such that $p(\mathbf{r}|\widetilde{\mathbf{x}}, \alpha) = p(\mathbf{r}|\mathbf{x} = \boldsymbol{h}(\widetilde{\mathbf{x}}, \alpha))$. The reduction $\boldsymbol{h} : (\widetilde{\mathbf{x}}, \alpha) \mapsto \mathbf{x}$ for this model is

   $$\mathbf{x} = \boldsymbol{h}(\widetilde{\mathbf{x}}, \alpha) = \widetilde{\mathbf{x}}/\alpha. \tag{2.88}$$

   This stems from the non-identifiability of $(\mathbf{x}, \sigma^2)$ in the 1-bit model. To see this, let $\widetilde{\mathbf{x}}_0$ and $\alpha_0 = 1$ be the true parameter values, the observed data $\mathbf{r}$ can be explained by any element in the set $\{(\widetilde{\mathbf{x}}, \alpha) \mid \widetilde{\mathbf{x}} = \widetilde{\mathbf{x}}_0/\alpha,\ \alpha > 0\}$.

2. **(Preservation of the complete data model)** At the null value $\alpha_0 = 1$, and for all $\mathbf{x}$ we require

   $$p(\mathbf{r}, \mathbf{y}|\widetilde{\mathbf{x}}, \alpha_0) = p(\mathbf{r}, \mathbf{y}|\mathbf{x} = \widetilde{\mathbf{x}}). \tag{2.89}$$

   We note that when $\alpha = 1$, the distribution of $\mathbf{y}$ from the original model and the

expanded model are identical. Also since $\mathbf{r}$ is completely determined by $\mathbf{y}$, the equivalence of the joint distribution follows.

We note that the two conditions above ensure that there is at least one element $(\widetilde{\mathbf{x}}, \alpha)$ in the expanded space that explains the data i.e. $p(\mathbf{r}|\widetilde{\mathbf{x}}, \alpha) = p(\mathbf{r}|\mathbf{x} = \boldsymbol{h}(\widetilde{\mathbf{x}}, \alpha))$ and with measure zero if $\mathbf{x}_1 \neq \mathbf{x}_2$, then $\boldsymbol{h}^{-1}(\mathbf{x}_1) \neq \boldsymbol{h}^{-1}(\mathbf{x}_2)$, i.e. the two orbits in the expanded parameter space are essentially disjoint. The set of reduction functions $\{\boldsymbol{h}_\alpha, \alpha \in \mathcal{A}\}$ typically has a group structure [77], e.g. it forms a translation or scale group.

At this point, the same EM steps are used on the expanded model to realize the algorithm. For the unpaired 1-bit model, note that $y_i|\mathbf{r}, \mathbf{x}$ are conditionally independent. In the E-step, the conditional expectation of $\mathbf{y}$ reduces to

$$\mathbb{E}_{y_i|\mathbf{r},\mathbf{x}^{(k)},\alpha^{(k)}}[y_i] = \mathbf{a}_i^\mathsf{T}\mathbf{x}^{(k)} + \frac{r_i \varphi(s_i^{(k)})}{\Phi(r_i s_i^{(k)})}, \ \forall i \tag{2.90}$$

where $s_i = \frac{\mathbf{a}_i^\mathsf{T}\mathbf{x}}{\alpha}$. Denote this quantity $y_i^{(k+1)}$. The M-step corresponds to

$$(\alpha^2)^{(k+1)} = \frac{1}{N}\sum_{i=1}^{N}(y_i^{(k+1)} - \mathbf{a}_i^\mathsf{T}\mathbf{x}^{(k)})^2 \tag{2.91}$$

$$\mathbf{x}^{(k+1)} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{y}^{(k+1)}. \tag{2.92}$$

Note that in (2.91) the previous estimate of $\mathbf{x}$ is used to compute α^2. This has been shown empirically to provide a slightly faster convergence compared to using the current residue. The full PX-EM algorithm is listed in Algorithm 1.

Remark 4. It is worthwhile to note that the estimated residue variance α^2 is an intermediate piece of data the algorithm uses to refine its trajectory. At convergence, this quantity is equal to the conditional variance of $w_i|\mathbf{r}, \mathbf{x}^*$, which is a biased estimate of the variance of the actual random noise.

Algorithm 2: PX-EM-AT algorithm

Input: 1-bit outputs $\mathbf{r} = [\mathbf{r}_1; \mathbf{r}_2]$, $\mathbf{r}_d = [r_{d1}, \ldots, r_{dN}]^\mathsf{T}$, design matrix $\mathbf{B} = [\mathbf{A}; \mathbf{A}]$, $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]^\mathsf{T}$, σ_w, $\epsilon > 0$

States: Latent variables $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2]$, $\mathbf{y}_d = [y_{d1}, \ldots, y_{dN}]^\mathsf{T}$

Initialize $\mathbf{x}^{(1)} = \mathbf{0}$, $\alpha = 1$

for $n := 1$ *to* N_{max} **do**

E-step: For each i and $d = 1, 2$, impute y_{di}

$$y_{di}^{(n+1)} = \mathbf{a}_i^\mathsf{T}\mathbf{x}^{(n)} + \frac{r_{di}\varphi(s_i^{(n)})}{\Phi(r_{di}s_i^{(n)})},\ s_i^{(n)} = \frac{\mathbf{a}_i^\mathsf{T}\mathbf{x}^{(n)}}{\alpha^{(n)}} \tag{2.93}$$

M-step: Estimate working parameter α^2

$$(\alpha^2)^{(n+1)} = \frac{1}{2N}\sum_{i=1}^{N}\sum_{d=1}^{2}(y_{di}^{(n+1)} - \mathbf{a}_i^\mathsf{T}\mathbf{x}^{(n)})^2 \tag{2.94}$$

Compute the LS solution

$$\mathbf{x}^{(n+1)} = (\mathbf{B}^\mathsf{T}\mathbf{B})^{-1}\mathbf{B}^\mathsf{T}\mathbf{y}^{(n+1)} \tag{2.95}$$

Terminate loop when $\|\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}\| < \epsilon$

end

Set $\mathbf{x}^* = \sigma_w(\mathbf{x}^{(n+1)}/\alpha^{(n+1)})$

2.5.5 PX-EM for antithetic dither ADC pairs

We step through the procedure to arrive at the PX-EM-AT algorithm. Note that in the original normalized likelihood of the complete data model is given by

$$\mathbf{y}_{*i}|\mathbf{x} \sim \mathcal{N}(\mathbf{A}_{*i}^{\mathsf{T}}\mathbf{x}, \mathbf{\Sigma}) \tag{2.96}$$

$$\mathbf{r}_{*i}|\mathbf{y}_{*i} = \mathcal{Q}(\mathbf{y}_{*i}) \tag{2.97}$$

where $\mathbf{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. Alternatively the model can be expanded to include the residue variance α^2 such that we have

$$\widetilde{\mathbf{y}}_{*i}|\widetilde{\mathbf{x}}, \alpha \sim \mathcal{N}(\mathbf{A}_{*i}^{\mathsf{T}}\widetilde{\mathbf{x}}, \alpha^2\mathbf{\Sigma}) \tag{2.98}$$

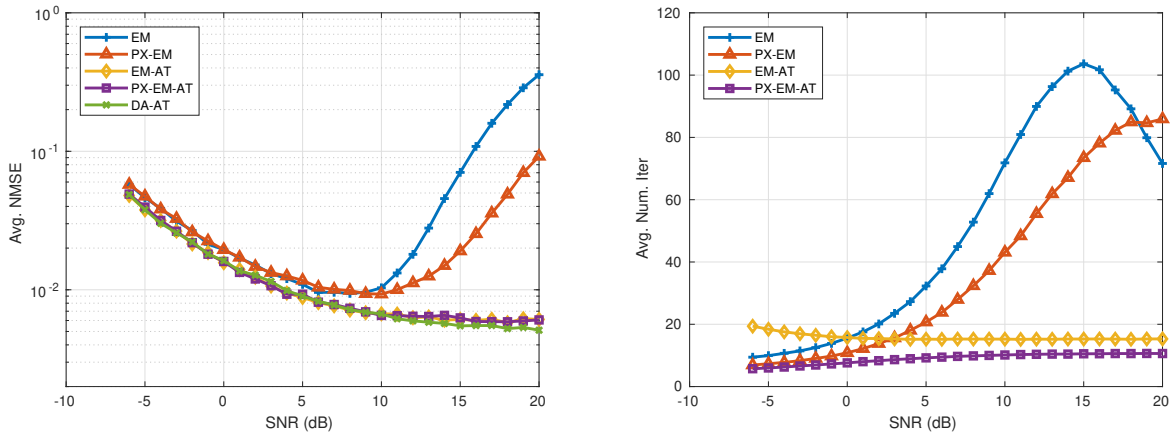
$$\mathbf{r}_{*i}|\mathbf{y}_{*i} = \mathcal{Q}(\widetilde{\mathbf{y}}_{*i}) \tag{2.99}$$

with $\mathbf{x} = \widetilde{\mathbf{x}}/\alpha$. When $\alpha^2 = 1$ we recover the original model. Here we assume the residue variance is the same in all observations thus we will be able to include all residues to estimate α^2.

Noting that the induced correlation is effectively a nuisance parameter, we employ the same strategy that ignores the covariance information and use an EM algorithm that climbs a pseudo likelihood function instead. Two approximations are made in the PX-EM-AT algorithm (Algorithm 2).

1. In the E-step, instead of taking the joint conditional expectation of $\mathbf{w}|\mathbf{r}, \mathbf{x}$, we approximate this by the expectation over the conditional marginals $w_i|r_i, \mathbf{x}$.

2. The optimal M-step is the Gauss Markov solution which includes the noise covariance $\mathbf{\Sigma}$. We approximate this using LS since covariance information is not known at this point.



(a) NMSE performance comparison

(b) Rate of convergence comparison

Figure 2.5: NMSE performance and rate of convergence of standard EM (EM), parameter expanded EM (PX-EM) using unpaired, non-dithered 1-bit ADCs, and standard EM (EM-AT), parameter expanded EM (PX-EM-AT) using antithetic dithered 1-bit ADCs.

Examining the algorithm closely, we realized that the algorithm is in fact climbing the pseudo likelihood defined in (2.56).

2.5.6 Exact-DA algorithm

To provide further empirical evidence that the proposed approximate EM algorithm performs adequately, we compare its performance with a computationally intensive data-augmentation (DA) algorithm [78, 79] that samples the posterior of the unknowns given the 1-bit observations, a flat prior and known noise covariance. The algorithm is asymptotically exact in the sense that as the number of samples tends to infinity, the posterior is fully recovered. The implementation of this algorithm involves embedding the truncated bivariate sampler into the DA algorithm. The algorithm is described in Algorithm 3.

2.6 Numerical results

A real valued system is generated with parameters $K = 10$, $N = 128 \times K$. SNR is defined as

$$\mathsf{SNR} = \frac{\mathbb{E}[\|\mathbf{Ax}\|^2]}{\mathbb{E}[\|\mathbf{n}_1\|^2]}. \tag{2.100}$$

The noise vectors $\mathbf{n}_1, \mathbf{n}_2$ are identical. The elements of matrix $\mathbf{A}$ is generated by sampling from an iid zero mean Gaussian variable. A dither with fixed known variance is generated such that $\mathbb{E}[\|\mathbf{d}\|^2] = \mathbb{E}[\|\mathbf{Ax}\|^2]$. This assumption is reasonable because typically automatic gain control (AGC) performs normalization on the received signal to some targeted level. Since performance and speed are conflicting objectives i.e. lowering NMSE increases convergence time and vice versa, it is only possible to optimize the dither power with application specific requirements. All results are plotted in Figure 2.5.

First, we compare the performance of EM and PX-EM in the original non-dithered single ADC model. The convergence rate comparison is plotted in Figure 2.5b. We use the average number of iterations for an algorithm to reach a fixed threshold ϵ as a measure of the rate of convergence. As seen in Figure 2.5b, the PX-EM has a faster rate of convergence at all SNRs. However due to the compression effect of the 1-bit ADC, both NMSE and rate of convergence deteriorate at high SNR. We remark here that at high SNR PX-EM has observably better performance compared to traditional EM, this may be explained in part by the quasi-concavity of the likelihood and the freedom of movement afforded by parameter expansion.

Next we introduce antithetic dithering using ADC pairs, the same rate of convergence improvement is seen in Figure 2.5b when EM-AT and PX-EM-AT are compared. However, we now have a stable monotonic increase in estimation performance and a significant improvement in convergence rate. Note that the convergence rate of PX-EM-AT outperforms all algorithms and topologies considered. Also from an implementation stand-

point, we can eliminate the threshold checking (step 3) and simply run the algorithm up to the upper bound of convergence in all SNRs using PX-EM-AT.

Since we have little theoretical understanding of this scenario, the authors are at first surprised by the outcome; that adding negative correlated dithering to the system improves NMSE at all SNRs. One interpretation of the result is that we have a constrained 2-bit representation of the received signal as opposed to single unpaired 1-bit representation in the standard setting.

Finally we compare the NMSE performance of PX-EM-AT and the asymptotically exact DA algorithm (DA-AT). We see in Figure 2.5a that the performance degrades only slightly as predicted by the pseudo-ML analysis.

2.7 Conclusion

We developed the notion of negative correlation and demonstrated how this can be exploited in systems that experience nonlinear monotonic transformation to recover the parameters of interest. Both linear and nonlinear ML estimators are shown to benefit from negative correlation. For the specific 1-bit antithetic dithered massive MIMO system considered, a fast PX-EM algorithm applied to such a system is shown to exhibit fast convergence, possessing an upper bounded convergence guarantee and a graceful monotonic estimation performance over large region of SNR range. The approximate PX-EM algorithm is shown to be an instance of the pseudo-ML method and as such enjoys the consistency and asymptotic normality property with little loss of efficiency. The proposed antithetic architecture requires a modest increase in complexity compared to the conventional 1-bit receiver. We have illustrated the potential symbol estimation improvement when a linear estimator is used at high SNR and provided a rigorous analysis for a low complexity nonlinear estimator for channel estimation over a wide range of SNRs.

Chapter 2, in part, is a reprint of the material as it appears in the paper: D. K. W. Ho and B. D. Rao, "Antithetic dithered 1-bit massive MIMO architecture: Efficient channel estimation via parameter expansion and PML", *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp.2291-2303, May 2019. The dissertation author was the primary investigator and author of this paper.

2.8 Appendix

2.8.1 Derivation of the score function

Using the identity

$$\nabla_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x}), \tag{2.101}$$

the score function is

$$\begin{aligned}
&\nabla_{\mathbf{x}} \log p(\mathbf{r}|\mathbf{x}) \\
&\quad = \frac{1}{p(\mathbf{r}|\mathbf{x})}\nabla_{\mathbf{x}} \int p(\mathbf{r},\mathbf{y}|\mathbf{x})d\mathbf{y} && (2.102)\\
&\quad = \frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{\mathcal{Q}^{-1}(\mathbf{r})} \nabla_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})d\mathbf{y} && (2.103)\\
&\quad = \frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{\mathcal{Q}^{-1}(\mathbf{r})} p(\mathbf{y}|\mathbf{x})\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x})d\mathbf{y} && (2.104)\\
&\quad = \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\int (\mathbf{y}-\mathbf{A}\mathbf{x})\frac{p(\mathbf{r},\mathbf{y}|\mathbf{x})}{p(\mathbf{r}|\mathbf{x})}\cdot\frac{p(\mathbf{y}|\mathbf{r},\mathbf{x})}{p(\mathbf{y}|\mathbf{r},\mathbf{x})}d\mathbf{y} && (2.105)\\
&\quad = \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{y}|\mathbf{r},\mathbf{x})}[\mathbf{y}-\mathbf{A}\mathbf{x}] && (2.106)\\
&\quad = \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}]. && (2.107)
\end{aligned}$$

2.8.2 Hessian of the log-likelihood

Using the formula

$$\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathsf{T}} p(\mathbf{y}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})\big\{\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x})(\mathbf{y}-\mathbf{A}\mathbf{x})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}\big\}, \tag{2.108}$$

the first step is shown in (2.109).

$$\nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathsf{T}} \log p(\mathbf{r}|\mathbf{x}) = \underbrace{\frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{\mathcal{Q}^{-1}(\mathbf{r})} \nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathsf{T}} p(\mathbf{y}|\mathbf{x})d\mathbf{y}}_{T_1} + \underbrace{\nabla_{\mathbf{x}}\Big\{\frac{1}{p(\mathbf{r}|\mathbf{x})}\Big\}\int_{\mathcal{Q}^{-1}(\mathbf{r})} \nabla_{\mathbf{x}}^{\mathsf{T}} p(\mathbf{y}|\mathbf{x})d\mathbf{y}}_{T_2} \tag{2.109}$$

For T_1, we have

$$\frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{\mathcal{Q}^{-1}(\mathbf{r})} \nabla_{\mathbf{x}}\nabla_{\mathbf{x}}^{\mathsf{T}} p(\mathbf{y}|\mathbf{x})d\mathbf{y} \tag{2.110}$$

$$= \frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{\mathcal{Q}^{-1}(\mathbf{r})} p(\mathbf{y}|\mathbf{x})\big\{\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{A}\mathbf{x})(\mathbf{y}-\mathbf{A}\mathbf{x})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}\big\}d\mathbf{y} \tag{2.111}$$

$$= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{y}|\mathbf{r},\mathbf{x})}[(\mathbf{y}-\mathbf{A}\mathbf{x})(\mathbf{y}-\mathbf{A}\mathbf{x})^{\mathsf{T}}]\boldsymbol{\Sigma}^{-1}\mathbf{A} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A} \tag{2.112}$$

$$= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}\mathbf{w}^{\mathsf{T}}]\boldsymbol{\Sigma}^{-1}\mathbf{A} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A} \tag{2.113}$$

$$= \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r})\boldsymbol{\Sigma}^{-1}\mathbf{A} - \mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}, \tag{2.114}$$

for T_2,

$$\nabla_{\mathbf{x}}\Big\{\frac{1}{p(\mathbf{r}|\mathbf{x})}\Big\}\int_{\mathcal{Q}^{-1}(\mathbf{r})} \nabla_{\mathbf{x}}^{\mathsf{T}} p(\mathbf{y}|\mathbf{x})d\mathbf{y} \tag{2.115}$$

$$= -1\cdot\frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{\mathcal{Q}^{-1}(\mathbf{r})} \nabla_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})d\mathbf{y}\cdot\frac{1}{p(\mathbf{r}|\mathbf{x})}\int_{R} \nabla_{\mathbf{x}}^{\mathsf{T}} p(\mathbf{y}|\mathbf{x})d\mathbf{y} \tag{2.116}$$

$$= -1\cdot\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}]\,\mathbb{E}_{p(\mathbf{w}|\mathbf{r},\mathbf{x})}[\mathbf{w}^{\mathsf{T}}]\boldsymbol{\Sigma}^{-1}\mathbf{A} \tag{2.117}$$

$$= -1\cdot\mathbf{A}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\overline{\mathbf{w}}(\mathbf{r})\overline{\mathbf{w}}(\mathbf{r})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{A}. \tag{2.118}$$

Combining T_1 and T_2, we have

$$\mathsf{D}^2_{\mathbf{x}} \log p(\mathbf{r}|\mathbf{x}) = \mathbf{A}^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \big[\mathbf{R}_{\mathbf{w}|\mathbf{r}}(\mathbf{r}) - \overline{\mathbf{w}}(\mathbf{r}) \overline{\mathbf{w}}(\mathbf{r})^{\mathsf{T}} - \boldsymbol{\Sigma} \big] \boldsymbol{\Sigma}^{-1} \mathbf{A}. \tag{2.119}$$

2.8.3 Noise variance estimation

We provide a detailed discussion of variance estimation with 1-bit ADCs and an adjustable noise dither. Consider a block based transmission scheme where the first portion are pilots symbols and the rest are data symbols. Assume for simplicity the pilots are used for channel estimation only. We split the pilots into two sub-blocks such that we inject dither with known but differing variances for each sub-block. For simplicity we set $\sigma^2_{d1} = 0$ and $\sigma^2_{d2} = 1$. We now have two solutions and two unknowns $(\mathbf{x}, \sigma^2_w)$. Denote $\widehat{\mathbf{x}}_1$ and $\widehat{\mathbf{x}}_2$ the solutions to the EM algorithm when the combined marginal noise variance are σ^2_w and $\sigma^2_w + 1$ respectively. Let $\widehat{\mathbf{x}}_0$ be the variance adjusted estimate. Then

$$\widehat{\mathbf{x}}_1 = \frac{\widehat{\mathbf{x}}_0}{\sigma_w}, \quad \widehat{\mathbf{x}}_2 = \frac{\widehat{\mathbf{x}}_0}{(\sigma^2_w + 1)^{\frac{1}{2}}}. \tag{2.133}$$

The ratio of the squared norms is

$$R(\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2) = \frac{\|\widehat{\mathbf{x}}_2\|^2}{\|\widehat{\mathbf{x}}_1\|^2} = \frac{\sigma^2_w}{\sigma^2_w + 1}. \tag{2.134}$$

The noise variance can be recovered via

$$\widehat{\sigma}^2_w = g(\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2) = \frac{R}{1 - R}. \tag{2.135}$$

Since σ^2_w is a function of $\mathbf{x}_1$ and $\mathbf{x}_2$, the estimate $\widehat{\sigma}^2_w$ inherits the consistency property of the constituent ML solutions as a consequence of functional invariance. This simplifies the algorithms because we can perform variance adjustment in a post-hoc manner.

Algorithm 3: Data Augmentation Posterior Sampler

Input: 1-bit outputs $\mathbf{r} = [\mathbf{r}_1; \mathbf{r}_2]$, $\mathbf{r}_d = [r_{d1}, \ldots, r_{dN}]^\mathsf{T}$, design matrix $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]^\mathsf{T}$, noise covariance $\boldsymbol{\Sigma}_\mathbf{w} = \sigma_w^2 \boldsymbol{\Sigma}$, $\boldsymbol{\Sigma} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$

States: Latent variables $\mathbf{y} = [\mathbf{y}_1; \mathbf{y}_2]$, $\mathbf{y}_d = [y_{d1}, \ldots, y_{dN}]^\mathsf{T}$,$\mathbf{w} = [\mathbf{w}_1; \mathbf{w}_2]$, $\mathbf{w}_d = [w_{d1}, \ldots, w_{dN}]^\mathsf{T}$, bounds $B_{di} = [a_{di}, b_{di}]$

Initialize $\mathbf{x}^{(1)} = \mathbf{0}$, $\mathbf{w}^{(1)} = \mathbf{0}$

for $n := 1$ *to* N **do**

% Generate $\mathbf{y}|\mathbf{r}, \mathbf{x}^{(n)}$

For each i and $d = 1, 2$, compute bounds for $\mathbf{w}^{(n+1)}$

$$B_{di} = \mathcal{Q}^{-1}(r_{di}) - \mathbf{a}_i^\mathsf{T} \mathbf{x}^{(n)} \tag{2.120}$$

Ensure $\mathbf{w}^{(n)}$ are within bounds

$$w_{di}^{(n)} = \max(a_{di}, w_{di}^{(n)}) \tag{2.121}$$

$$w_{di}^{(n)} = \min(b_{di}, w_{di}^{(n)}) \tag{2.122}$$

% Generate $S_i \mid w_{1i}^{(n)}, w_{2i}^{(n)}$

Draw $z_i \sim \exp(1)$

Assign $s_i^{(n+1)} = 2(z_i + e_i^{\mathsf{lo}})$ where

$$e_i^{\mathsf{lo}} = [(w_{1i}^{(n)})^2 - 2\rho w_{1i}^{(n)} w_{2i}^{(n)} + (w_{2i}^{(n)})^2]/[2(1-\rho^2)] \tag{2.123}$$

% Generate $W_{1i} \mid w_{2i}^{(n)}, s_i^{(n+1)}$

Draw $w_{1i}^{(n+1)} \sim \mathcal{U}(u_i^{\mathsf{lo}}, u_i^{\mathsf{hi}})$, where

$$u_i^{\mathsf{lo}} = \max\big(a_{1i}, w_{2i}^{(n)}\rho - \sqrt{(s_i^{(n+1)} - (w_{2i}^{(n)})^2)(1-\rho^2)}\big) \tag{2.124}$$

$$u_i^{\mathsf{hi}} = \min\big(b_{1i}, w_{2i}^{(n)}\rho + \sqrt{(s_i^{(n+1)} - (w_{2i}^{(n)})^2)(1-\rho^2)}\big) \tag{2.125}$$

% Generate $W_{2i} \mid w_{1i}^{(n+1)}, s_i^{(n+1)}$

Draw $w_{2i}^{(n+1)} \sim \mathcal{U}(u_i^{\mathsf{lo}}, u_i^{\mathsf{hi}})$ where

$$u_i^{\mathsf{lo}} = \max\big(a_{2i}, w_{1i}^{(n+1)}\rho - \sqrt{(s_i^{(n+1)} - (w_{1i}^{(n+1)})^2)(1-\rho^2)}\big) \tag{2.126}$$

$$u_i^{\mathsf{hi}} = \min\big(b_{2i}, w_{1i}^{(n+1)}\rho + \sqrt{(s_i^{(n+1)} - (w_{1i}^{(n+1)})^2)(1-\rho^2)}\big) \tag{2.127}$$

Set $y_{di}^{(n+1)} = \mathbf{a}_i^\mathsf{T} \mathbf{x}^{(n)} + w_{di}^{(n+1)}$

% Generate $\mathbf{x}|\mathbf{y}^{(n+1)}$

Draw $\mathbf{x}^{(n+1)} \sim \mathcal{N}(\widehat{\mathbf{x}}^{(n+1)}, \boldsymbol{\Sigma}_{\widehat{\mathbf{x}}})$ where

$$\widehat{\mathbf{x}}^{(n+1)} = (\mathbf{A}_*^\mathsf{T} \boldsymbol{\Sigma}_*^{-1} \mathbf{A}_*)^{-1} \mathbf{A}_*^\mathsf{T} \boldsymbol{\Sigma}_* \, \mathbf{y}_*^{(n+1)} \tag{2.128}$$

$$\boldsymbol{\Sigma}_{\widehat{\mathbf{x}}} = (\mathbf{A}_*^\mathsf{T} \boldsymbol{\Sigma}_*^{-1} \mathbf{A}_*)^{-1} \tag{2.129}$$

$$\boldsymbol{\Sigma}_* = \operatorname{diag}\{\boldsymbol{\Sigma}\}_1^N \tag{2.130}$$

$$\mathbf{A}_* = [\mathbf{a}_{*1}, \mathbf{a}_{*2}, \ldots, \mathbf{a}_{*N}]^\mathsf{T} \tag{2.131}$$

$$\mathbf{y}_* = [\mathbf{y}_{*1}, \mathbf{y}_{*2}, \ldots, \mathbf{y}_{*N}]^\mathsf{T} \tag{2.132}$$

end

Estimate $\widehat{\mathbf{x}} = \text{mean}(\{\mathbf{x}^{(n)}, n \geq C\})$

Set $\widehat{\mathbf{x}}^* = \sigma_w \widehat{\mathbf{x}}$

2.8.4 Evaluation of truncated multivariate Gaussians

We provide a modified version of Damien and Walker's algorithm [80]. The cdf inversion technique for truncated exponentials suffers from numerical issues when the lower bound reaches the machine precision of $\log(1 - F_x(a))$ (e.g. $a > 37$ for double precision types). We instead observe that for any $x \sim \exp_{\text{trunc.}}(a, \infty)$ (truncated exponential with $x \in [a, \infty)$), the density can be written as

$$p(x) = Z(a) \exp(-x) \mathbb{1}(x > a) \tag{2.136}$$

$$= Z(a) \exp(-a) \exp(-y) \mathbb{1}(y > 0) \tag{2.137}$$

$$\propto \exp(-y) \mathbb{1}(y > 0), \quad \text{with } x = y + a \tag{2.138}$$

where $y \sim \exp(1)$ is a standard exponential. This amounts to simulating a standard exponential and translating each sample by a. See algorithm 4.

Algorithm 4: Truncated Bivariate Normal Gibbs Sampler

Input: correlation coefficient ρ, bounds $B_i = [a_i\, b_i], i = 1, 2$

Initialize $x_1^{(1)}, x_2^{(1)}$

for $n := 1$ *to* N **do**

`% Generate` $Y \mid x_1^{(n)}, x_2^{(n)}$

Draw $z \sim \exp(1)$

Set $y^{(n+1)} = 2(z + e^{\mathsf{lo}})$ where

$$e^{\mathsf{lo}} = [(x_1^{(n)})^2 - 2\rho x_1^{(n)} x_2^{(n)} + (x_2^{(n)})^2]/[2(1-\rho^2)] \tag{2.139}$$

`% Generate` $X_1 \mid x_2^{(n)}, y^{(n+1)}$

Draw $x_1^{(n+1)} \sim \mathcal{U}(u^{\mathsf{lo}}, u^{\mathsf{hi}})$, where

$$u^{\mathsf{lo}} = \max\big(a_1, x_2^{(n)}\rho - \sqrt{(y^{(n+1)} - (x_2^{(n)})^2)(1-\rho^2)}\big) \tag{2.140}$$

$$u^{\mathsf{hi}} = \min\big(b_1, x_2^{(n)}\rho + \sqrt{(y^{(n+1)} - (x_2^{(n)})^2)(1-\rho^2)}\big) \tag{2.141}$$

`% Generate` $X_2 \mid x_1^{(n+1)}, y^{(n+1)}$

Draw $x_2^{(n+1)} \sim \mathcal{U}(u^{\mathsf{lo}}, u^{\mathsf{hi}})$ where

$$u^{\mathsf{lo}} = \max\big(a_2, x_1^{(n+1)}\rho - \sqrt{(y^{(n+1)} - (x_1^{(n+1)})^2)(1-\rho^2)}\big) \tag{2.142}$$

$$u^{\mathsf{hi}} = \min\big(b_2, x_1^{(n+1)}\rho + \sqrt{(y^{(n+1)} - (x_1^{(n+1)})^2)(1-\rho^2)}\big) \tag{2.143}$$

end

Chapter 3

Feed Forward Type Neural 1-bit Receivers

One-bit analog-to-digital converters (ADCs) has the potential to reduce complexity and lower power consumption of massive multiple-input multiple-output (MIMO) communication systems. However, the severe distortion from 1-bit quantizers gives rise to a highly nonlinear model. In this chapter, we investigate the use of feed forward networks to learn this nonlinear relationship purely from data. We develop feed forward neural network (FFNN) based soft-detectors and provide an analysis of the performance these detectors, address and partially resolve the issues that prevent the networks from reaching the ideal maximum likelihood (ML) performance limit. Next we turn to the simpler symbol detection problem and assess the performance issues by analyzing the performance of several state of the art FFNN architectures. We demonstrate that the performance limit cannot be overcome by modern deep learning architectures and training techniques, and provide possible explanations to this behavior.

3.1 Introduction

Massive multiple-input multiple-output (MIMO) is a technology that promises to bring many fold increase in spectral efficiency through the use a large number of base station antennas. The consequence is that a large number of radio-frequency (RF) chains and analog components are required to support these antennas. The use of 1-bit analog-to-digital converters (ADCs) can simplify the RF-chain significantly by eliminating the need for automatic gain control (AGC). A 1-bit ADC also uses significantly less power compared to a high resolution ADC [5]. However, since the quantizer retains only the sign bit of each component of the receive signal, the resulting distortion is extremely severe.

The universal approximation capability of neural networks makes them an attractive candidate for learning the highly nonlinear input-output relationship. In this work, we investigate training feed-forward networks to learn the nonlinear model purely from data.

We report on the performance of soft-detectors constructed using feed forward neural networks, describe the techniques used and propose solutions to address training issues. Specifically, we reveal that learning of the LLR is impeded by vanishing gradients for large LLR magnitudes and provide a method to stabilize the stochasticity of the gradient. We evaluated a parameter efficient method, feature-wise linear modulation (FLM), to learn output distributional properties depending on a conditioning input.

In the second half of the report, we take a step back to analyze the simpler problem of symbol detection and show empirically that there is a limit to what can be achieved using existing feed-forward architectures.

3.2 Background

Much of the work on deep neural network (DNN) receiver are based on iterative algorithms and the resemblance of a DNN to unfolding of an iterative algorithm [49].

DetNet [50] was inspired by unfolding the projected gradient descent algorithm. A novel contribution in the paper is the observation that every layer produce an estimate of $\mathbf{x}$, thus a loss function is proposed to incorporate the loss due to the inaccuracy in each layer's output. Unlike AMP-based algorithms, the performance of DetNet is shown not to degrade in highly correlated channels. In [81], the authors developed a deep learning receiver based on the OAMP algorithm. The OAMP requires that the sensing matrix $\mathbf{H}$ be unitarily invariant to reach Bayes-optimality, the OAMP-Net introduces learnable parameters that allow the model to be more flexible. However, like any AMP variant, the performance of this receiver degrades substantially for correlated channels.

Khani et al. [82] proposed an online deep learning detector (MMNet) that balances the flexibility and the rigidity of DetNet and OAMP-Net respectively, reaching within 1.5 dB from the performance of ML. A subsequent work by Goutay et al. [83] explores the use of a hypernetwork to learn the parameters of MMNet offline to eliminate online learning altogether.

The works above considered large dimensional symbol detection with full precision unquantized inputs. For quantized input receivers, a one-bit receiver inspired by sphere decoding is developed in [84]. While a sphere decoder is less complex compared to ML, the non-determinism of the algorithm makes it unappealing for practical implementation. In [45], the authors considered an encoder-decoder architecture for end-to-end 1-bit OFDM precoder-receiver design. None of the these considered computing the soft decisions, an important component necessary in modern communication systems.

The nonlinearity of the soft-demapper is considered in [48]. The author proposed a shallow neural network based soft-demapper to address the inaccuracies of the piecewise

linear, max-log approximation common in existing implementations.

None of the work considered the joint symbol detection and soft demapping as a nonlinear estimation problem, and most importantly the implication of performance degradation with model mismatch.

3.2.1 Relation to sphere decoder

It is worthwhile to take a different viewpoint when a deep neural network is employed. In the most usual case, one invokes the argument that a neural network is an universal function approximator, leading one to assume that this is the only way a neural network can arrive at its answer. By observing how the sphere detector infer the LLR based on the points defined within a neighborhood, it might be a fruitful avenue to view the internal representation of a deep net as an encoding of the neighborhood from which the LLRs are produced. This gives us additional impetus to formulate a neural soft detector in the hope of discovering useful and efficient representation.

3.3 System model

We consider a MIMO uplink channel with N_r receive antennas and N_s single antenna users. The analog receive signal is given by

$$\bar{\mathbf{r}} = \overline{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{n}} \tag{3.1}$$

with source symbol vector $\bar{\mathbf{x}} \in \mathcal{X}^{N_s} \subset \mathbb{C}^{N_s}$, channel $\overline{\mathbf{H}} \in \mathbb{C}^{N_r \times N_s}$, noise vector $\bar{\mathbf{n}} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ and analog receive vector $\bar{\mathbf{r}} \in \mathbb{C}^{N_r}$. $\mathcal{X}$ is a finite M-QAM constellation (symbol space) such that $|\mathcal{X}| = M$. The channel is assumed to be block fading and known at the receiver. The transmitted signal for the kth user is assumed to have unit power constraint

$\mathbb{E}[|\bar{x}_k|^2] = 1$. SNR is defined as $\rho = 1/\sigma^2$. A mapper $f(\mathbf{b})$ defines an invertible mapping from bit vector $\mathbf{b} \in \{0,1\}^{N_b}$ to a point on a M-QAM constellation, with $N_b = \log_2 M$. This mapping associates a bit vector $\mathbf{b}$ to a particular symbol $\bar{x}$. Each symbol in $\bar{\mathbf{x}}$ is mapped from independent source bits using the same mapper function. The distribution of the source bits are equiprobable and the bits are assumed independent. The induced distribution on the symbol space are likewise equiprobable and the symbols independent.

It is convenient to transform the problem into an equivalent real-valued system, using the following notation

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{3.2}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathsf{re}(\overline{\mathbf{H}}) & -\mathsf{im}(\overline{\mathbf{H}}) \\ \mathsf{im}(\overline{\mathbf{H}}) & \mathsf{re}(\overline{\mathbf{H}}) \end{bmatrix} \tag{3.3}$$

and

$$\mathbf{r} = \begin{bmatrix} \mathsf{re}(\bar{\mathbf{r}}) \\ \mathsf{im}(\bar{\mathbf{r}}) \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \mathsf{re}(\bar{\mathbf{x}}) \\ \mathsf{im}(\bar{\mathbf{x}}) \end{bmatrix}, \ \mathbf{n} = \begin{bmatrix} \mathsf{re}(\bar{\mathbf{n}}) \\ \mathsf{im}(\bar{\mathbf{n}}) \end{bmatrix} \tag{3.4}$$

The 1-bit quantized signal is generated by

$$\mathbf{y} = \mathcal{Q}(\mathbf{r}) \tag{3.5}$$

where $\mathcal{Q}$ is an element-wise application of the sign function, $\mathrm{sgn}(\cdot)$, defined by

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{otherwise.} \end{cases} \tag{3.6}$$

3.4 Signal detection

3.4.1 Likelihood function

We begin by presenting the likelihood function used in the subsequent sections. The likelihood of receiving $\mathbf{y}$ given source vector $\mathbf{x}$ for a 1-bit receiver is given by [56]

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{2N_r} \Phi\Big(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathsf{T}} \mathbf{x}\Big) \tag{3.7}$$

where $\Phi(x)$ is the standard normal cdf, y_i and $\mathbf{h}_i$ are the ith element and ith row of the receiver vector and the channel matrix resp. We remark here that unlike the full-precision case in [50], we expect a competitive algorithm to depend on the noise variance input to achieve ML like performance over the full SNR range considered.

3.4.2 Soft detection

The soft detection of each bit requires the computation of the log-likelihood ratio (LLR): this forms the prior to be used by the decoding algorithm present in most communication systems. The LLR corresponding the kth bit in the sth symbol is defined as

$$L_{s,k}(\mathbf{y}) = \log\Big(\frac{p(\mathbf{y}|b_{s,k}=1)}{p(\mathbf{y}|b_{s,k}=0)}\Big) \tag{3.8}$$

$$= \log\Bigg(\frac{\sum_{\mathbf{x}\in X^1_{s,k}} p(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{x}\in X^0_{s,k}} p(\mathbf{y}|\mathbf{x})}\Bigg) \tag{3.9}$$

where $X^b_{s,k}$ is the subset of symbol vectors in which the sth symbol and kth bit has value b. The LLR is a nonlinear function of the likelihood, which itself is nonlinear in $\mathbf{y}$. We can now form a neural network to approximate this function for each s, k combination.

3.4.3 Combined symbol estimator and soft demapper

Alternatively, a soft detector can be implemented as a two stage process. The first stage consists of a symbol estimator that maps received symbol vector $\mathbf{y}$ to an estimate $\widehat{\mathbf{x}} \in \mathbb{C}^{N_s}$. A soft-demapper takes as input the effective channel

$$\mathbf{x} \to \mathbf{y} \to \widehat{\mathbf{x}} \tag{3.10}$$

and computes the LLRs based on the likelihood $p(\widehat{\mathbf{x}}|\mathbf{x})$.

3.5 Deep learning receiver

In this section, we begin the exploration of the neural network model and training objective for the soft detector design.

3.5.1 Variational information maximization

We formulate the information maximization problem for a variational decoder. As shown in [85], we have a variational lower bound I_{BA}:

$$I(\mathbf{x};\mathbf{y}) \equiv H(\mathbf{x}) - H(\mathbf{x}|\mathbf{y}) \tag{3.11}$$

$$\geq H(\mathbf{x}) + \sum_{\mathbf{x},\mathbf{y}} p(\mathbf{y},\mathbf{x}) \log q(\mathbf{x}|\mathbf{y}) \triangleq I_{BA} \tag{3.12}$$

where $H(\mathbf{x})$ is trivially $N_s N_b$ bits. This bound is exact if $q(\mathbf{x}|\mathbf{y}) = p(\mathbf{x}|\mathbf{y})$. This maximization is equivalent to the minimization of the KL distance $\mathbb{E}_{p(\mathbf{y})}[D_{\mathsf{KL}}(p(\mathbf{x}|\mathbf{y}) \,\|\, q(\mathbf{x}|\mathbf{y}))]$. Note that all quantities are conditioned on the channel $\mathbf{H}$. The $\log(\cdot)$ function is base-2 unless stated otherwise.

Remark 5. The variational lower bound I_{BA} can be used as a metric to monitor the

performance of the neural network during training; the upper bound can be computed from exact LLR computation, the cross entropy term on the RHS is the loss function that is computed during training.

3.5.2 Relation to cross entropy learning

Since $\mathbf{x} \leftrightarrow \mathbf{b}$ is a one-to-one correspondence, we can consider the equivalent Infomax problem with $\mathbf{b}$ as the source variable. Now consider the factored variational distribution:

$$q(\mathbf{b}|\mathbf{y}) \triangleq \prod_{k=1}^{N_b} q_k(b_k|\mathbf{y}). \tag{3.13}$$

Thus we seek to maximize (assuming the dimension of $\mathbf{b}$ is known):

$$\sum_{\mathbf{b},\mathbf{y}} p(\mathbf{b},\mathbf{y}) \sum_k \log q_k(b_k|\mathbf{y}) \tag{3.14}$$

$$= \sum_{\mathbf{b},\mathbf{y}} p(\mathbf{b},\mathbf{y}) \sum_k \big[\, b_k \log \sigma(f_k(\mathbf{y})) + (1-b_k) \log\big(1-\sigma(f_k(\mathbf{y}))\big) \,\big] \tag{3.15}$$

where $\sigma(x) = 1/(1+\exp(-x))$ is the logistic sigmoid. The logit function $f_k(\mathbf{y})$ is modeled using a deep neural network. We recognize here this problem is equivalent to binary cross entropy learning jointly across batch samples and all bits in the symbol.

3.5.3 Equivalence of logits to LLRs

It is important to note that the main goal of this problem is to compute the LLRs. Can we accomplish this by learning the posterior via cross entropy learning? A close examination of the logits output of the DNN reveals that they are exactly the LLRs if the optimization of the above objective leads to the convergence of $q(b_k|\mathbf{y})$ to $p(b_k|\mathbf{y})$. This

follows from Bayes rule [86]:

$$p(b=1|\mathbf{y}) = \frac{p(\mathbf{y}|b=1)}{p(\mathbf{y}|b=1)+p(\mathbf{y}|b=0)} \tag{3.16}$$

$$= \frac{1}{1+\exp\big(-L(\mathbf{y})\big)} \tag{3.17}$$

where

$$L(\mathbf{y}) = \log\Big(\frac{p(\mathbf{y}|b=1)}{p(\mathbf{y}|b=0)}\Big) \tag{3.18}$$

3.5.4 BICM scheme

This section provides theoretical justification that the factored distribution (3.13) is a suitable metric for use in modern communication systems.

Practical communication systems utilize the bit-interleaved coded modulation (BICM) scheme [87] for bit decoding. The BICM scheme is also called bit metric decoding (BMD) in [88]. In a BICM decoder, the N_b bits in a symbol are treated as independent and a symbol metric proportional to the product aposteriori marginals $p(b_k|\mathbf{y})$ [89] is used. Precisely, we define the metric

$$q(\mathbf{x},\mathbf{y}) = \prod_{k=1}^{N_b} q_k(b_k(\mathbf{x}),\mathbf{y}). \tag{3.19}$$

In particular, if the kth bit decoding metric q_k is proportional to the kth posterior bit probability, then the BICM decoder can be shown to achieve BICM capacity for a memoryless channel [90, Theorem 1].

Remark 6. For the special case of uniform input, the capacity achieving kth bit metric can be obtained by any function that is proportional to the marginalized kth bit channel transition probability.

Clearly, the bit metric is a suboptimal metric compared to the general metric

$q(\mathbf{x}, \mathbf{y})$, and the achievable rate of the BICM decoder is upper bounded by the coded modulation (CM) scheme [91], which does not have such restriction. However, the BICM scheme is appealing since it enables the decoupling of the demodulator and the decoder while retaining respectable performance, especially when gray labeling [92] and iterative decoding are used [93, 94].

The achievable rate for the BICM decoder is

$$R = H(\boldsymbol{B}) - \sum_{k=1}^{N_b} H(B_k|Y) \tag{3.20}$$

We also note here that for independent information bits $B_1, B_2, ..., B_m$, the achievable rate can be written as

$$R = \sum_{k=1}^{m} I(B_j; Y). \tag{3.21}$$

This result can be extended to a memoryless channel with i.i.d. channel state parameters $\mathbf{H}$ together with the assumption that B_k and $\mathbf{H}$ are independent. We obtain BICM capacity for perfect CSI [92]

$$R = H(\boldsymbol{B}) - \sum_{k=1}^{N_b} H(B_k|Y, \boldsymbol{H}). \tag{3.22}$$

Using the following cross-entropy inequality

$$\mathbb{E}_{B_k,Y,\boldsymbol{H}}[-\log q_k(b_k|y, \mathbf{H})] \geq \mathbb{E}_{B_k,Y,\boldsymbol{H}}[-\log p(b_k|y, \mathbf{H})] \tag{3.23}$$

$$= H(B_k|Y, \boldsymbol{H}) \tag{3.24}$$

we can learn a suitable metric by minimizing the sum cross-entropy

$$L(\mathbf{q}) = \sum_{k=1}^{N_b} \mathbb{E}_{B_k,Y,\boldsymbol{H}}[q_k(b_k|y, \mathbf{H})]. \tag{3.25}$$

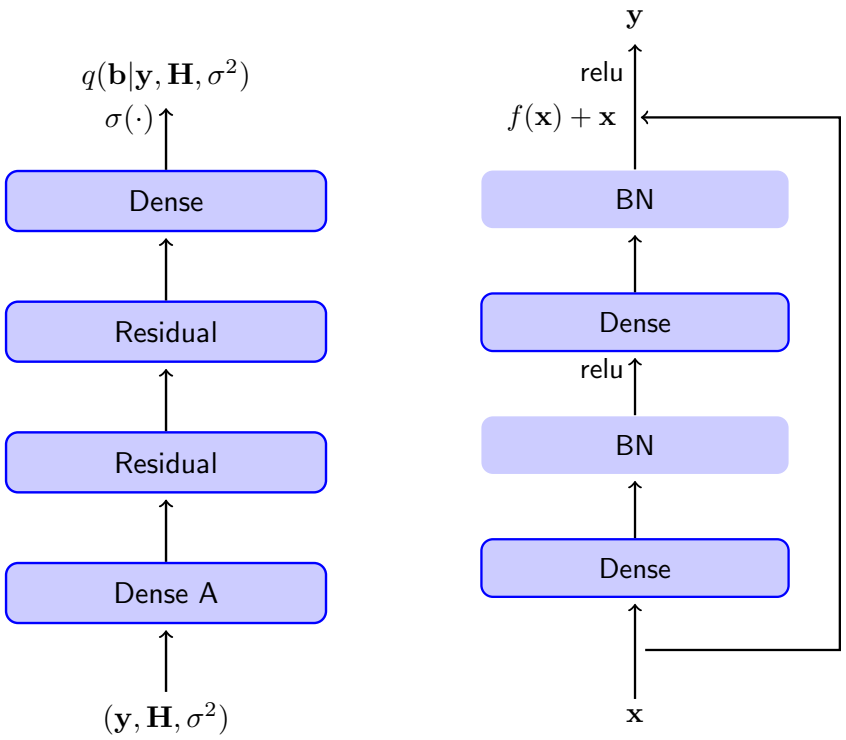
In theory if a neural network is given enough capacity, the sum cross-entropy converges to

the sum conditional entropy on the RHS of (3.22), thus the BICM rate can be attained. We note here that similar analysis were derived in [95].

3.6 Deep learning soft detector

We begin our exploration by asking what the limiting factors to the performance of a deep network are and which of those are main contributors. First we explored the capacity of an DNN model. We conducted experiments on a 2x2 MIMO channel using 16-QAM modulated symbols with unquantized inputs and arrived at the Resnet model [19] with ReLU activations. The topology of the Resnet model is given in Fig. 3.1. The first dense layer (Dense A) lifts the input dimension to the hidden dimension ($N_h = 1024$), which remains the same for all residual blocks. This is necessary for the implementation of identity skip connections in the residual block. Note that Dense A is a composite layer consisting of a full connected layer, batch normalization followed by RELU activations. Each residual block contains two full connected dense layers. The last dense layer converts the internal features into logits for bit prediction. The parameters of the networks are described in Table 3.1. The first row shows the concatenation of complex to real received signal vector and channel matrix. The subsequent rows depict the dimension of the weight matrix in the fully connected layers, ignoring the bias parameters. We also experimented with the revised Resnet model (ResnetV2) [96], however, we will defer the discussion of ResnetV2 to section 3.12.

Based on our experimentation, a 5-layer Resnet is sufficient to reach ML like performance for QPSK.



(a) 5-layer Resnet (b) Residual block

Figure 3.1: Neural soft-detector model implemented as a Resnet

3.6.1 Learning across SNRs

Recall from section 3.4.1 that a soft-detector can attain ML performance over a range of SNRs only if noise statistics information is given. Even when this is known, how should this be used to influence a deep neural network? Note that feeding SNR at the input of the network, we are exhausting the capacity of the network solely to propagate the SNR information. A better strategy is to manipulate the nodes of the network directly, at the same time keeping the scope of manipulation to a minimum.

It can also be observed that the LLR distributions are shifted farther apart at higher SNRs. Thus a deep net must produce an output distribution dependent on this SNR parameter. We investigated conditional neural network architectures that can vary the output distribution with the above efficiency requirement.

Table 3.1: Network architectures for the 6x2 QPSK receiver and the 18x2 16-QAM receiver

layer name	output size	5 layer, QPSK	5 layer, 16QAM
input (y, H, σ^2)	-	(12, 24, 1)	(36, 72, 1)
dense	1024	$[\ 37 \times 1024\ ]$	$[\ 109 \times 1024]$
residual1	1024	$\begin{bmatrix} 1024 \times 1024 \\ 1024 \times 1024 \end{bmatrix}$	$\begin{bmatrix} 1024 \times 1024 \\ 1024 \times 1024 \end{bmatrix}$
residual2	1024	$\begin{bmatrix} 1024 \times 1024 \\ 1024 \times 1024 \end{bmatrix}$	$\begin{bmatrix} 1024 \times 1024 \\ 1024 \times 1024 \end{bmatrix}$
dense (b)	-	$[\ 1024 \times 4\ ]$	$[\ 1024 \times 8]$

This led us to the feature-wise linear modulation technique (FLM) [97], which can be viewed as a generalization of the conditional normalization (CN) methods [98, 99].

These methods can be traced back to the seminal work of Gatys *et al.* [100] on image style transfer. They reasoned that style could be interpreted as a form of texture, which has a stationary cross-correlation across the space of the image. To successfully transfer style from a reference image to a target image, one simply has to match the second order statistics of the features of each layer in a trained convolutional network while keeping the content (high-level features) close to the target. Beyond image stylization, CN has seen success in visual question answering [97, 101] and style-based image generation [102]. The success of CN demonstrates the technique's capability and applicability to other problem domains.

FLM can be interpreted as performing conditional computation [97]: depending on the conditioning input it can selectively activate/deactivate and emphasize/deemphasize nodes in the network.

We conjecture that the distributional properties of the neural network's output are governed predominantly by the statistical properties of the features, leading us to propose

this method for conditional LLR ouput generation. This is similar in spirit to the style-based image generation in [102] where the layers of a generative adversarial network (GAN) are modulated to produce the desire style output.

3.6.2 FLM

Similar to instance normalization methods [98, 99, 103], FLM manipulates the output of a neural network via a learned affine transformation on the network's features dependent on a conditioning input $\mathbf{w}$. Precisely, FLM learns a function

$$(\alpha_c, \beta_c) = f_c(\mathbf{w}),\ c \in \mathcal{C} \tag{3.26}$$

for each feature c in some set $\mathcal{C}$. The output is then used to transform the networks activations z_c such that, we have the transformed output for the cth feature:

$$g_c(z|\alpha, \beta) = \alpha_c z_c + \beta_c. \tag{3.27}$$

We note that the weights that connect the layers within the neural network are independent of the conditioning input. Thus FLM is a highly parameter efficient method of manipulating the network.

The FLM based model is depicted in Fig. 3.2. The model is identical to the one previously described, with the exception that the batch normalization layer is replaced with a FLM layer at the input to the RELU activation functions.

We implement conditioning as a lookup of a particular row on the matrix given one-hot representation $\mathbf{z}$ of some SNR s. Thus there is an independent set of parameters for each input $\mathbf{z}$. Note that this direct encoding scheme is transitive since it cannot infer values that has not been seen during training. We define a vector $\boldsymbol{\alpha}_c$ and $\boldsymbol{\beta}_c$ and select

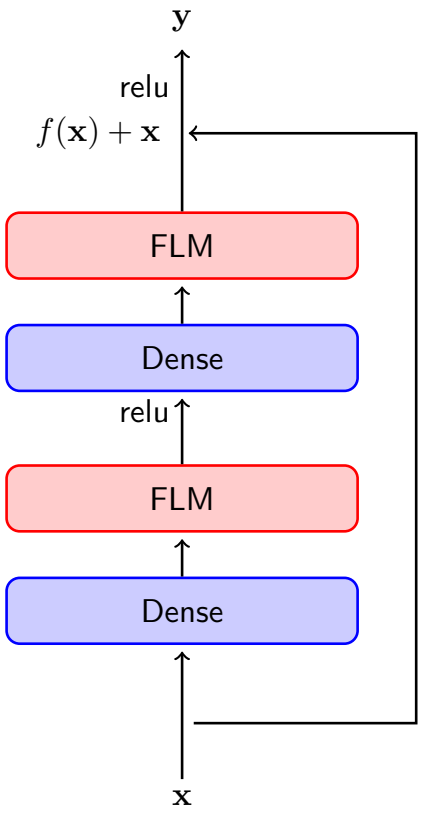


Figure 3.2: Residual block with feature-wise linear modulation

the parameter using a one-hot representation $\mathbf{z}$, such that

$$\alpha_{c,s} = \boldsymbol{\alpha}_c^{\mathsf{T}}\mathbf{z}, \quad \beta_{c,s} = \boldsymbol{\beta}_c^{\mathsf{T}}\mathbf{z}. \tag{3.28}$$

We note here that this method is a generalization of the one-hot encoded input method used in CGAN [104]. To see this, note that a constant input node with full connections to the next layer, layer i, can arbitrarily bias the activations of layer i via its learnable weights. Thus for each conditioning input, it can learn a new set of biases for the ith layer's activations. Contrast this to FLM's ability to scale and bias any nodes in the network, it is clear CGAN is a strict subset. Suppose, we have a conditioning input that is real and this is fed into the DNN at the input layer, the network in this case is even more constrained.

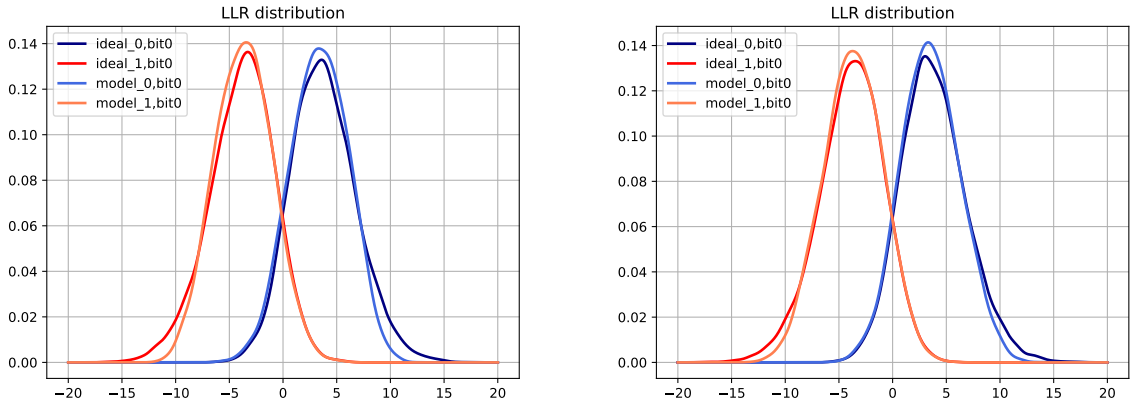
3.7 Model training

In this section, we describe the training procedure and provide an analysis of the LLRs distributions from the trained model to reveal the limitations in training using a logistic sigmoid function and the passing of gradients to the lower layers. We provide two different training schedules for comparison. Note that we are using a model that can match the performance of a 16-QAM receiver, thus the model has much more capacity than is required for this scenario.

In the training phase, random bits are generated and mapped to symbols in the complex domain. For QPSK, a 6x2 MIMO channel is used. For 16-QAM, a 18x2 MIMO channel is used. The mapped symbols are then passed through a Rayleigh channel, corrupted with AWGN noise and 1-bit quantized. The random bit input $\mathbf{b}^{(i)}$, together with the noisy channel output $\mathbf{y}^{(i)}$ form the training sample $D = \{\mathbf{b}^{(i)}, \mathbf{y}^{(i)}\}$ used to train the neural network, evaluated using the joint binary entropy loss (3.15). The network is trained for $\{20, 40\}$ epochs; each epoch contains 5000 minibatches each with $\{256, 1024\}$ samples. Samples are generated on the fly, thus no samples are repeated during training. Cross validation is performed on 1000 minibatches after each epoch. Batch normalization [23] is used at the input of every RELU activations. The Adam optimizer [26] is used with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and initial learning rate 1e-3 that decreases by a factor of 10 using learning rate schedules $\{[10, 15], [10, 25]\}$, the values in the list represent the epoch indices. We did not use any weight decay as we do not observe overfitting, a phenomenon that is more prevalent when samples are repeated.

3.7.1 Limitation due to vanishing gradient

In the first experiment we trained the model at a fixed SNR (2 dB). The results are shown in Figs. 3.3, 3.5. We note that using modern training techniques and a small



(a) 256-batch, 20 epochs

(b) 1024-batch, 40 epochs

Figure 3.3: Comparison of the learned LLR distribution vs the ideal LLR distribution at 2dB conditional on the source bit.

sample (relative to the number of parameters), the model can be trained to reach within 0.5dB performance of an ML detector. Getting past this performance bottleneck requires examining the model output further.

In Fig. 3.3 the learned LLR distribution is compared with the ideal LLR distribution conditional on the value of the source bit. The distribution reveals that, for either bit value, the distribution is faithfully learned on the side that has smaller LLR magnitudes compared to the opposite side. We argue that this asymmetry in learning is due to the vanishingly small gradient values at large LLR magnitudes, noting that the gradient of $\sigma(x)$ decay exponentially with $|x|$. One method that was found to be effective was using a large batch size and allowing model learning more time to settle. We believe that large batches reduced the stochasticity of the computed gradients, permitting a larger weight update because updates are inversely scaled by the average power of the gradients.

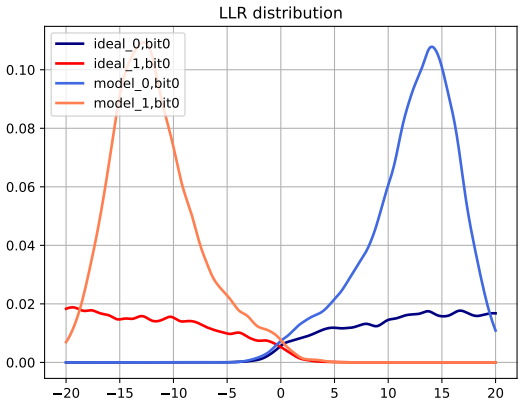


Figure 3.4: Comparison of trained FLM model LLR to ideal LLR distribution at 15dB, conditional on the source bit.

3.7.2 Conditional learning across SNRs

In this setting, the model is trained using data generated with random SNR, sampled from a discrete uniform distribution in log scale (dB). When applicable, the model accepts a one-hot representation of the SNR value, which is then fed to different parts of the neural network.

The true SNR dependent LLR distribution is compared with the learned LLR distribution in Fig. 3.4. The FLMNet can much more accurately separate the LLRs from the respective bits and it is much better able to learn the overlapping region which is critical for predicting mutual information. It can be seen that LLRs with large magnitudes are severely saturated, with most of the density concentrated around $|L| = 15$.

3.8 Performance evaluation

The performance of the neural receiver is compared with the exact LLR computed using (3.9). For BER performance, we also compare the neural receiver to linear receivers with a per stream Gaussian soft-demapper. The receiver is evaluated on the coded BER

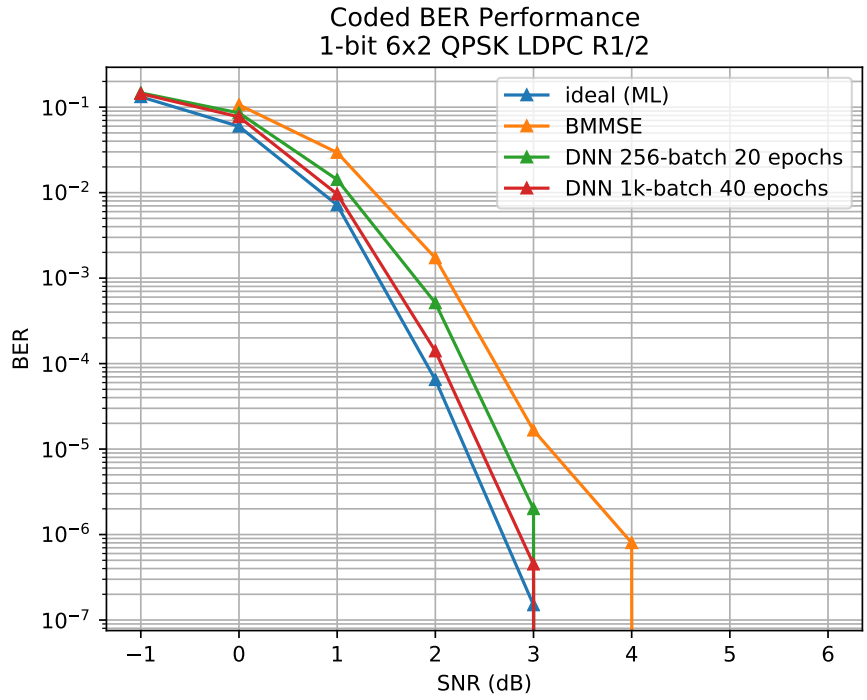


Figure 3.5: Coded BER performance of ideal ML, Bussgang MMSE (BMMSE), and 5-layer Resnet (DNN) models trained with batch size of 256/1024 and 20/40 epochs over a 6x2 Rayleigh channel with QPSK modulated symbols and rate 1/2 LDPC code.

performance using an LDPC code with a code rate of 1/2. We also evaluated the achieved mutual information (MI) across a wide range of SNRs to analyze the adaptability of the DNN to varying noise statistics. For MI estimation, we used a derivative of the KSG algorithm [105] for discrete-continuous variable pairs to compute the individual bit to LLR MI and evaluated the total MI achieved across all bits. For coded BER, LDPC encoder is used to generate the source bits; at the decoding stage an offset min-sum decoder is used for error correction.

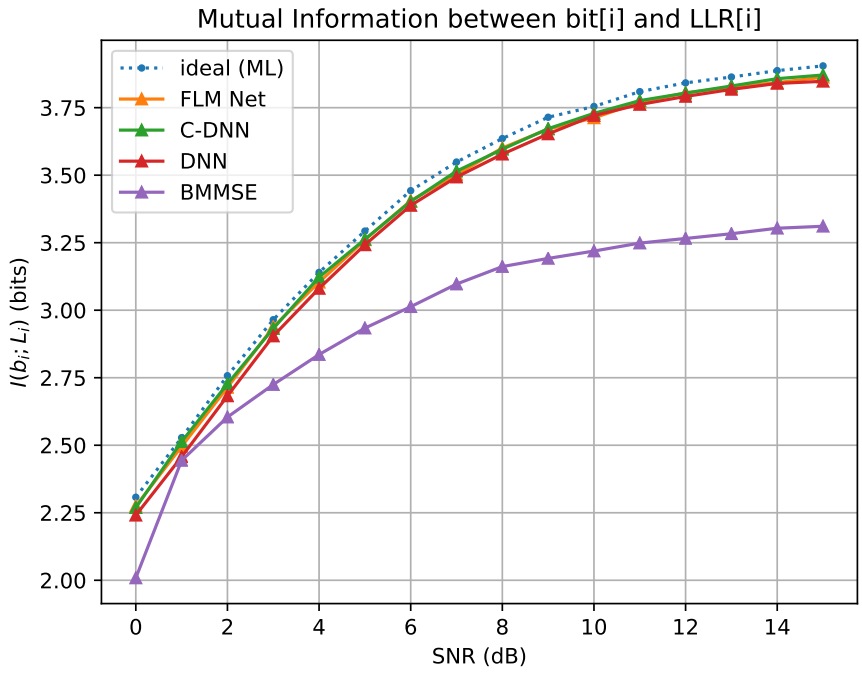


Figure 3.6: Comparison of achieved MI across SNRs for the ideal ML, Bussgang MMSE (BMMSE), conditional DNN (C-DNN) and DNN model with QPSK modulated symbols.

3.9 Numerical Results

3.9.1 QPSK performance

The effect on BER performance is plotted in Fig. 3.5. The DNN performance can approach ML by doubling the baseline minibatch size and doubling the training time with appropriate learning rate schedule. The best known linear receiver, Busgang MMSE, is also plotted for comparison. As noted in section 3.7.1, we believe the performance of the network is negatively effected by the logistic sigmoid function and the vanishing gradient during training.

For QPSK, the achieved mutual information of the FLMNet is compared to a DNN trained with data generated from varying SNRs but has no knowledge about SNR. We also compared it to a conditional DNN with a one-hot representation of the SNR as input,

the results are plotted in Fig. 3.6.

Surprisingly, the DNN trained without noise variance information is already reaching good performance relative to ML and other conditional DNNs. It appears the DNN is able to learn the structure of the channel in the presence of noise. This may be explained by the simple channel (having small channel capacity) in a QPSK system. There is thus little room for improvement from conditioning the DNN with SNR information.

3.9.2 16QAM performance

Next, we compare the 16-QAM performance of various deep net architectures.

We experimented with Resnet [19] and the revised Resnet [96] architectures, varying the depth of the deep net, and considered various loss functions that include the cross-entropy, MSE and exponential loss. The BER performance comparing a 7-layer Resnet (DNN) to ML and linear receiver models is shown in Fig. 3.7. In terms of BER, the Resnet performed better compared to the linear receiver. However, we observed that the Resnet performance quickly saturated and the gap between the Resnet and ML widened at higher SNR.

Examining the achievable MI in Fig. 3.8, a large gap was observed at high SNR. The performance of all these deep networks saturated beyond 15dB without exception. We did not observe any improvement even as we increased the network depth to 13. MSE loss (DNN,MSE) produced worse performance compared to cross-entropy loss (DNN,XENT). The exponential loss (ResnetV2, Exp), reported as a suitable loss function for likelihood ratio estimation [106], could not outperform entropy-loss (ResnetV2, XENT). We also experimented with FLMNet in this setting, however it did not close the gap to ML. It showed the same saturation behavior in high SNR.

We next analyze the conditional LLR distribution to determine the deep network's ability to learn the statistics of the bit LLRs, shown in Fig. 3.9. We note here that the true

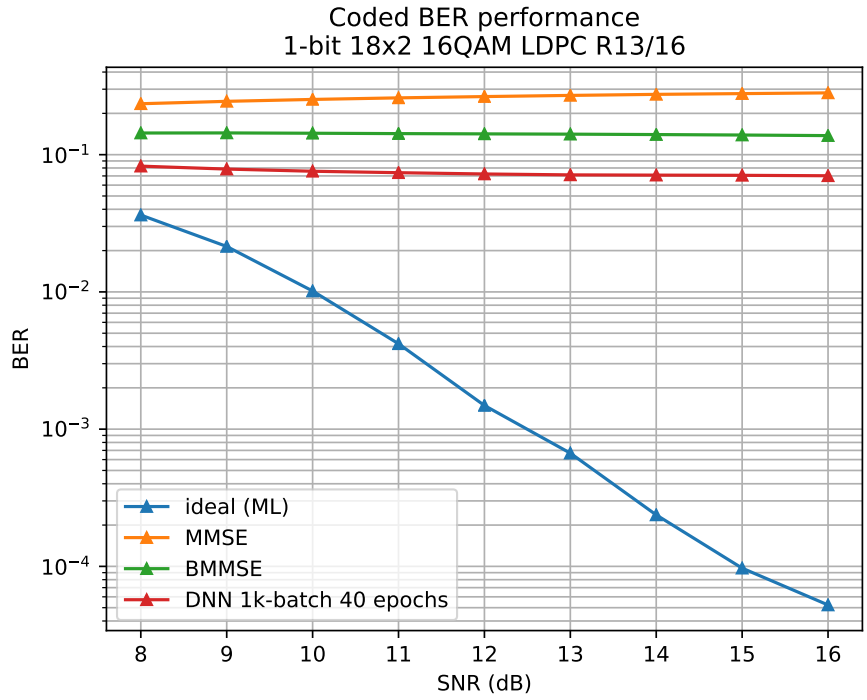


Figure 3.7: Coded BER performance of ideal ML, MMSE, BMMSE and DNN models over a 18x2 Rayleigh channel with 16QAM symbols and rate 13/16 LDPC code.

LLR distribution for bit 1 takes on a much more complex form. The Resnet has difficulty learning the distribution. The fact that we formed an approximate factored distribution may not allow the network to exploit the correlation between bit0 and bit1 at the output of the channel (they are modulated on the same component in the complex plane).

3.10 Conclusion for LLR estimation

We explored the use of feed-forward dense network architecture to the problem of LLR estimation in a system with 1-bit quantized received signals. For the simple QPSK case, a Resnet with sufficient capacity can achieve ML like performance by increasing the batch size, increasing training time and a well tuned learning schedule.

In 16QAM, a significant gap is observed with respect to coded BER, achievable MI

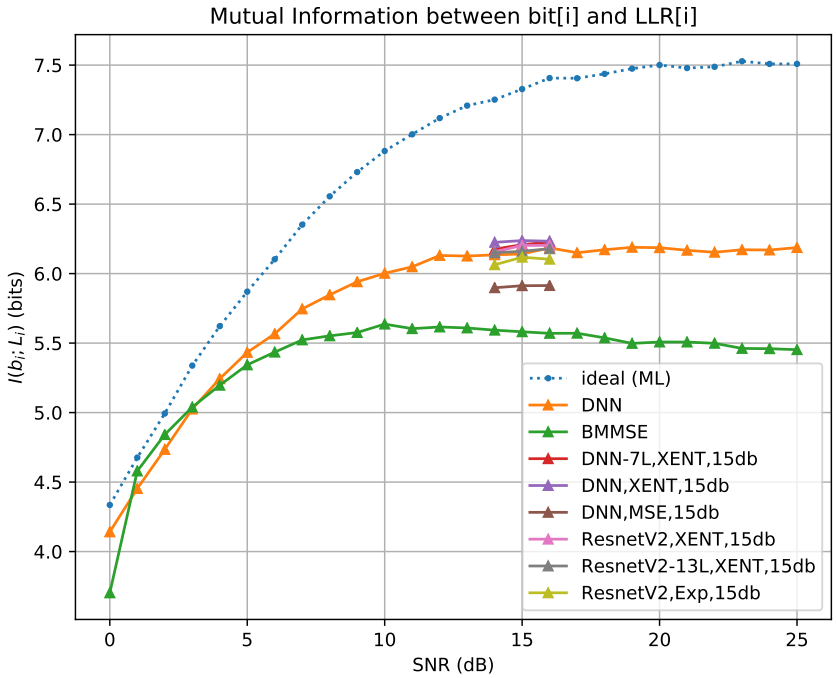
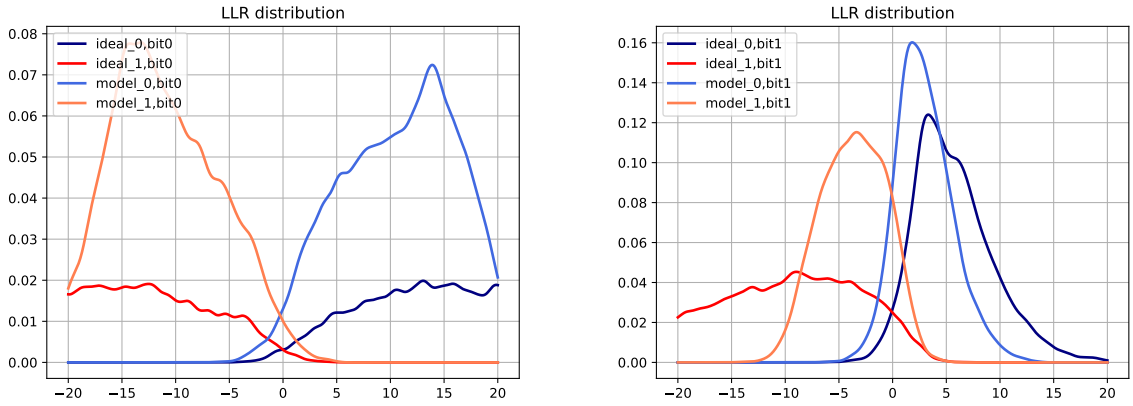


Figure 3.8: Comparison of achieved MI for Resnets (DNN), revised Resnets (ResnetV2) over a 18x2 Rayleigh channel with 16-QAM modulated symbols.

and learned LLR distribution, irrespective of the training techniques and the version of Resnet used. Based on our analysis, one possible issue is that we are performing approximate inference using a factored distribution of the bits, which does not allow the deep network to learn the correlation adjacent bits experienced in the complex channel. Another major factor is that none of these architecture incorporate the generative model, or any relevant inductive bias to assist it, making the learning task difficult. It is also a fundamental issue from the beginning that a 1-bit receiver discards much of the information that would otherwise be preserved in a full-precision (unquantized) receiver.

In the sequel, we take a step back to analyze a simpler problem of symbol detection and hope to identify the key contributors that impeded learning in this class of networks.



(a) LLR distribution, bit0

(b) LLR distribution, bit1

Figure 3.9: Comparison of trained model LLR to ideal LLR distribution at 15dB conditional on the source bit, 1-bit 16QAM case.

3.11 Symbol detection

We proceed to evaluate the performance of various deep net architectures in the symbol detection problem.

Recall from section 3.4.1 that the likelihood is

$$p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{2N_r} \Phi\Big(\sqrt{2\rho} y_i \mathbf{h}_i^{\mathsf{T}} \mathbf{x}\Big). \tag{3.29}$$

The symbol detection problem amounts to finding a candidate symbol that maximizes the likelihood, i.e.

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{X}} p(\mathbf{y}|\mathbf{x}). \tag{3.30}$$

3.11.1 Deep network model

Since the symbol alphabet is finite and discrete, we can construct a deep net that outputs the symbol probability via a dense softmax layer. We note that this does not scale well to high order modulation and large number of users, but it suffices for our current

investigation into the issue of learning. Precisely, the deep net forms an approximate posterior of the form

$$q(\mathbf{x}|\mathbf{y},\mathbf{H}) = \sigma(f(\mathbf{y},\mathbf{H})) \tag{3.31}$$

where σ denotes the softmax function, the ith output is given by

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N_x} e_j^z}, \quad \forall i \in [N_x] \tag{3.32}$$

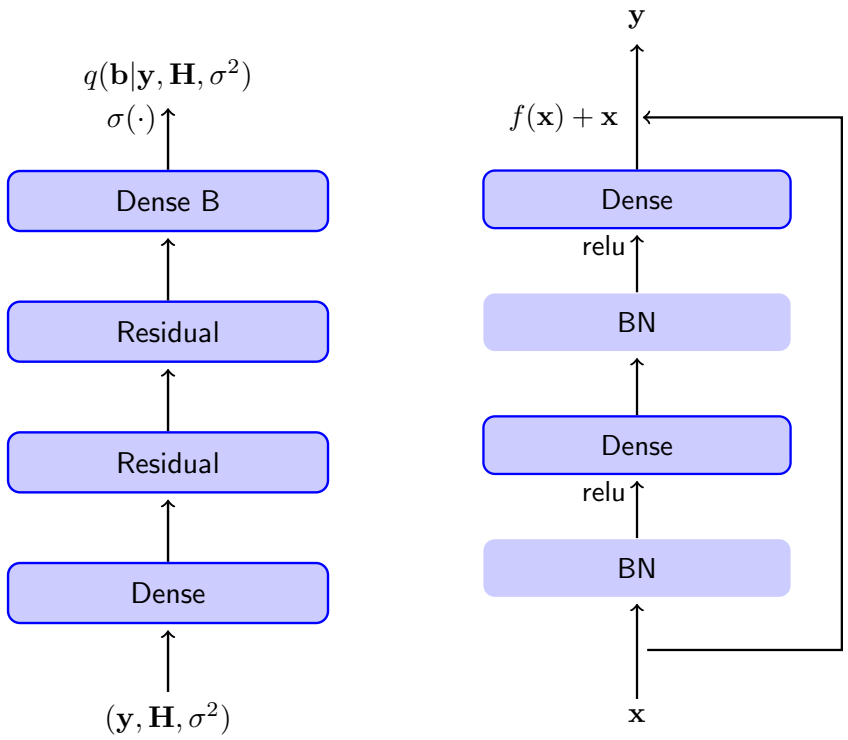
and $f(\mathbf{y},\mathbf{H})$ is any deep network chosen for the problem. We remark here that this does not require an approximate factored posterior as devised in the first part of this work, eliminating this potential performance bottleneck in our analysis.

3.12 Network architectures

3.12.1 Resnets

As in the previous exploration, we experimented with the Resnet and revised Resnet (ResnetV2). The Resnet architecture was described in section 3.6. The only difference is that the output now is the posterior of the symbol $p_\theta(\mathbf{x}|\mathbf{y})$, produced by a softmax function at the last stage, as opposed to the bit posterior computed in LLR estimation. Here we give a brief overview of the revised Resnet model. ResnetV2 was proposed in [96] to provide a direct unimpeded path to all residual blocks. This substantially improved the gradient flow from the output layer to any intermediate layer, allowing for faster and more stable learning of ultra deep networks.

The ResnetV2 architecture is shown in Fig. 3.10. The residual block now performs *pre-activation* in the nonlinear branch. Batch normalization and RELU activations are applied before a fully connected layer. Following the pre-activation approach, the last layer (Dense B) of the revised Resnet is modified accordingly such that batch normalization,



(a) 5-layer revised Resnet (b) Revised residual block

Figure 3.10: Neural soft-detector model implemented as a revised Resnet

RELU are applied before a fully connected layer.

Note that ResnetV2 simply rearranges the application of batch normalization, RELU and dense layers in each residual block. Thus, unless otherwise stated, we always compare two same sized Resnets.

3.12.2 Hypernetwork

Hypernetworks [107] was first proposed as a method to learn the structure of the parameters of a larger network, based on observations that up to 95% of the parameters in the network carries no additional information [108]. For our purpose, we prefer to have a small network that can perform symbol detection for a fixed channel $\mathbf{H}$, and we wish to construct a hypernetwork that can infer the parameters to the smaller detector

network. Precisely, we have a detector network whose task is to compute the symbol posterior $p_\theta(\mathbf{x}|\mathbf{y})$ associated with a fixed channel $\mathbf{H}$. We construct another network, the hypernetwork, that can learn the mapping $g_\phi : \mathbf{H} \to \theta$ for every $\mathbf{H}$. The schematic and the architecture is depicted in Fig. 3.11 and table 3.2. The detector network is reduced to a 3-layer Resnet with 256 nodes in each hidden layer in an effort to reduce the number of hypernetwork to detector network connections. Note that the detector network itself does not contain any parameters; thus the rightmost column of Table 3.2 depicts the dimensions of the connections within each layer, shown inside parentheses. The hypernetwork is tasked to produce the weights to all these connections in the detector network. The dimensions of the parameters inside the hypernetwork are shown in square brackets. We remark here that, this hyper detector network as a whole actually increased the overall network size substantially. When using hypernetworks, factorization techniques [109] are often used to reduce the number of parameters needed to be learned.

Table 3.2: Hyper Network architecture for the 18x2 receiver.

layer name	output size	hypernetwork	detector network
input (H, σ^2)	-	(72, 1)	-
input (y, σ^2)	-	-	(36, 1)
dense	1024	$[\ 73 \times 1024\ ]$	-
residual	1024	$\begin{bmatrix} 1024 \times 1024 \\ 1024 \times 1024 \end{bmatrix}$	-
embedding(θ_1)	9742	$[\ 1024 \times 9742]$	(37×256)
embedding(θ_2)	65336×2	$\begin{bmatrix} 1024 \times 65336 \\ 1024 \times 65336 \end{bmatrix}$	$\begin{pmatrix} 256 \times 256 \\ 256 \times 256 \end{pmatrix}$
embedding(θ_3)	65336	$[\ 1024 \times 65336\ ]$	(256×256)

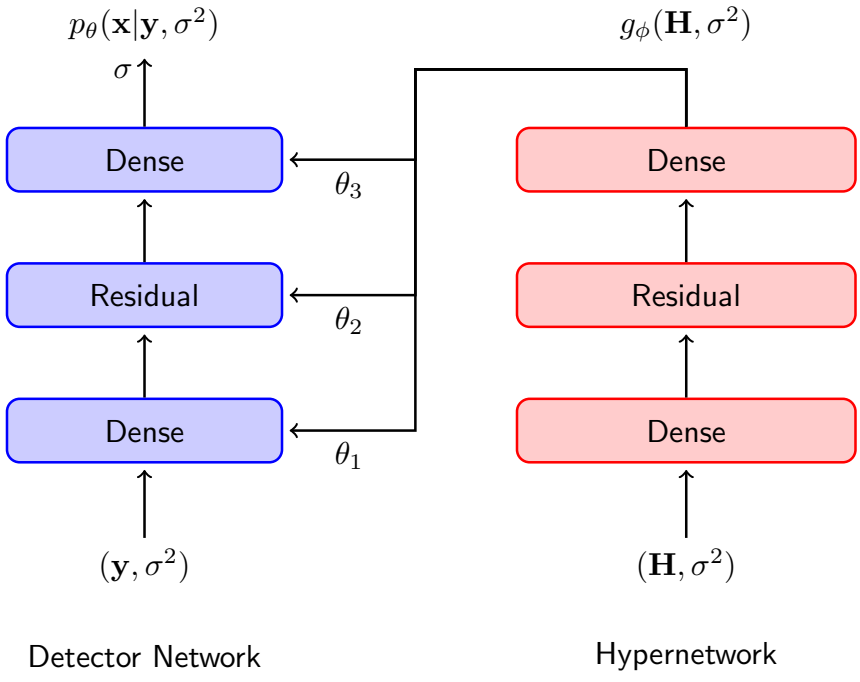


Figure 3.11: Schematic of the hyper detector network

3.12.3 Densenet

According to the authors of Densenets [22], this class of networks is a natural extension to Resnets. To allow unimpeded flow of information among all layers, Densenets employ the following novel techniques. First, Densenets ensure that all layers are directly connected: each layer obtains inputs from all preceding layer and relays its own output to all subsequent layers. Second, the features are not combined via summation, they are combined via concatenation, allowing direct access to features in all previous layers.

One striking feature of Densenets is that each layer only introduces a small set of features, while retaining features of all previous layers for maximum reuse. This makes them highly parameter efficient, requiring only a fraction of the parameters to achieve equal performance of Resnets.

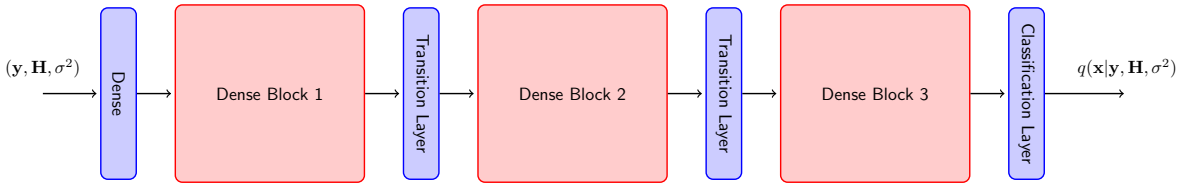


Figure 3.12: A neural receiver implemented as a three-block Densenet.

3.12.4 Densenet components

We describe the components of Densenets used in our experiments. The lth layer is a nonlinear function that receives features from all preceding layers in concatenated form:

$$\mathbf{x}_l = H_l([\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{l-1}]) \tag{3.33}$$

where $\mathbf{x}_l$ are the features of the lth layer and $\mathbf{x}_0$ are the inputs to the network, the operator $[\mathbf{a}, \mathbf{b}, ...]$ concatenates the input features. The nonlinear function H_l is a composite layer that combines batch normalization, RELU activation followed by a dense fully connected layer as in [96]. The growth rate k is determined by the number of features produced by each layer. Denote k_0 the number of features in the input layer, then the lth layer has $k_0 + k \times l$ input features. The growth of feature size is controlled by inserting transition layers between *dense* blocks. The transition layer is a composite layer that reduces the dimension of the input features by a compression factor θ, with $0 < \theta \leq 1$. It is implemented as a composite of BN, RELU and a fully connected layer. Bottleneck layers, a common technique used to reduce the number of input feature maps, does not apply here since we cannot perform 1x1 convolution in non-convolutional networks. The classification layer is structurally identical to the transition layer. It is the final layer that convert the features to logits of the individual class. A three-block Densenet is depicted in Fig. 3.12.

3.12.5 Densenet implementation details

As a reference, we use a baseline setup with the following parameters. The input layer dimension is $k_0 = 512$. It consists of three dense blocks, each with a total of $n_s = 12$ sublayers. The growth rate is set to $k = 32$. The compression ratio is set to $\theta = 0.5$. The reference Densenet is further described in Table 3.3. Note that the input to each layer in the dense block varies and increases with l, thus the stacked layers' dimensions are indicated with $[(k_0 + l \times k) \times k]_{l=0}^{n_s - 1}$. We experimented with three and four block Densenets, $k_0 \in \{512, 1024, 2048\}$, $k \in \{32, 48\}$, $n_s = \{12, 24\}$ and $\theta \in \{0.5, 1\}$. For the three block Densenet we also varied n_s in each block.

Table 3.3: A three block Densenet architecture for the 18x2 16-QAM receiver with $\theta = 0.5$ and $n_s = 12$.

layer name	output size	Densenet($k = 32$)
input (y, H, σ^2)	-	(36, 72, 1)
dense layer	512	[109×512]
dense block	896	$[(512 + l \times 32) \times 32]_{l=0}^{11}$
transition layer	448	[896×448]
dense block	832	$[(448 + l \times 32) \times 32]_{l=0}^{11}$
transition layer	416	[832×416]
dense block	800	$[(416 + l \times 32) \times 32]_{l=0}^{11}$
classification layer	256	[800×256]

3.13 Model training and evaluation

Training for the most part remains the same as the previous investigation (see Section 3.7) with the following exceptions.

Similar to the first part of this work, we generate non-repeating random symbols $\mathbf{x}$ from the discrete constellation, apply the 1-bit 18x2 random Rayleigh channel model

and produce the received pattern $\mathbf{y}$. The samples $D = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}$ are used to train the network using a categorical cross entropy loss.

Instead of Adam, Densenet uses SGD with momentum [25]. Momentum is set to 0.9, with initial learning rate of 0.1. The learning rate is reduced by a factor of 10 at 50% and 75% interval during training. It is worthwhile to note that the observed inferiority of Adam over SGD with momentum has been analyzed by Loshchiov and Hutter [110]. They attributed it to the general assumption that $l2$-regularization and weight decay are equivalent, which they revealed in the paper to be invalid for Adam. Their proposed AdamW optimizer [110] decouples the optimal choice of weight decay from the learning rate, and as a result made Adam's generalization performance competitive to SGD with momentum on image classification tasks. However, since we do not use weight decay in all our experiments, we do not observe performance differences between the two optimizers.

All receivers are evaluated on their symbol error rate (SER) performance.

3.14 Numerical results

3.14.1 Performance of Densenets

We compare the symbol error rate (SER) performance of the reference three block Densenet to other variants that increase the size and capacity of the network, either by increasing the number of dense blocks, increasing the number of sublayers, or increasing the dimension of the input layer (i.e. the output of the first dense layer). The result is plotted in Fig. 3.13. We see that already, the reference Densenet is outperforming the BMMSE receiver for SNRs up to 15dB. However, without exception, we see that any attempt at increasing the capacity of the network produces little to no performance gain. This provides a partial indication that a performance limit exists in these feed-forward, RELU activated networks. Next, we review the general result comparing the architectures

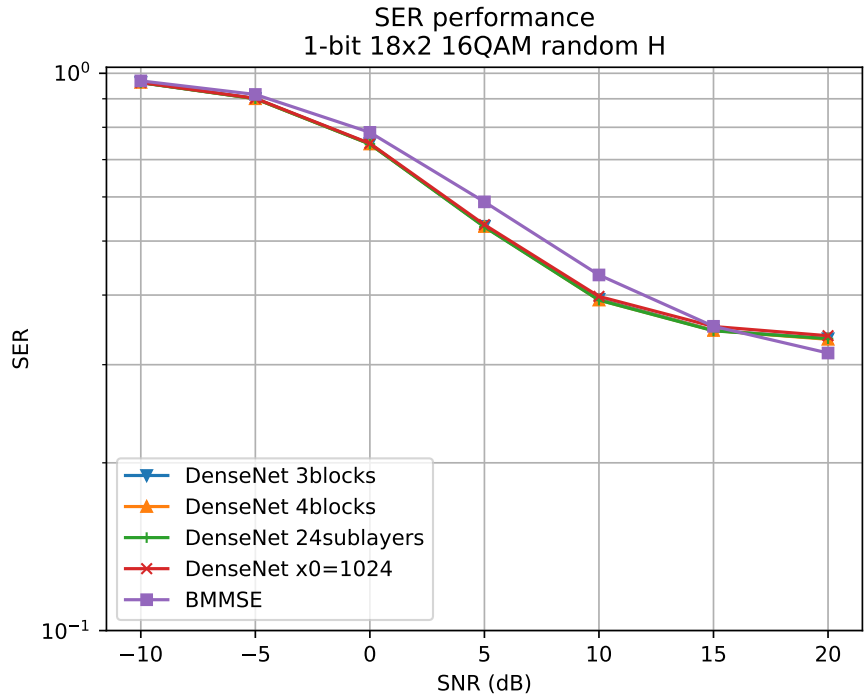


Figure 3.13: SER performance of neural receivers implemented as a 3-block Densenet (DenseNet 3blocks), 4-block Desenet (DenseNet 4blocks), 3-block Densenet with 24 sublayers (DenseNet 24sublayers), and Densenet with 1024 input features (DenseNet x0=1024).

described in section 3.12.

3.14.2 Performance of feed-forward networks

Here, we compare the SER performance of two 7-layer Resnet variants, the Hyper detector network and the three-block reference Densenet. The SER performance of these networks are plotted along side ideal ML and BMMSE in Fig. 3.14. It is revealing that these networks fail to close the gap to ML. As SNR increases, we see that they perform only marginally better than BMMSE. Noting that Densenets have direct information flow to all layers, essentially performing deep supervision, we are still unable to get a substantial gain in performance. Upon closer examination, there is no perceived performance difference between the two 7-layer Resnet variants. We note that the significant performance gain

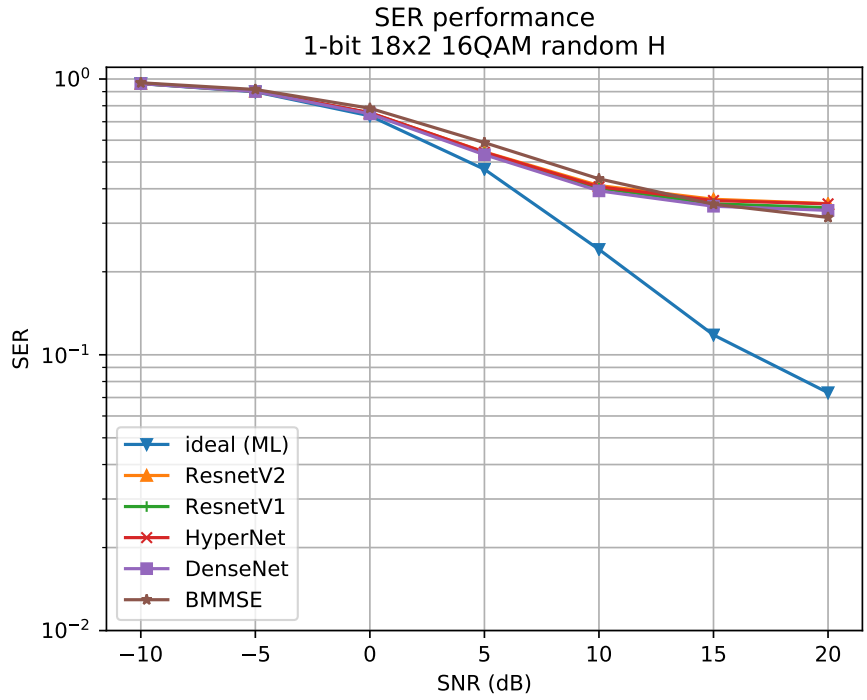


Figure 3.14: Comparison of SER performance of ideal ML, Bussgang MMSE (BMMSE) receiver and neural receivers implemented as a 7-layer Resnet (ResnetV1), 7-layer revised Resnet(ResnetV2), Hyper detector network (HyperNet) and Densenet.

reported in [96] was for a 1001-layer network. For the symbol detector, we see performance saturation already at 7 layers. In this regime, the direct connections and the resulting increased gradient flow produced negligible gain in detection performance. The hyper detector network performed slightly worse than other networks. We believe this is due to overfitting from the large amount of parameters in the network (over 200 million).

3.14.3 Performance of full precision receiver

We illustrate the difficulty in learning the 1-bit model by showing the performance of the reference Densenet on the same channel model with full-precision inputs. Precisely the model is given received signal $\mathbf{r}$ from (3.2) instead of the quantized signal $\mathbf{y}$ in (3.5). The Densenet is trained in the SNR range of $[0, 15]$ dB. Without any modification the

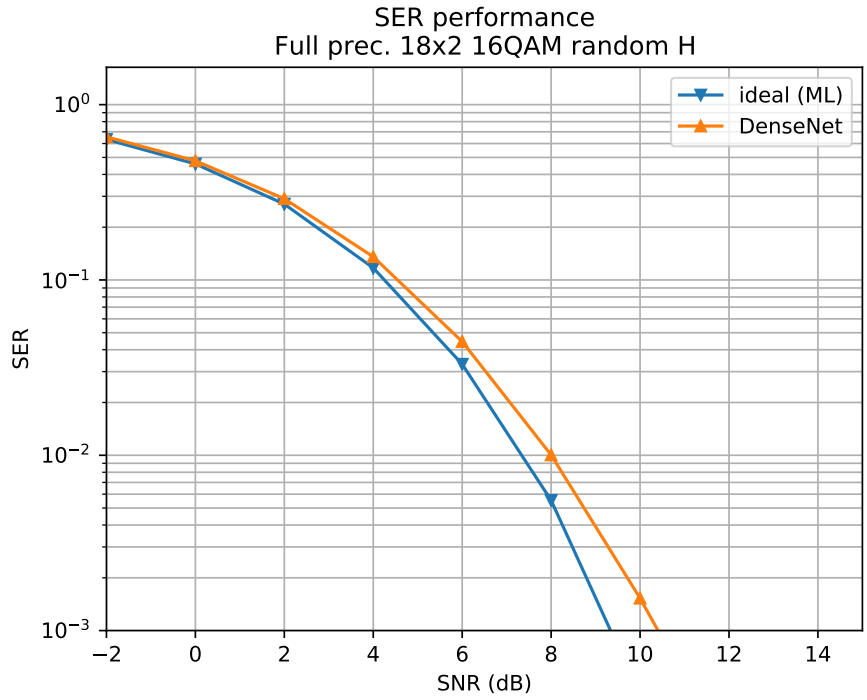


Figure 3.15: SER Performance of the reference Densenet on the same channel model with full-precision inputs.

Densenet is performing within 1dB of ML.

We conjecture that at high SNRs, the amount of information lost from 1-bit quantization and the absence of a dithering source (noise) prevented the network from learning useful structure at this region, leading to poor performance from a high-capacity, non-linear neural network.

3.14.4 Performance with DC-bias

Although none of the neural networks can match the performance of an ML detector, unlike the ML detector, it can still learn structure to maintain performance when the true data distribution deviates from the specified generative model.

We demonstrate this by introducing a DC offset that effectively changes the threshold of the quantizer decision boundary. LO self mixing is a common impairment found in

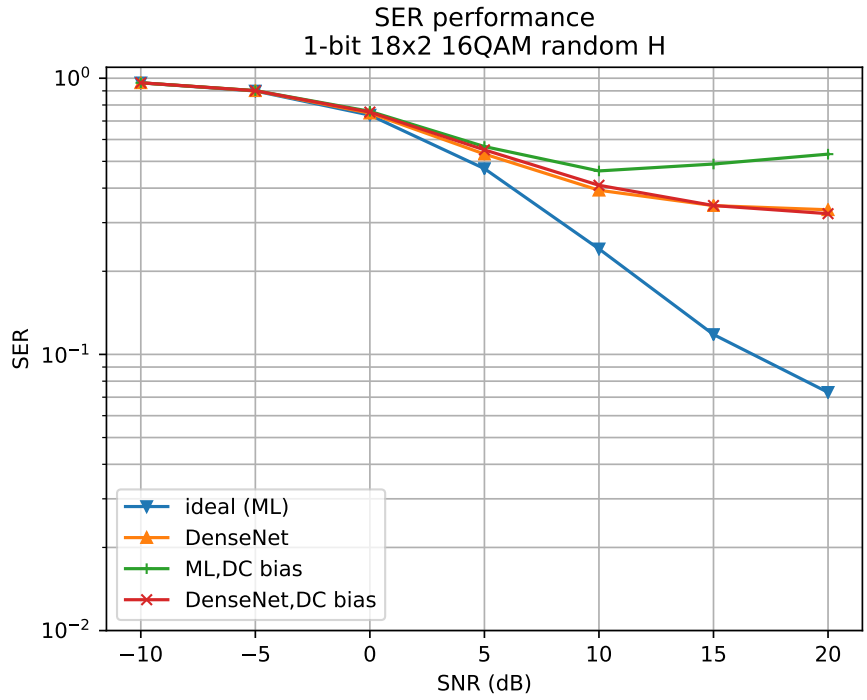


Figure 3.16: SER performance of ML detector when data is corrupted with DC bias (ML,DC bias) versus the performance of Densenet trained on data with DC bias (DenseNet,DC bias). Performance of ML and Densenet without the DC bias impairment are displayed as reference.

practical receiver designs, this manifests itself as a constant DC offset at the quantizer input. We model this by adding an offset with magnitude 0.25 at the input to each quantizer. The sign of the offset is generated from a Bernoulli distribution with probability $p = 0.8$. Once the offsets are generated, they are kept constant during training and evaluation.

The performance of ML and a Densenet trained on the impaired data is shown in Fig. 3.16. The likelihood function becomes more sensitive to changes at the quantizer input as SNR increases, thus we observe that the performance of ML deteriorates with SNR, to the point that it is outperformed by Densenet beyond 5dB. The degradation of ML is drastic especially at 15-20dB. On the other hand, the Densenet correctly learns the DC biases and performs almost identically as the baseline where no impairment was applied.

3.14.5 Densenet symbol estimator

Alternatively, we can construct a Densenet symbol estimator to first estimate the approximate symbol value. A simple minimum distance detector is then used to determine the candidate closest to the estimated value. The performance comparison of these symbol estimator based detectors with a Densenet classifier and the BMMSE receiver is shown in Fig. 3.17. Three configurations are considered: the first is the original reference 3-block Densenet with the output layer modified to generate 4 real values corresponding to the estimate in $\mathbb{C}^2$. The second is a 1-block Densenet, where the initial dimension is lowered to 256 from 512, number of sublayers is increased from 12 to 24, and the growth rate is set to 32. The final layer is a linear dense layer with dimensions 1024x4. This Densenet has half the number of parameters as the reference 3 block Densenet. Lastly, we experimented with the idea of augmenting the Bussgang receiver with a Densenet. Thus we generate an estimate using a combination of the Bussgang estimator and the Densenet's output:

$$\widehat{\mathbf{x}} = \widehat{\mathbf{x}}_{BMMSE} + \widetilde{\mathbf{x}}_{DNN} \tag{3.34}$$

The Densenet is tasked at learning the residue from the Bussgang receiver estimate only. Out of the three ideas considered, this is the only approach that can improve upon the BMMSE. We also note here that the Densenet is fundamentally being boosted by a BMMSE receiver. Thus the Densenet by itself has difficulty in learning the behavior of this function, especially at high SNR.

3.15 Conclusion for symbol detection

In this investigation, we gained additional insight into the difficulty in learning the 1-bit receiver model. We have empirically shown that the feed-forward type neural network

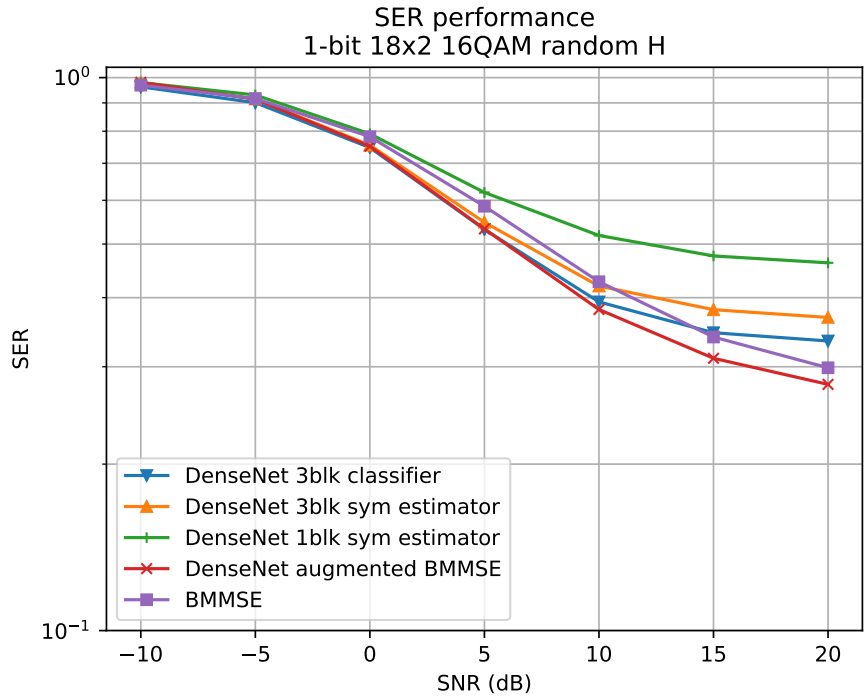


Figure 3.17: SER performance comparison of a 3-block Densenet classifier, a 3-block Densenet symbol estimator, a 1-block Densenet symbol estimator, and a Densenet augmented BMMSE receiver.

architectures are unable to close the gap to ML receivers. In the simpler setting of symbol detection, where no approximate inference method was used, we still observed a notable performance gap.

It has been noted in [111] that standard inference models are restricted to direct mappings from data to approximate posterior estimates. Methods that can refine the current posterior estimate [111,112] over a number of iterations using some type of recurrence may provide the missing link to closing this *amortization gap*. Based on our experience, sufficient inductive bias, perhaps in the form of a generative model (e.g. the likelihood) must be provided to the network to reduce the difficulty of the learning task.

Chapter 3, in part, is a reprint of unpublished material as it appears in the technical report: "Performance limit of feed-forward type neural 1-bit receivers". The dissertation

author was the primary investigator and author of this report.

3.16 Appendix

3.16.1 Neural EM algorithm

Another interesting avenue is a neural EM algorithm. We consider the following model. Let $\mathbf{y}$ be the observed and $\mathbf{r}$ be the latent variables, the parameter we wish to estimate is θ, in addition, the channel $\mathbf{H}$ is assumed known. The problem of interest also has a uniform prior on θ. Computing the posterior is reduced to computing the likelihood.

Thus the complete data likelihood is

$$p(\mathbf{r}, \mathbf{y}|\theta; \mathbf{H}) = p(\mathbf{r}|\mathbf{y}, \theta; \mathbf{H})\, p(\mathbf{y}|\theta; \mathbf{H}). \tag{3.35}$$

Normally in an EM algorithm, we initialize $\theta^{(0)}$, and iteratively generate refinements to the estimates, by running the E and M step. For time t, the E-step computes $\widehat{p}^{(t)} \triangleq \widehat{p}(\mathbf{r}|\mathbf{y}, \theta^{(t-1)}; \mathbf{H})$. It is followed by the M-step which finds that maximizer $\theta^{(t)}$ to the expected likelihood $\mathbb{E}_{\widehat{p}^{(t)}}[\log p(\mathbf{r}, \mathbf{y}|\theta, \mathbf{H})]$.

In the deep learning framework, we can learn the distribution $\widehat{p}^{(t)} \triangleq \widehat{p}(\mathbf{r}|\mathbf{y}, \theta^{(t)}; \mathbf{H})$ as an embedding $\mathbf{z}$ in some high dimensional space; this embedding is trained by having it perform the estimation of $\mathbf{r}$. With $\mathbf{z}$ as a learned representation of $\widehat{p}^{(t)}$, we supply this to another network that computes the expected likelihood. In other words, we have

$$g(\theta; \mathbf{y}, \mathbf{H}, \mathbf{z}) \simeq \mathbb{E}_{\widehat{p}}[\log p(\mathbf{r}, \mathbf{y}|\theta, \mathbf{H})] \tag{3.36}$$

for $\theta \in \Theta$. In our particular problem Θ is a discrete, finite set.

Chapter 4

Structured Deep Learning Detectors

Massive multiple-input multiple-output (MIMO) systems implemented with one-bit analog-to-digital converters (ADCs) offers a viable solution to control hardware complexity and lower power consumption. It is also desirable to obtain a low complexity data detector that can attain maximum likelihood (ML) performance. However, the severe distortion on the received signal made the design of a low-complexity data detector challenging. In this work, existing model based approaches are evaluated to identify the shortcomings of gradient methods that rely on the exact likelihood. We focus on a set of algorithms based on variational methods and propose a structured deep learning detector based on stochastic variational inference. Stochastic estimate of the mean field update is introduced to reduce complexity of the algorithm. Damping is added to further improve the performance of mean field inference (MFI). Parallel processing is proposed to reduce inference time. The proposed PSMFNet contains few parameters and can be trained efficiently using standard deep learning techniques. In numerical experiments, the proposed detector is shown to outperform existing methods that do not employ a second candidate search step.

4.1 Introduction

To meet the ever increasing demand in spectral efficiency while maintaining complexity at a reasonable level, a breakthrough in wireless communications technology is necessary. Massive multiple-input multiple-output (MIMO) has been touted as one such technology that can realize such benefits. It has the potential to increase throughput and spectral efficiency by orders of magnitudes [113].

However, a massive MIMO system places a large number of antennas at the base station, possibly hundreds of antennas. This requires the implementation a large number of analog receive chains, significantly increasing the power consumption and complexity of the analog front end. The use of low-resolution analogy-to-digital converters (ADCs) can lead to lower power consumption [4]. The analog chain can be much more simplified if only the sign bit is preserved in the signal, in which case a comparator can be used and the automatic gain control (AGC) subsystem can be eliminated [7].

A major challenge with low-resolution ADCs is that the model becomes nonlinear due to the severe quantization on the signal. This necessitates a nonlinear receiver to achieve near optimal performance.

The recent advances in deep learning had seen deep neural networks outperforming traditional methods in many areas [17, 100, 114–116]. The ability of a neural network to approximate nonlinear functions [13, 14] also made it a natural candidate for this application.

We begin by evaluating existing model based approaches, provide a detailed assessment on the shortcomings of gradient methods that rely on the exact likelihood. We introduce a set of algorithms based on variational methods, and propose a deep learning detector based on mean field inference that is capable of outperforming the current state of the art. A background of related research in this area is presented next.

An extensive body of literature have been devoted to the study of 1-bit symbol

detection. Low complexity linear detector were studied in [11, 12, 117]. In [117], an advanced linear detector based on the Bussgang decomposition was shown to outperform previously proposed linear detectors. However, there remains a significant performance gap that can be addressed by more sophisticated detectors. GAMP based detectors were proposed in [57,118,119]. A BiG-AMP based joint channel and data estimation algorithm was studied in [120]. A scheme that combines variational infernence and bilinear GAMP was developed in [121]. A near Maximum Likelihood (nML) detector [56] was developed using projected gradient method over the likelihood function with a relaxed constraint. The proposed two stage detection method was shown to achieve near ML performance. In [84] a one-bit sphere decoding (OSD) algorithm was developed to reduce the candidate search space. The detection complexity of OSD is significantly reduced, but the complexity of the list construction is exponential in the number of users as indicated in [122].

Recently, machine learning techniques are explored in [54, 122–125]. Several works [123–125] considered blind detection. Blind detection bypasses the channel estimation stage. Instead, the training sequence is used to learn the nonlinear relationship between the channel output and the source directly via supervised learning. However, these schemes are only capable of supporting a limited number of users and low-order modulations. The similarity of the 1-bit detection problem to that of learning a binary classifier is exploited in [122,126]. In [122], the SVM-based detector demonstrated robust performance with the absence of perfect CSI. However, the equivalent hinge loss and the exclusion of SNR information means its performance is still limited compared to a detector that fully incorporates model knowledge. The same authors proposed the OBMNet [54], a scheme that incorporates the likelihood meaningfully to the problem in a supervised learning framework. They suggested the use of the sigmoid approximation and showed that the reformulated problem using the approximated likelihood achieved the same performance as ML. The OBMNet demonstrated superior performance over the SVM-based detector.

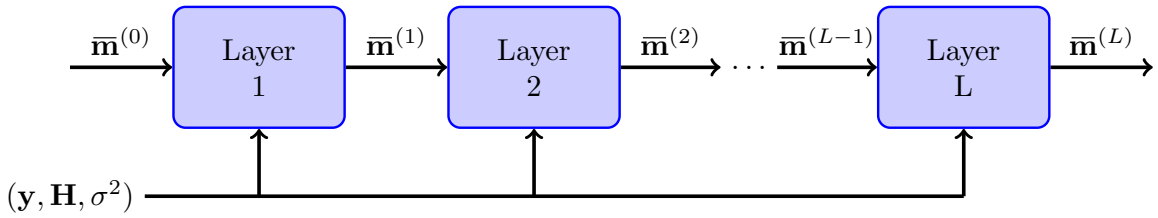


Figure 4.1: Common architecture of the unfolded MFNets.

The network required that the SNR information be removed and used a single learned scaling factor in place of it in order to make the training process convergent. Thus it is still suboptimal compared to ML.

We propose a detector based on mean-field inference. The algorithm incorporates the model using the approximated log likelihood. The resulting mean field network (MFNet) is inspired by [127]. The iterative algorithm is unfolded [49], leading to a deep network that can be trained end-to-end via supervised learning (see Fig. 4.1). Note that the deep networks considered in [49] has 4 and 25 layers respectively. We next introduce a variant that replaces the gradient with a stochastic estimate, eliminating the exponential complexity of computing the exact gradient. This technique is related to the stochastic variational inference (SVI) method proposed in [128]. Parallel updates are incorporated to eliminate dependencies from sequential updates, providing a further acceleration to the algorithm. All variants of the MFNet are shown to have superior performance compared to the state of the art.

The main contributions of this work are summarized here.

- We identify the features of the exact likelihood that make it difficult for iterative optimization methods to converge to the true solution, and explain in detail how the sigmoid approximation makes these methods more robust, especially at high SNR scenarios.

- We develop three algorithms based on variational methods, assess the performance of each of these approximated approaches and ultimately settle on damped mean field inference (MFI) that surpasses the performance of the current state of the art.
- Stochastic gradient and parallel updates are introduced to lower the complexity and accelerate the algorithm compared to the vanilla MFI.
- In contrast to nML [56] and SVM based method [122], our proposed detector achieves near ML performance in the SNR region of interest without the need of a second candidate search stage.
- Using two architectural configurations, we show that structured deep learning enables the detector to achieve the best possible hand-tuned result.

The remainder of this paper is organized as follows. Section 4.2 describes the system model and the data detection problem. Section 4.3 derives the gradient descent methods and provides an assessment of the weaknesses and merits of each algorithm. Section 4.4 introduces two of the three variational methods. Section 4.5 is devoted to mean field inference, damped mean field inference and its relation to proximal gradient descent. Section 4.6 describes in detail the techniques employed for the proposed algorithm. Section 4.7 describes the network architecture and discusses model training issues. In Section 4.8, the proposed algorithms are evaluated in simulations. Section 4.9 ends this work with concluding remarks.

Throughout the paper, a_{ij} refers to the (i,j)th entry of matrix $\mathbf{A}$. $\mathbf{A}^{\mathsf{T}}$ denotes the transpose of $\mathbf{A}$. $\mathsf{re}(\cdot)$ and $\mathsf{im}(\cdot)$ extracts the real and imaginary components from a complex quantity respectively. $\log(\cdot)$ is the natural logarithm. $\mathrm{diag}(\cdot)$ converts the vector argument into a diagonal matrix. $\phi(x)$ and $\Phi(x)$ denotes the standard normal probability density function (pdf) and cumulative distribution function (cdf) respectively. $\mathbb{1}_A(x)$ denotes the indicator function that takes on the value 1 if $x \in A$ and 0 if $x \notin A$.

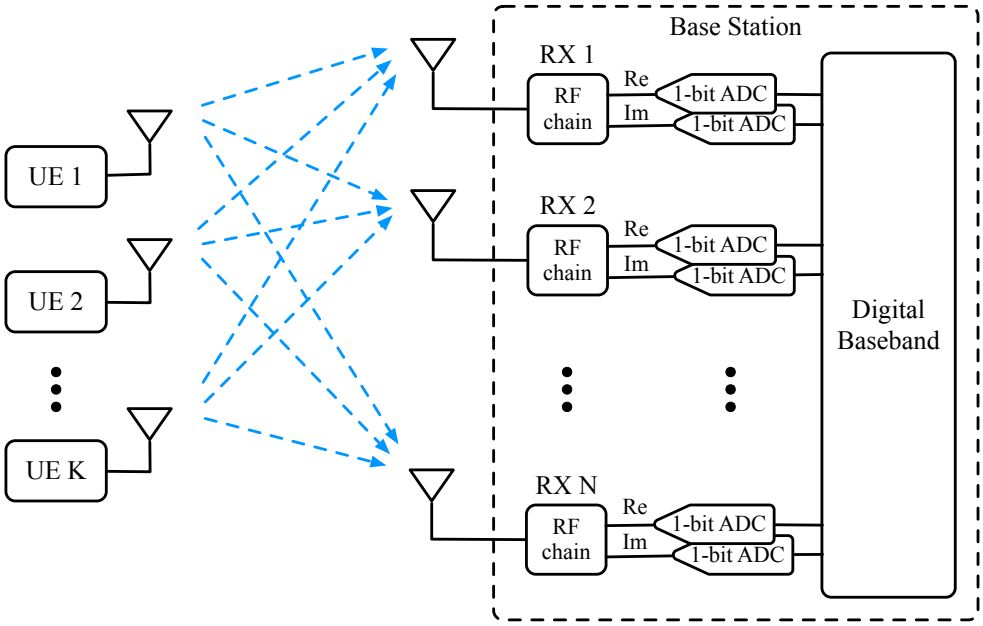


Figure 4.2: Model of a 1-bit massive MIMO system.

4.2 System model and symbol detection

Consider a massive MIMO uplink as depicted in Fig. 4.2 with K single antenna users and N base station antennas, such that $N > K$. The analog receive signal of a MIMO communication channel is given by

$$\bar{\mathbf{r}} = \overline{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{n}} \tag{4.1}$$

with source symbol vector $\bar{\mathbf{x}} \in \overline{\mathcal{X}}^K \subset \mathbb{C}^K$, channel $\overline{\mathbf{H}} \in \mathbb{C}^{N\times K}$, noise vector $\bar{\mathbf{n}} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I})$ and analog receive vector $\bar{\mathbf{r}} \in \mathbb{C}^N$. $\overline{\mathcal{X}}$ is a finite M-QAM constellation such that $|\overline{\mathcal{X}}| = M$. The channel is assumed to be block fading and known at the receiver. The transmitted signal for the k-th user is assumed to have unit power constraint $\mathbb{E}[|\bar{x}_k|^2] = 1$. The SNR is defined as $\rho = 1/\sigma^2$.

It is convenient to transform the problem into an equivalent real-valued system,

using the following notation

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{4.2}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathsf{re}(\overline{\mathbf{H}}) & -\mathsf{im}(\overline{\mathbf{H}}) \\ \mathsf{im}(\overline{\mathbf{H}}) & \mathsf{re}(\overline{\mathbf{H}}) \end{bmatrix} \tag{4.3}$$

and

$$\mathbf{r} = \begin{bmatrix} \mathsf{re}(\bar{\mathbf{r}}) \\ \mathsf{im}(\bar{\mathbf{r}}) \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \mathsf{re}(\bar{\mathbf{x}}) \\ \mathsf{im}(\bar{\mathbf{x}}) \end{bmatrix}, \ \mathbf{n} = \begin{bmatrix} \mathsf{re}(\bar{\mathbf{n}}) \\ \mathsf{im}(\bar{\mathbf{n}}) \end{bmatrix}. \tag{4.4}$$

The 1-bit quantized signal is generated by

$$\mathbf{y} = \mathcal{Q}(\mathbf{r}) \tag{4.5}$$

where $\mathcal{Q}$ is an element-wise application of the signum function, $\mathrm{sgn}(\cdot)$, defined by

$$\mathrm{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{otherwise.} \end{cases} \tag{4.6}$$

Note that each orthogonal real symbol x_k is drawn from the PAM constellation $\mathcal{X} \subset \mathbb{R}$ with cardinality $|\mathcal{X}| = M^{1/2}$.

4.2.1 Log likelihood function

The log likelihood of receiving $\mathbf{y}$ given source vector $\mathbf{x}$ for a 1-bit receiver is given by [56]

$$\log p(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{2N} \log \Phi\Big(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathsf{T}} \mathbf{x}\Big) \tag{4.7}$$

where y_i and $\mathbf{h}_i^{\mathsf{T}}$ are the ith element and ith row of the receive vector and the channel matrix resp. We remark here that unlike the full-precision case in [50], we expect a competitive

algorithm to depend on the noise variance input to achieve ML like performance over a wide range of SNRs.

The symbol detection problem amounts to finding a candidate symbol that maximizes the log likelihood, i.e.

$$\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbf{X}} \log p(\mathbf{y}|\mathbf{x}). \tag{4.8}$$

This is a discrete optimization problem and typically requires an exhaustive search.

4.3 Iterative gradient descent methods

In this section, we derive gradient descent algorithms via two objective functions, the exact model likelihood and the sigmoid approximated likelihood. We then discuss the problems associated with the exact likelihood.

4.3.1 Optimization via exact model likelihood

The ML detection problem for the 1-bit model is given by

$$\widehat{\mathbf{x}}_{\mathsf{ML}} = \arg\max_{\mathbf{x}\in\mathcal{X}^{2K}} \sum_{i=1}^{2N} \log \Phi(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathsf{T}}\mathbf{x}). \tag{4.9}$$

The problem is relaxed such that we solve the problem

$$\widehat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathbb{R}^{2K}} \sum_{i=1}^{2N} \log \Phi(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathsf{T}}\mathbf{x}). \tag{4.10}$$

Denote $\widetilde{\mathcal{P}}(\mathbf{x})$ the objective function we wish to optimize

$$\widetilde{\mathcal{P}}(\mathbf{x}) = \sum_{i=1}^{2N} \log \Phi(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathsf{T}}\mathbf{x}). \tag{4.11}$$

Let $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_{2N}]^\mathsf{T} = \mathrm{diag}(y_1, \ldots, y_{2N})\mathbf{H}$. Then the gradient of $\widetilde{\mathcal{P}}$ is

$$\nabla\widetilde{\mathcal{P}}(\mathbf{x}) = \sum_{i=1}^{2N} \sqrt{2\rho}\, \mathbf{g}_i \frac{\varphi(\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x})}{\Phi(\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x})} \tag{4.12}$$

$$= \sqrt{2\rho}\mathbf{G}^\mathsf{T} \frac{\varphi(\sqrt{2\rho}\mathbf{G}\mathbf{x})}{\Phi(\sqrt{2\rho}\mathbf{G}\mathbf{x})} \tag{4.13}$$

where $\varphi(\cdot)$ and $\Phi(\cdot)$ function applies elementwise. We abuse the notation and let the operations of $\varphi(\mathbf{z})/\Phi(\mathbf{z}) \mapsto \mathbf{r}$ be applied element wise, resulting in a vector output $\mathbf{r}$. We note here that the $\varphi(x)/\Phi(-x)$ is the normal hazard function used in survival analysis.

An iterative descent algorithm can be constructed based on the following iteration

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \alpha\mathbf{G}^\mathsf{T} \frac{\varphi(\sqrt{2\rho}\mathbf{G}\mathbf{x})}{\Phi(\sqrt{2\rho}\mathbf{G}\mathbf{x})}. \tag{4.14}$$

Note that the SNR dependent coefficient $\sqrt{2\rho}$ is replaced by a constant step size α. This reduces sensitivity of the algorithm to the steep landscape at high SNR. Further discussion can be found in section 4.3.3. The algorithm also benefits from the norm constraint of the transmitted symbols. Thus a projected gradient algorithm is used as in [56].

The Hessian of $\widetilde{\mathcal{P}}(\mathbf{x})$ is given by

$$\nabla^2\widetilde{\mathcal{P}}(\mathbf{x}) \tag{4.15}$$

$$= \sum_{i=1}^{2N} 2\rho \left\{ -\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x} \frac{\varphi(\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x})}{\Phi(\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x})} - \frac{\varphi^2(\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x})}{\Phi^2(\sqrt{2\rho}\mathbf{g}_i^\mathsf{T}\mathbf{x})} \right\} \mathbf{g}_i\mathbf{g}_i^\mathsf{T} \tag{4.16}$$

$$= 2\rho\mathbf{G}^\mathsf{T}\, \mathrm{diag}\left\{ -\sqrt{2\rho}\mathbf{G}\mathbf{x} \frac{\varphi(\sqrt{2\rho}\mathbf{G}\mathbf{x})}{\Phi(\sqrt{2\rho}\mathbf{G}\mathbf{x})} - \frac{\varphi^2(\sqrt{2\rho}\mathbf{G}\mathbf{x})}{\Phi^2(\sqrt{2\rho}\mathbf{G}\mathbf{x})} \right\} \mathbf{G}. \tag{4.17}$$

We note that the Hessian around the solution is near singular due to the flatness along the direction defined by the line $\{\alpha\mathbf{x}_0 | \alpha \in \mathbb{R}^+\}$. This makes second order approaches such as Newton's method unsuitable. The condition number of the Hessian at the solution is larger than 10e+6 at 30dB SNR.

4.3.2 Sigmoid approximated likelihood

The approximated likelihood approach for the 1-bit problem was first used in [121] and then later used in training a deep unfolded network in [54]. It has been well known in logistic regression that the sigmoid function can be well approximated by the probit function [129, §4.5]. In [130], a systematic study of the sigmoid approximation to the normal cdf produced the tightest result known at the time. Specifically, the normal cdf can be approximated by

$$\Phi(t) \approx \sigma(ct) = \frac{1}{1+e^{-ct}} \tag{4.18}$$

with $c = 1.702$. It was shown that $|\Phi(t) - \sigma(ct)| \leq 0.0095, \forall t \in \mathbb{R}$.

Applying this approximation to the likelihood leads to the approximated maximum likelihood (AML):

$$\widehat{\mathbf{x}}_{\mathsf{AML}} = \arg\min_{\mathbf{x}\in\mathcal{X}^K} \sum_{i=1}^{2N} \log(1+e^{c\sqrt{2\rho}\mathbf{g}_i^{\mathsf{T}}\mathbf{x}}). \tag{4.19}$$

Again we solve the relaxed problem

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{R}^{2K}} \sum_{i=1}^{2N} \log(1+e^{c\sqrt{2\rho}\mathbf{g}_i^{\mathsf{T}}\mathbf{x}}). \tag{4.20}$$

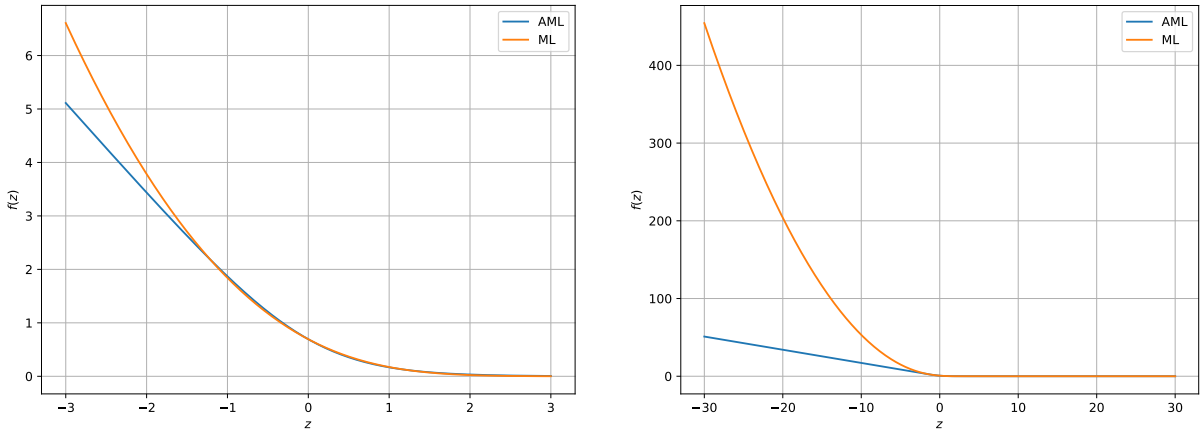
Let $\mathcal{P}(\mathbf{x})$ denote the objective function

$$\mathcal{P}(\mathbf{x}) = \sum_{i=1}^{2N} \log(1+e^{c\sqrt{2\rho}\mathbf{g}_i^{\mathsf{T}}\mathbf{x}}) \tag{4.21}$$

Then the gradient of $\mathcal{P}(\mathbf{x})$ is

$$\nabla\mathcal{P}(\mathbf{x}) = \sum_{i=1}^{2N} \frac{-c\sqrt{2\rho}\mathbf{g}_i}{1+e^{c\sqrt{2\rho}\mathbf{g}_i^{\mathsf{T}}\mathbf{x}}} \tag{4.22}$$

$$= -c\sqrt{2\rho}\mathbf{G}^{\mathsf{T}}\sigma(-c\sqrt{2\rho}\mathbf{G}\mathbf{x}). \tag{4.23}$$



(a) (b)

Figure 4.3: Comparison of the penalty functions for AML and ML. (a) The penalty function of AML and ML when the prediction is consistent with the observed sign. (b) The penalty of function of AML and ML at high SNR when the prediction is inconsistent with the observed sign.

An iterative gradient descent update rule can be defined as

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \mathbf{G}^{\mathsf{T}} \sigma(-c\sqrt{2\rho}\mathbf{G}\mathbf{x}) \tag{4.24}$$

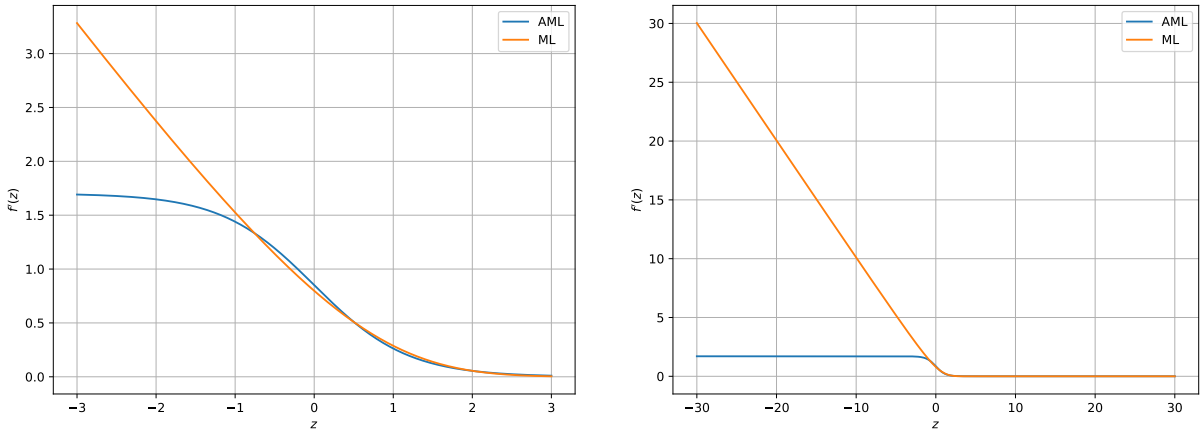
where we have omitted the SNR dependent term $c\sqrt{2\rho}$ similar to (4.14). Finally, the output is normalized because the likelihood is insensitive to the scaling of the solution at high SNR

$$\widehat{\mathbf{x}} = K \frac{\mathbf{x}^{(L)}}{\|\mathbf{x}^{(L)}\|}. \tag{4.25}$$

where L denote the last iteration.

4.3.3 Optimization landscape

We note that the negative log likelihood function is convex. The function has a narrow ridge along the line connecting the origin to the solution $\mathbf{x}$. This ridge extends beyond the solution and becomes nearly flat along this direction in the high SNR regime



(a) (b)

Figure 4.4: Comparison of the gradient functions for AML and ML. Gradient is much larger for ML at high SNR when the channel prediction is inconsistent with the observed sign (i.e. when the function argument takes on large negative value).

since the amplitude of the solution is non-identifiable [122]. To see this, let $\sigma^2 \to 0$, for any positive scalar $\alpha > 0$, the 1-bit outputs are unchanged

$$\mathbf{y} = \text{sgn}(\mathbf{Hx}) = \text{sgn}(\alpha \mathbf{Hx}). \tag{4.26}$$

The basic strategy to solving this problem is by finding the optimal direction from the origin and then normalize the user's symbol via the unit power constraint

$$\|\mathbf{x}\|^2 = K. \tag{4.27}$$

We now examine the penalty function and its gradient to gain insight into the structure of the log likelihood. Define $f(z) = \log \Phi(z)$ and $f(z) = \log \sigma(cz)$ as the penalty functions of the respective likelihoods with argument $z \triangleq \sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathsf{T}} \mathbf{x}$. We define the gradient functions as the derivative of the penalty functions. The argument to the penalty function has the following properties. When the channel output prediction $\mathbf{h}_i^{\mathsf{T}} \widehat{\mathbf{x}}$ is consistent with the observed sign y_i, the value is positive. When it is inconsistent with the

observed sign, the value is negative. In addition, it is scaled by the square root of the SNR.

This sigmoid approximation is beneficial to the problem for the following reasons:

In the neighborhood of the solution, when the channel output prediction is consistent with the observed quantizer outputs, the two penalty functions behave almost identically. This is illustrated in Fig. 4.3a.

When the prediction is inconsistent with the observed sign, the penalty from the approximated likelihood is linear in its argument as opposed to the quadratic penalty for the exact likelihood. As expected, the gradient of the exact likelihood takes on increasingly large values at high SNR. This helps explain the sensitivity of gradient descent at high SNR. See Figs. 4.3b, 4.4b. The attributes associated with the exact likelihood also made training unfolded deep network and general feed forward networks difficult.

Iterative algorithms based on the sigmoid approximation are more stable at high SNR. The overall effect is a reduction in the spectral norm of the Hessian, thus larger step sizes can be tolerated at high SNR, or the same step size can be used over a larger range of SNRs. This property is applicable to MFI, where the log likelihood is used to compute the gradient of each step.

It is noted in section 4.8 that both algorithms fall short in reaching the performance of ML at low SNR. We seek alternative approaches that can reach that goal without adding another stage. In the sequel, we turn our attention to variational methods and evaluate the performance achievable using these approximation schemes.

4.4 Variational methods

Three variational methods are considered in this section. The first method applies variational inference to the exact likelihood and approximates the source symbols

as Gaussians. The second method makes use of the approximated likelihood described in Section 4.3.2. It lower bounds the approximate likelihood with a Gaussian approximation [129, §10.5], when the source is assumed Gaussian, this lead to an approximate Gaussian posterior with a variational parameter to be optimized via an EM algorithm. The last method is mean field inference, where the posterior is approximated by a fully factored distribution, In contrast to the first two methods, the source distribution remains discrete.

4.4.1 Variational Bayes

In this section, we consider variational Bayes (VB) method directly applied to the exact likelihood. Full derivation of the VB method as well as its relation to the EM-MAP is provided in Section 4.10.1.

The joint likelihood of the 1-bit model has the form [131]

$$p(\mathbf{y}, \mathbf{r}|\mathbf{x}) = \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) p(\mathbf{r}|\mathbf{x}) \tag{4.28}$$

where

$$p(\mathbf{r}|\mathbf{x}) = \frac{1}{(\pi\sigma^2)^K} \exp\Big\{-\frac{1}{\sigma^2}\|\mathbf{r} - \mathbf{H}\mathbf{x}\|^2\Big\}. \tag{4.29}$$

$\mathcal{Q}^{-1}(\mathbf{y})$ is the set $\{\mathbf{r} : \mathcal{Q}(\mathbf{r}) = \mathbf{y}\}$. Assume the prior is truncated Gaussian with distribution, $p(\mathbf{x}) \sim \mathcal{TN}(\mathbf{0}, \frac{1}{2}\mathbf{I}, [-1/\sqrt{2}, 1/\sqrt{2}])$. The truncated region is a suitable condition for QPSK.

The posterior is approximated with a fully factored distribution

$$q(\mathbf{r}, \mathbf{x}) = \prod_{i=1}^{2N} q(r_i) \prod_{j=1}^{2K} q(x_j). \tag{4.30}$$

Given $q(\mathbf{x})$ and all other $q(r_j), j \neq i$, the optimal $q(r_i)$ is a truncated Gaussian with

$$q(r_i) \sim \mathcal{TN}(m_i, \frac{\sigma^2}{2}, \mathcal{Q}^{-1}(y_i)) \tag{4.31}$$

where

$$m_i = \mathbf{h}_i^{\mathsf{T}} \, \mathbb{E}_{q(\mathbf{x})}[\mathbf{x}]. \tag{4.32}$$

The mean of r_i can be computed as

$$\mathbb{E}[r_i] = m_i + \frac{y_i \phi(s_i)\sigma_i}{\Phi(y_i s_i)}, \quad s_i = \frac{m_i}{\sigma_i}. \tag{4.33}$$

In order to obtain the optimal $q(x_i)$, first we define the quantities

$$\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\widehat{\mathbf{r}} \tag{4.34}$$

$$\boldsymbol{\Lambda} = 2\mathbf{I} + \mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{H} \tag{4.35}$$

where $\widehat{\mathbf{r}} = \mathbb{E}_{q(\mathbf{r})}[\mathbf{r}]$. Given $q(\mathbf{r})$ and all other $q(x_j), j \neq i$, the optimal $q(x_i)$ is also a truncated Gaussian $q(x_i) \sim \mathcal{TN}(m_i, \lambda_{i,i}^{-1}, [-1/\sqrt{2}, 1/\sqrt{2}])$ with mean

$$m_i = \mu_i - \lambda_{i,i}^{-1}\boldsymbol{\lambda}_{i,\setminus i}^{\mathsf{T}}(\mathbb{E}_q[\mathbf{x}_{\setminus i}] - \boldsymbol{\mu}_{\setminus i}) \tag{4.36}$$

where $\boldsymbol{\lambda}_{i,\setminus i}^{\mathsf{T}} = [\lambda_{i,i}, \lambda_{i,i-1}, \lambda_{i,i+1}, \lambda_{i,2K}]$ and $\lambda_{i,j}$ denotes the i,jth entry of $\boldsymbol{\Lambda}$.

The algorithm proceeds as follows. First all factored distributions are initialized with zero mean and unit variance. Then at each iteration, each factored distribution is updated while holding all other distributions fixed.

4.4.2 Variational lower bound

Using a variational characterization of a convex function, the sigmoid function can be lowered bounded by a function that allows us to produce a closed form approximation of the posterior. The idea was first proposed in [132] and disseminated in [129].

Lower bounding the approximate likelihood

A variational lower bound of the sigmoid can be obtained as follows. Noting that the function

$$f(x) = -\log(e^{x/2} + e^{-x/2}) \tag{4.37}$$

is convex function in x^2, a supporting hyperplane can be formed such that it is a global lower bound of $f(x)$. Using this result, a lower bound can be realized for any ξ

$$\sigma(z) \geq \sigma(\xi) \exp\Big\{ \frac{z-\xi}{2} - \lambda(\xi)(z^2 - \xi^2) \Big\} \tag{4.38}$$

where

$$\lambda(\xi) = \frac{1}{2\xi}\Big[\sigma(\xi) - \frac{1}{2}\Big]. \tag{4.39}$$

This lower bound is exact at $\xi^2 = z^2$.

We let the prior be Gaussian with distribution $p(\mathbf{x}) = \mathcal{N}(0, \frac{1}{2}\mathbf{I})$. Using the sigmoid approximation, the likelihood becomes

$$\log p(\mathbf{y}|\mathbf{x}) \approx \sum_{i=1}^{2N} \log \sigma(\mathbf{g}_i^{\mathsf{T}}\mathbf{x}) \tag{4.40}$$

where $\mathbf{g}_i = c\sqrt{2\rho} y_i \mathbf{h}_i$. Denote $p(y_i|\mathbf{h}_i, \mathbf{x}) = \sigma(\mathbf{g}_i^{\mathsf{T}}\mathbf{x})$. We now apply the variational lower bound

$$p(y_i|\mathbf{h}_i, \mathbf{x}) = \sigma(\mathbf{g}_i^{\mathsf{T}}\mathbf{x}) \geq h(\mathbf{g}_i^{\mathsf{T}}\mathbf{x}, \xi_i) \tag{4.41}$$

where

$$h(\mathbf{g}_i^{\mathsf{T}}\mathbf{x}, \xi_i) = \sigma(\xi_i) \exp\Big\{ \frac{\mathbf{g}_i^{\mathsf{T}}\mathbf{x} - \xi_i}{2} - \lambda(\xi_i)\big((\mathbf{g}_i^{\mathsf{T}}\mathbf{x})^2 - \xi_i^2\big) \Big\}. \tag{4.42}$$

The lower bound is in quadratic form, we can now approximate the posterior with a Gaussian. Taking the log of the joint distribution, we have

$$\log p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) \geq \log h(\mathbf{x}, \boldsymbol{\xi}) + \log p(\mathbf{x}) \tag{4.43}$$

$$= -\frac{1}{2}\|\mathbf{x}\|^2 + \sum_{i=1}^{2N} \Big\{ \log \sigma(\xi_i) + \frac{\mathbf{g}_i^{\mathsf{T}}\mathbf{x} - \xi_i}{2} - \lambda(\xi_i)\big((\mathbf{g}_i^{\mathsf{T}}\mathbf{x})^2 - \xi_i^2\big) \Big\} \tag{4.44}$$

$$= -\frac{1}{2}\|\mathbf{x}\|^2 + \sum_{i=1}^{2N} \Big\{ \frac{\mathbf{g}_i^{\mathsf{T}}\mathbf{x}}{2} - \lambda(\xi_i)\mathbf{x}^{\mathsf{T}}\mathbf{g}_i\mathbf{g}_i^{\mathsf{T}}\mathbf{x} \Big\} + \text{ const.} \tag{4.45}$$

The Gaussian posterior $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ has mean and variance

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\Big(\sum_{i=1}^{2N} \frac{\mathbf{g}_i}{2}\Big) \tag{4.46}$$

$$\boldsymbol{\Sigma}^{-1} = \mathbf{I} + 2\sum_{i=1}^{2N} \lambda(\xi_i)\mathbf{g}_i\mathbf{g}_i^{\mathsf{T}}. \tag{4.47}$$

An EM update is required to optimize each ξ_i, which is given by [129]

$$(\xi_i)^2 = \mathbf{g}_i^{\mathsf{T}}\, \mathbb{E}[\mathbf{x}\mathbf{x}^{\mathsf{T}}]\mathbf{g}_i = \mathbf{g}_i^{\mathsf{T}}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\mathsf{T}})\mathbf{g}_i. \tag{4.48}$$

Thus we iterate between the E step using (4.46) and (4.47) and the M step via (4.48).

The next three sections are devoted to the introduction of MFI, a detailed description of the techniques introduced to arrive at the inference algorithm, and lastly the deep learning techniques used in the proposed structured deep learning detector.

4.5 Mean field inference

Instead of computing the posterior directly we make use of variational inference (VI) to iteratively approximate the posterior. This allows us to leverage established techniques in VI to come up with an efficient and scalable algorithm. We begin by providing a brief overview.

Mean field inference approximates the posterior with a factorized distribution of the form

$$q(\mathbf{x}) = \prod_i q_i(x_i). \tag{4.49}$$

It has a well defined sequential update rule that is guaranteed to converge due to the convexity of the objective function with respect to each factor q_i [129]. It can be applied to very general functions and convergent parallel update algorithms have been developed to speed up inference time [133, 134].

First we define the expression that relates the evidence, the evidence lower bound (ELBO) and the KL divergence.

$$\log p(\mathbf{y}) = \mathcal{L}(q) + D_{\mathsf{KL}}(q\|p) \tag{4.50}$$

where

$$\mathcal{L}(q) = \mathbb{E}_q\left[\log \frac{p(\mathbf{y},\mathbf{x})}{q(\mathbf{x})}\right] \tag{4.51}$$

$$D_{\mathsf{KL}}(q\|p) = \mathbb{E}_q\left[\log \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})}\right]. \tag{4.52}$$

We proceed to approximate q by maximizing the ELBO (4.51), this is equivalent to minimizing

$$\mathcal{F}(q) = \underbrace{-\mathbb{E}_q[\log p(\mathbf{y},\mathbf{x})]}_{\mathcal{E}(q)} + \underbrace{\mathbb{E}_q[\log q(\mathbf{x})]}_{-\mathcal{H}(q)}. \tag{4.53}$$

$\mathcal{F}$ is called the variational free energy. The first term is the average energy and the second term is the negative entropy. The negative entropy can be viewed as a regularizer that enforces some minimum amount of uncertainty in q. Sequential updates monotonically decrease $\mathcal{F}$ at each step.

4.5.1 Proximal gradient descent

In this section, we outline a connection between damped mean field inference and a specific form of proximal gradient descent [133], the specific instance is called the composite objective mirror descent [135]. This connection is used to justify damping as a means to slow down the movement of the iterates and to provide a formal argument to guarantee convergence in the case of parallel MF inference.

Note that (4.53) is a composite objective function of the form

$$f(\mathbf{x}) + r(\mathbf{x}). \tag{4.54}$$

where r is a regularizing function. Proximal gradient descent defines an iterative algorithm in which each step solves a minimization sub-problem

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \nabla f(\mathbf{x}^t)\rangle + r(\mathbf{x}) + \lambda\Psi(\mathbf{x}, \mathbf{x}^t) \tag{4.55}$$

where $\Psi(\cdot,\cdot)$ is a non negative proximal function that satisfies $\Psi(\mathbf{x}, \mathbf{x}^t) = 0$ if and only if $\mathbf{x} = \mathbf{x}^t$. Here we are minimizing a first order approximation of f while keeping the next iterate close relative to $\mathbf{x}^t$, all the while preserving the regularizing property of r. A judicial choice of Ψ allows us to capture the natural geometry of the domain $\mathcal{X}$. It turns out that when we consider the problem of estimating distributions q, a suitable proximal function would be the KL divergence. We note that λ is intimately related to the inverse

of the step size.

4.5.2 Application to mean-field inference

Recall that our objective is to minimize the variational free energy:

$$\min_{q \in \mathcal{Q}} \mathcal{F}(q). \tag{4.56}$$

When we replace f with $\mathcal{E}$ and r with $-\mathcal{H}$, we obtain the composite form. We now designate Ψ as the KL divergence with parameter d:

$$\Psi(q, q^t) = dD_{\mathsf{KL}}(q \| q^t). \tag{4.57}$$

Note that Ψ is a valid proximal function. This leads to the proximal update rule:

$$q^{t+1} = \arg\min_{q \in \mathcal{Q}} \langle q, \nabla \mathcal{E}(q^t) \rangle - \mathcal{H}(q) + dD_{\mathsf{KL}}(q \| q^t). \tag{4.58}$$

The minimization has a closed form expression and is given by

$$\log q_i^{t+1}(x_i) = \alpha \, \mathbb{E}_{j \neq i}[\log p(\mathbf{x}, \mathbf{y})] + (1 - \alpha) \log q_i^t(x_i) + \text{const}_i \tag{4.59}$$

where the expectation is w.r.t. $\prod_{j \neq i} q_j(x_j)$ and $\alpha = 1/(1 + d)$.

Remark 7. A convergence quarantee for parallel MF can be established if the parameterized KL term is L-strongly convex and L is the Lipschitz constant of $\nabla \mathcal{E}$.

Remark 8. Since we have shown that damping limits the movement of the next iterate q^{t+1} from q^t, it can be used additionally as a way to avoid getting trapped in an extreme point in a non-convex constraint set. We will show empirically in section 4.8 that damping

significantly improves the performance of MFI such that it outperforms OBMNet in all SNR regions.

4.6 Mean field inference method

In this section, we introduce the techniques used in the proposed algorithm.

4.6.1 Sequential MF update

We consider symbols drawn uniformly from the PAM constellation, $x_i \in \mathcal{X}$. The coordinate descent algorithm updates each factor q_i sequentially while holding all other factors fixed. This leads to the following update rule for sequential mean field update

$$\log q_i^*(x_i) = \mathbb{E}_{j \neq i}[\log p(\mathbf{y}|\mathbf{x})] + \text{const.} \tag{4.60}$$

This defines the sequential update procedure and it is guaranteed to converge to a local minimum since the ELBO is convex w.r.t. each factor q_i [129]. This coordinate descent procedure requires re-evaluating the expectation every time each q_i is updated, thus it is computationally intensive, not amenable to parallelization and not scalable as the expectation computation has complexity exponential in K. We address this scalability issue in section 4.6.4.

The MFI algorithm was able to attain ML performance at low SNR but performance quickly degraded at high SNR. We suspect in the high SNR region, the objective surface has many local minima and the algorithm is being trapped in one. Next, we employ a number of techniques to improve the optimization landscape.

4.6.2 Sigmoid approximation

Using the insight gained in section 4.3.3, we replace the exact likelihood with the sigmoid approximation. The sigmoid approximation has the same effect of flattening the objective surface, leading to reduced effective step sizes for the mean field updates. Indeed we observed improved performance in the high SNR region. However, the next technique is also required to push the performance beyond that of OBMNet.

4.6.3 Damping

Optimization perspective

The fully factorized distribution used in naive mean field approximation leads to a non-convex constraint set $\mathcal{M}_F(G)$ [136]. The extreme points of the set corresponds to delta distributions that place all its mass on some $\mathbf{x}$. Since this set is a proper subset of the full polytope $\mathcal{M}(G)$, there may not exist a convex combination between two extreme points. Coordinate descent algorithms that move too quickly to one extreme point in the set will remained trapped there. In our simulations, we observed that MF inference often arrive at a delta distribution in a single iteration. Although one of these delta distributions or extreme points may correspond to the global minimum, we argue that at high SNR scenario, the fast movement along any coordinate frequently traps the algorithm to a non-optimal extreme point.

One technique that limits the movement of the iterate is damping. Damping has been extensively used as a heuristic in many settings [137–140]. In section 4.5.2, we highlighted that damped mean field inference is equivalent to proximal gradient descent where the proximal function is the KL divergence limiting how far the next iterate can move with respect to the current iterate. This limited gradient descent movement allows the iterates to be directed to the global optimum without being trapped.

Specifically, we control movement of the iterates with a damping factor α, such that the new update has the form

$$\overline{m}_i^{(t)}(x_i) = (1-\alpha)\overline{m}_i^{(t-1)}(x_i) + \alpha m_i(\log p(\mathbf{y}|\mathbf{x})) \tag{4.61}$$

where $m_i, \overline{m}_i \in \mathbb{R}^{M^{1/2}}$ are the current MF update message and the damped MF update message respectively. Equation (4.61) is equivalent to (4.59) without the normalization constant. We intentionally change the notation since the expectation will be replaced with an approximation in the final algorithm. Although damping is a technique typically used to guarantee convergence of parallel MF updates, this is an essential element to produce superior performance even when sequential MF updates are used.

Complexity reduction and inference speed are considered in the following subsections.

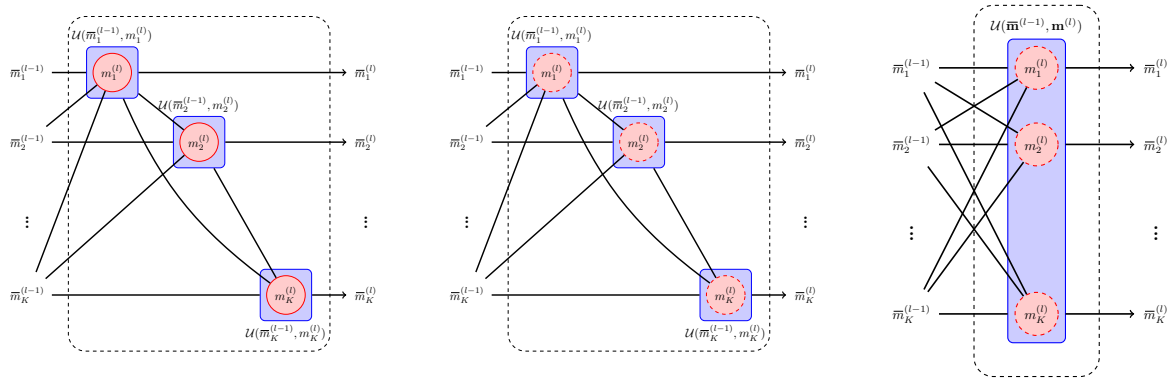
4.6.4 Stochastic estimate of the expectation

Instead of computing the high dimensional expectation in (4.60), we can draw samples from $q(x)$ and approximate this with the average of the log likelihood over these samples. Thus we approximate the expectation by

$$\mathbb{E}_{j\neq i}[\log p(\mathbf{y}|\mathbf{x})] \approx \sum_{s=1}^{S} \log p(\mathbf{y}|\mathbf{x}^{(s)}), \quad x_j^{(s)} \sim q_j(x_j),\ j \neq i \tag{4.62}$$

where $\mathbf{x}^{(s)} = [x_1^{(s)}, ..., x_{i-1}^{(s)}, x_i, x_{i+1}^{(s)}, ..., x_K^{(s)}]$.

The sampling method reduces the number of operations from $\mathcal{O}(KM^K)$ to $\mathcal{O}(KM^{1/2}S)$.



(a) (b) (c)

Figure 4.5: Structure of a single layer of (a) MFNet, (b) SMFNet, and (c) PSMFNet. The expectation is represented as a solid circle. The stochastic average is represented as a dashed circle. Serial processing is depicted by the dependence of the nodes with the layer.

4.6.5 Parallel MF update

Despite the serial nature of sequential MF, many practitioners opt to implement the MF updates in parallel. This provided a speed advantage over sequential updates. However, the convergence guarantee is no longer valid in this setting. We note that damped parallel MF updates in the natural parameter space has the desirable property that it converges when the damping induced proximal term is L-strongly convex where L is the Lipschitz constant of the energy gradient, $\nabla\mathcal{E}$ [133]. Thus, parallel MF can be implemented to obtain a speed advantage with no added cost since damping is already exploited to improve the accuracy of the detector.

4.7 Deep learning MF detector

4.7.1 Mean field network

For each layer l, the damped MF update can be expressed as

$$\overline{m}_i^{(l)} = m_i^{(l)} + \overline{\alpha}(\overline{m}_i^{(l-1)} - m_i^{(l)}). \tag{4.63}$$

For parallel MF update, the messages can be grouped together, leading to

$$\overline{\mathbf{m}}^{(l)} = \mathbf{m}^{(l)} + \overline{\alpha}(\overline{\mathbf{m}}^{(l-1)} - \mathbf{m}^{(l)}) \tag{4.64}$$

where $\mathbf{m}, \overline{\mathbf{m}} \in \mathbb{R}^{2KM^{1/2}}$.

When $\overline{\alpha} = 0$, it is standard MF without damping. When $\overline{\alpha} = 1$, the algorithm retains the value from the last iteration. Note also that $\overline{\alpha} = 1 - \alpha$.

In our experiments, we observed that α is SNR dependent, thus we first define a network that learns α as a function of σ^2. Specifically, we define a parameterized function

$$\overline{\alpha}(\sigma^2) = \overline{\sigma^2} + f(\overline{\sigma^2}) \tag{4.65}$$

where f is a one-layer MLP with 8 hidden nodes and $\overline{\sigma^2} = 1 - \sigma^2$. Next, we replace the second term in (4.64) with a trainable function $H(\cdot)$, which has the following form

$$H(\overline{\mathbf{m}} - \mathbf{m}, \sigma^2) = \overline{\alpha}(\sigma^2) g(\overline{\mathbf{m}} - \mathbf{m}) \tag{4.66}$$

where g is a residual layer, i.e. $g(\mathbf{x}) = \mathbf{x} + h(\mathbf{x})$. Note that g is a set that includes of the identity map. The specific structure of the residue network $h(\cdot)$ is described in Appendix 4.10.2.

Note that $H(\cdot)$ is constructed such that MF and damped MF can be recovered from specific choices of H. For example, we can recover (4.64) by setting $H(\mathbf{x}) = \bar{\alpha}\mathbf{x}$.

Remark 9. In our experiments, the value of the noise variance is truncated to $[a, 1]$ before it is fed to $\bar{\alpha}(\cdot)$. The exact value of a is determined through cross validation.

The network layer specific to each of the MFNet variants are illustrated in Fig. 4.5. The MFNet performs sequential processing and fully computes the expectation for each mean field update. The sampling MFNet (SMFNet) replaces the expectation with a stochastic average. The Parallel SMFNet (PSMFNet) samples once based on the incoming messages $m^{(l-1)}$, and then computes the mean field updates in parallel at each layer. Note that the update function $\mathcal{U}(\cdot, \cdot)$ in the diagram encapsulates the input-output relationship defined in (4.63) and (4.64).

4.7.2 Choice of loss function

Since the MFNet returns an estimate at every layer, it is possible to apply the cross entropy loss to every layer. The shorter gradient paths to the initial layers can speed up convergence during training. This technique is used in many DNN applications based on unrolling [50, 52, 53]. However, CE penalizes low confidence estimates heavily [141], thus the penalty it assigns to early layers means that a drastic movement is made early to move away from a low confident estimate q. This is undesirable as our goal is to restrict early movements made by the MF updates. In Fig. 4.6, the damping factor that produced the lowest BER, the lowest cross entropy loss at the output layer and lowest cross entropy sum over all layers are shown. It can be seen that using cross entropy loss at the output layer produced a result more aligned with BER performance. Precisely we define the loss function as

$$L(q^{(L)}) = \frac{1}{2BK} \sum_{i=1}^{B} \sum_{k=1}^{2K} CE(p_{i,k}, q_{i,k}^{(L)}) \tag{4.67}$$

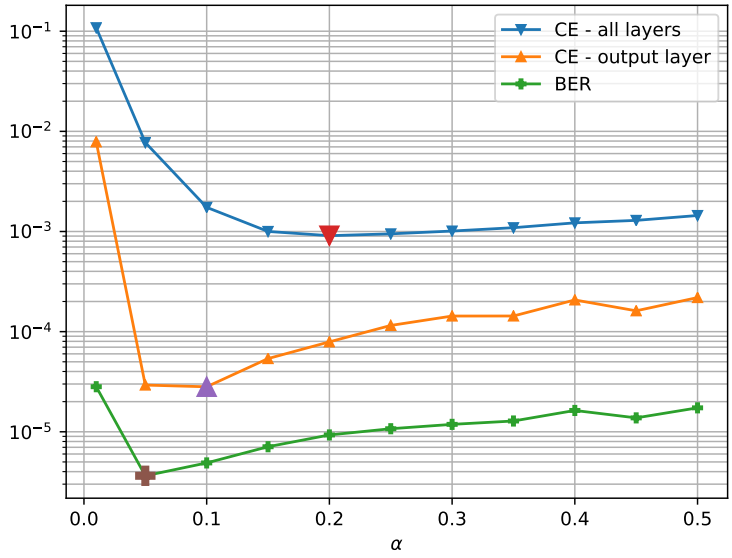


Figure 4.6: Comparison of the minimum of BER vs the summed cross-entropy of all layers and the cross entropy of the output layer only.

where B is the batch size, $p_{i,k}$, $q_{i,k}^{(L)}$ are the true label and the predicted distribution for the kth user in the ith sample respectively. The predicted distribution can be computed based on

$$q_j(x_j) = \frac{e^{m_j(x_j)}}{\sum_{x \in \mathcal{X}} e^{m_j(x)}}. \tag{4.68}$$

4.7.3 Training

The data samples are generated as follows. Random bits are generated and mapped to symbols $\bar{\mathbf{x}}$ in the designated QAM constellation. The elements of the channel matrix are generated randomly for every channel use. The random channel $\overline{\mathbf{H}}$ is applied to the transmitted symbol and AWGN noise is added. The noise corrupted signal is then 1-bit quantized to produce the observation $\bar{\mathbf{y}}$. The real valued inputs $\mathbf{x}_i$, $\mathbf{H}_i$ and outputs $\mathbf{y}_i$ form the data set. New samples are generated on the fly so training data are never reused. The networks are trained across the full range of SNRs uniformly in the log domain.

Tensorflow [142] is used to develop the deep learning flow. The unfolded network is trained end-to-end with the joint cross entropy loss. Each epoch is defined to be 100 minibatches and each minibatch consists of 1024 samples. The network is trained for a duration of 40 epochs. The SGD optimizer with Nesterov acceleration [25] is used with learning rate 1e-2 and momentum is set to 0.9. The learning rate is reduced by a factor of 10 at epoch indices $\{20, 30\}$.

In our experiments, we observed that the $\alpha(\cdot)$ function alone can converge to an optimal function each time. However, using the more general function H, i.e. including the network $g(\cdot)$, training almost always settled into a bad local minimum unless the network was initialized close to a good minimum. This observation reveals that bad local minima are prevalent in these type of networks. It is left as a future investigation to examine modifications that can eliminate bad local minima [143, 144] for complex composite objective such as those we encountered here.

Recall that the proposed algorithm requires sampling at each layer. Without damping, gradient must be back propagated through the sampling operation. However, we note that sampling a discrete distribution is a non-differentiable operation. Continuous relaxation of the discrete distribution with a known reparameterization were first developed in [145, 146] and later refined in [147, 148]. We experimented with the Gumbel-Softmax distribution but ultimately settled on a solution that stopped the gradient from propagating through the sampling operation. This is possible because damping introduces an alternate gradient path for back propagation between the layers.

4.7.4 Complexity

The complexity of MFNet is of order $\mathcal{O}(M^K K^2 NL)$. For PMFSNet and MFSNet it is $\mathcal{O}(M^{1/2} SK^2 NL)$. The complexity of PMFSNet and MFSNet are linear in the number of antennas N and the number of layers L, and quadratic in the number of users K. We

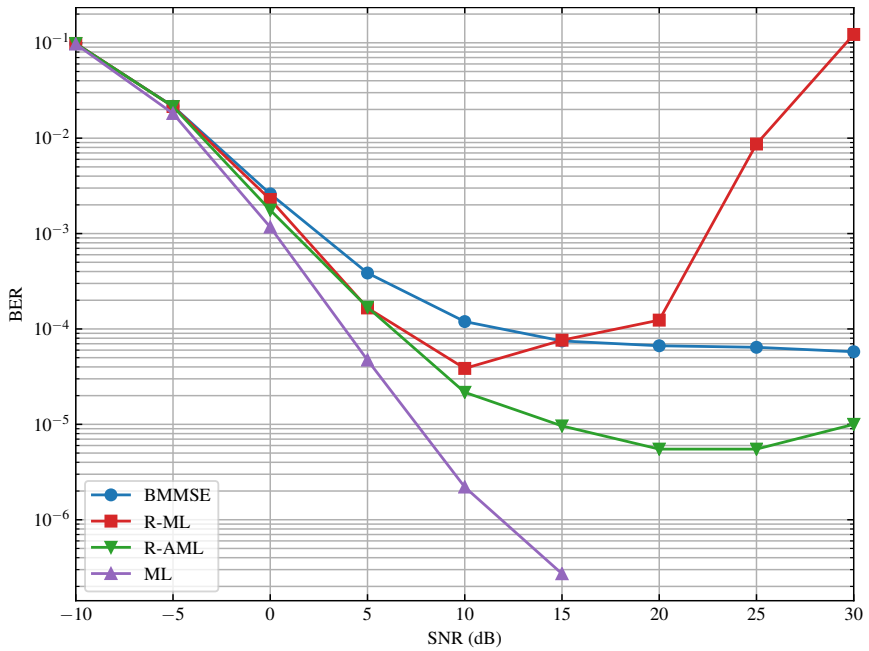


Figure 4.7: BER performance of the relaxed-ML and relaxed-AML gradient descent algorithms. $N = 32, K = 4$ and $M = 4$ (QPSK).

note that PMFSNet computes the updates in parallel thus it has a speed advantage over MFSNet.

4.8 Numerical results

The BER performance of the relaxed gradient descent algorithms is evaluated. The results for relaxed-ML (R-ML) and relaxed-AML (R-AML) are shown in Fig. 4.7. The elements in $\mathbf{H}$ are assumed i.i.d. with distribution $\mathcal{CN}(0, 1)$. The large gradient from R-ML causes the algorithm to diverge at high SNR, leading to substantially degraded performance beyond 20dB. Whereas R-AML benefited from the constant gradient and shows a more robust performance even at 30dB SNR. However, there remains a gap between these algorithms and ML at low SNR.

The performance of all variational methods are compared in Fig. 4.8. VB only achieved a slight performance advantage over BMMSE. Whereas variational lower bound

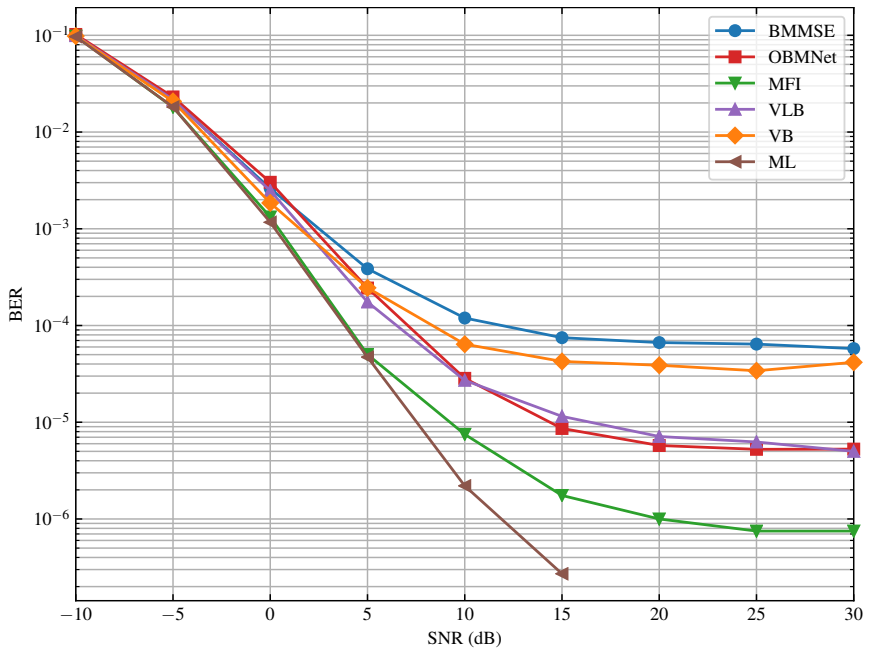


Figure 4.8: BER performance of variational Bayes (VB), variational lower bound (VLB) and mean field inference (MFI) are compared against reference BMMSE and ML detectors. $N = 32, K = 4$ and $M = 4$ (QPSK).

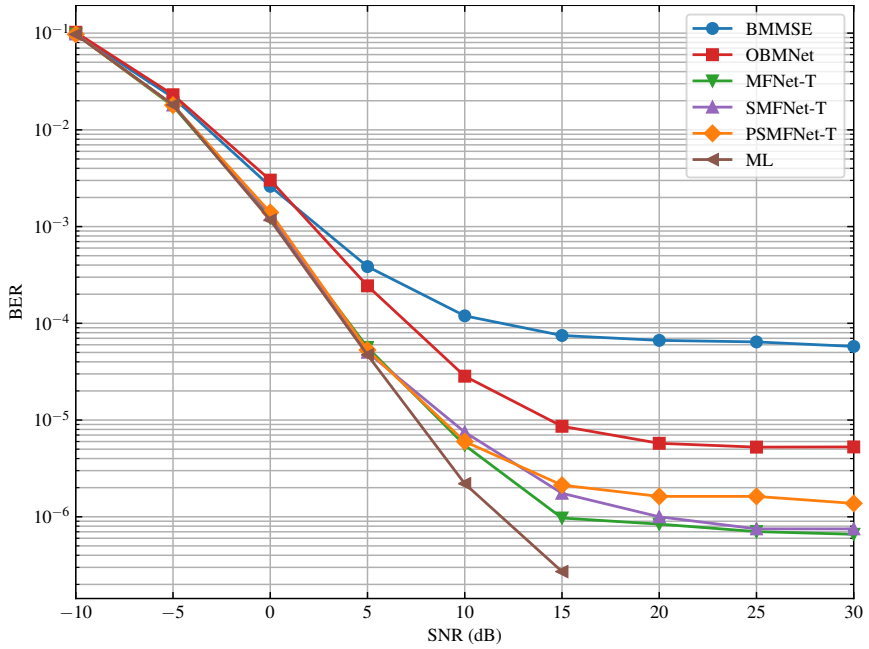


Figure 4.9: Performance comparison of the MFNet, sampling MFNet (SMFNet) and the parallel SMFNet (PSMNet) using the same set of hand tuned damping parameters. $N = 32, K = 4$, and $M = 4$ (QPSK), The parameters of the MFNets are set to $S = 10, L = 10$.

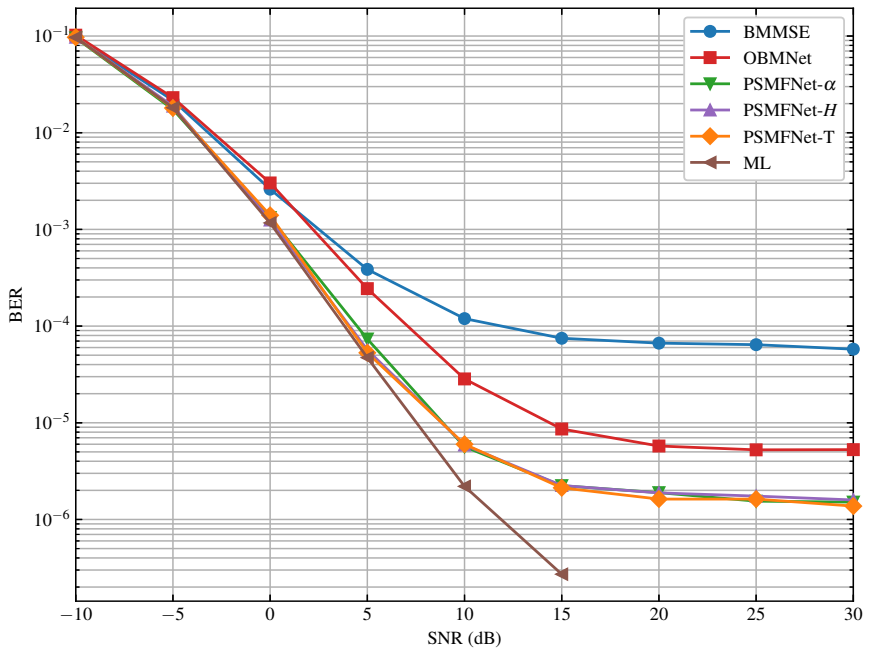


Figure 4.10: Comparison of learned PSMFNet variants (PSMFNet-α, PSMFNet-H) against the hand-tuned PSMFNet (PSMFNet-T). $N = 32, K = 4$, and $M = 4$ (QPSK), The parameters of the MFNets are set to $S = 10, L = 10$.

(VLB) achieved comparable performance to the OBMNet. The best performance was achieved with mean field inference (MFI), it was able to attain ML performance at low SNR and outperformed all other algorithms in all SNRs.

We compare the performance of mean field inference when sampling and parallel inference techniques are employed. In Fig. 4.9, the full expectation computation was carried out in MFNet. SMFNet introduced sampling, and PSMFNet perform parallel update using the sampled average. Incorporating sampling and parallel updates have little impact to performance at low SNR. In the high SNR region, sampling causes a small loss compared to full expectation, however the number of likelihood evaluations is reduced from M^K to $M^{1/2}S$. Parallel update (PMSNet) degrades the performance slightly compared to serial update, however it still outperformed all non MF algorithms and it has a speed advantage compared to MF algorithms using serial updates.

Two deep learning PSMFNet configurations are compared with the hand-tuned

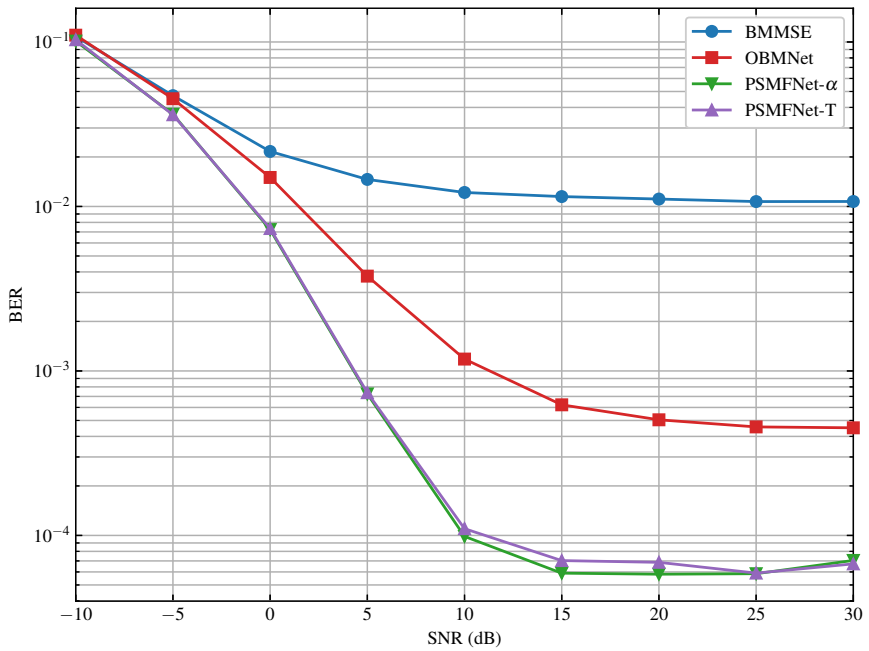


Figure 4.11: Performance comparison of the hand-tuned PSMFNet (PSMFNet-T) and the learned PSMFNet (PSMFNet-α) against existing methods for $K = 8$, $N = 128$ and $M = 16$ (16QAM), The parameters of the PSMFNet is set to $S = 32$ and $L = 15$.

PSMFNet (PSMFNet-T) in Fig. 4.10. The hand-tuned version selects the best possible damping factor at each SNR from a set of values. The PSMFNet-α learns the damping factor as a function of noise variance; the PSMFNet-H combines a scaling network and a network that transforms the messages. These two configurations correspond to eqns. (4.65) and (4.66) respectively. We note that both PSMFNet-α and PSMFNet-H are able match the performance of the hand-tuned version. Lastly, we demonstrate the scalability of PSMFNet in Fig. 4.11. At 16QAM, $K = 8$ and $N = 128$, the reduced complexity PSMFNet continues to outperform OBMNet across all SNRs.

4.9 Conclusion

In this work, we evaluated existing model based approaches for 1-bit data detection, revealed the shortcomings of iterative methods that rely on the exact likelihood and

identified the properties that make the sigmoid approximation more robust. We next developed a number of algorithms based on variational methods and demonstrated that damped mean field inference produced superior performance compared to the state of the art. We focused on an algorithm based on stochastic variational inference to reduce the complexity of naive mean field inference and introduced parallel processing to speed up inference time without compromising the performance of the detector. The unfolded algorithm trained via deep learning was able to achieve the performance of a hand-tuned network, showing the viability of using deep learning to optimize the algorithm. The final proposed algorithm achieved near ML performance in the low SNR regime, the SNR region of interest for massive MIMO systems.

Chapter 4, in part, is a reprint of material that is currently being prepared for submission for publication to be identified as: D. K. W. Ho, M. Welling, B. D. Rao, "Structured Neural 1-bit Massive MIMO Detector Based on Stochastic Variational Inference". The dissertation author was the primary investigator and author of this material.

4.10 Appendix

4.10.1 Variational Bayes and EM-MAP

Model

Given the 1-bit model

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n}, \quad \mathbf{y} = \mathcal{Q}(\mathbf{r}). \tag{4.69}$$

The joint likelihood has the form

$$p(\mathbf{y}, \mathbf{r}|\mathbf{x}) = \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) p(\mathbf{r}|\mathbf{x}) \tag{4.70}$$

where

$$p(\mathbf{r}|\mathbf{x}) = \frac{1}{|2\pi\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\Big\{-\frac{1}{2}(\mathbf{r}-\mathbf{H}\mathbf{x})^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{r}-\mathbf{H}\mathbf{x})\Big\}. \tag{4.71}$$

Assume the prior is Gaussian with distribution, $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \frac{1}{2}\mathbf{I})$, i.e.

$$p(\mathbf{x}) = \frac{1}{\pi^K} \exp\big\{-\|\mathbf{x}\|^2\big\}. \tag{4.72}$$

EM-MAP

In the EM-MAP algorithm, given an initial estimate $\widehat{\mathbf{x}}$, the E-step computes the expected log likelihood.

$$\begin{aligned}
\mathcal{Q}(\mathbf{x}, \widehat{\mathbf{x}}) &= \mathbb{E}_{\mathbf{r}|\widehat{\mathbf{x}},\mathbf{y}}[\log p(\mathbf{y}, \mathbf{r}, \mathbf{x})] && (4.73)\\
&= \mathbf{x}^{\mathsf{T}}\mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{\mathbf{r}|\mathbf{y},\widehat{\mathbf{x}}}[\mathbf{r}] - \frac{1}{2}\mathbf{x}^{\mathsf{T}}(2\mathbf{I} + \mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{H})\mathbf{x} + \text{const} && (4.74)\\
&= \mathbf{x}^{\mathsf{T}}\mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{H}\widehat{\mathbf{x}} + \mathbb{E}_{\mathbf{n}|\mathbf{y},\widehat{\mathbf{x}}}[\mathbf{n}]) - \frac{1}{2}\mathbf{x}^{\mathsf{T}}(2\mathbf{I} + \mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{H})\mathbf{x} + \text{const}. && (4.75)
\end{aligned}$$

The M-step now maximizes the posterior of a Gaussian linear model with expected quantizer input

$$\widehat{\mathbf{r}} \triangleq \mathbf{H}\widehat{\mathbf{x}} + \mathbb{E}_{\mathbf{n}|\mathbf{y},\widehat{\mathbf{x}}}[\mathbf{n}]. \tag{4.76}$$

Note that $\mathbf{n}|\mathbf{y}, \widehat{\mathbf{x}}$ is a multivariate truncated Gaussian random vector. Thus we have

$$\widehat{\mathbf{x}}^{\text{new}} = (2\mathbf{I} + \mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\mathbf{H})^{-1}\mathbf{H}^{\mathsf{T}}\boldsymbol{\Sigma}^{-1}\widehat{\mathbf{r}}. \tag{4.77}$$

Variational Bayes

We now consider the Variational Bayes framework. We approximate the posterior with a factored distribution

$$q(\mathbf{r}, \mathbf{x}) = q(\mathbf{r})q(\mathbf{x}) \tag{4.78}$$

Given $q(\mathbf{x})$, the mean field update for $q(\mathbf{r})$

$$\log q(\mathbf{r}) = \mathbb{E}_{q(\mathbf{x})}[\log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) + \log p(\mathbf{r}|\mathbf{x})p(\mathbf{x})] \tag{4.79}$$

$$= \log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) + \mathbb{E}_{q(x)}[\mathbf{x}^\mathsf{T}]\mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{r} - \frac{1}{2}\mathbf{r}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{r} + \text{const.} \tag{4.80}$$

From inspection, $q(\mathbf{r})$ is a truncated gaussian centered at $\mathbf{H}\widehat{\mathbf{x}}$, with $\widehat{\mathbf{x}} = \mathbb{E}_{q(\mathbf{x})}[\mathbf{x}]$.

Given $q(\mathbf{r})$, the update equation for $q(\mathbf{x})$ is

$$\log q(\mathbf{x}) = \mathbb{E}_{q(\mathbf{r})}[\log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) + \log p(\mathbf{r}|\mathbf{x})p(\mathbf{x})] \tag{4.81}$$

$$\stackrel{a}{=} \mathbb{E}_{q(\mathbf{r})}[\log p(\mathbf{r}|\mathbf{x})p(\mathbf{x})] \tag{4.82}$$

$$= \mathbf{x}^\mathsf{T}\mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{q(\mathbf{r})}[\mathbf{r}] - \frac{1}{2}\mathbf{x}^\mathsf{T}(2\mathbf{I} + \mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{H})\mathbf{x} + \text{const} \tag{4.83}$$

$$= \mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_\mathbf{x} - \frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Sigma}_\mathbf{x}^{-1}\mathbf{x} + \text{const} \tag{4.84}$$

where (a) follows from the fact that the support of $q(\mathbf{r})$ is contained in $\mathcal{Q}^{-1}(\mathbf{y})$. Let $\widehat{\mathbf{r}} = \mathbb{E}_{q(\mathbf{r})}[\mathbf{r}]$. The quadratic form gives rise to a gaussian factor $q(\mathbf{x})$

$$q(\mathbf{x}) \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_\mathbf{x}) \tag{4.85}$$

where

$$\boldsymbol{\mu}_\mathbf{x} = \boldsymbol{\Sigma}_\mathbf{x}\mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\widehat{\mathbf{r}} \tag{4.86}$$

$$\boldsymbol{\Sigma}_\mathbf{x}^{-1} = 2\mathbf{I} + \mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{H}. \tag{4.87}$$

Fully factored distribution

Since the expectation $\mathbb{E}_{q(\mathbf{r})}[\mathbf{r}]$ does not admit a closed form, we approximate $q(\mathbf{r})$ with a fully factored distribution $\prod_i q(r_i)$. Define the following representations for $\boldsymbol{\Lambda}$ and

H

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_{11} & \boldsymbol{\lambda}_{12}^{\mathsf{T}} \\ \boldsymbol{\lambda}_{21} & \mathbf{\Lambda}_{22} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_1^{\mathsf{T}} \\ \mathbf{H}_2 \end{bmatrix}. \tag{4.88}$$

Let $\boldsymbol{\mu} = \mathbf{H}\mathbf{x}$. We have

$$\log q(r_1) \tag{4.89}$$

$$= \mathbb{E}_{q(\mathbf{r}_{\setminus 1})q(\mathbf{x})}[\log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) + \log p(\mathbf{r}|\mathbf{x})p(\mathbf{x})] \tag{4.90}$$

$$= \mathbb{E}_{q(\mathbf{r}_{\setminus 1})q(\mathbf{x})}\Big[\log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) - \frac{1}{2}(\mathbf{r}-\boldsymbol{\mu})^{\mathsf{T}}\mathbf{\Lambda}(\mathbf{r}-\boldsymbol{\mu})\Big] + \text{const} \tag{4.91}$$

$$= \mathbb{E}_{q(\mathbf{r}_{\setminus 1})q(\mathbf{x})}\Big[\log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) - \frac{1}{2}r_1^2\lambda_{11} + r_1\lambda_{11}\mu_1 - r_1\boldsymbol{\lambda}_{12}^{\mathsf{T}}(\mathbf{r}_{\setminus 1} - \boldsymbol{\mu}_{\setminus 1})\Big] + \text{const} \tag{4.92}$$

$$= \mathbb{E}_{q(\mathbf{r}_{\setminus 1})q(\mathbf{x})}\Big[\log \mathbb{1}_{\mathcal{Q}^{-1}(\mathbf{y})}(\mathbf{r}) - \frac{1}{2}r_1^2\lambda_{11} + r_1\lambda_{11}\big(\mu_1 - \lambda_{11}^{-1}\boldsymbol{\lambda}_{12}^{\mathsf{T}}(\mathbf{r}_{\setminus 1} - \boldsymbol{\mu}_{\setminus 1})\big)\Big] + \text{const} \tag{4.93}$$

$$\overset{a}{=} \log \mathbb{1}_{\mathcal{Q}^{-1}(y_1)}(r_1) - \frac{1}{2}r_1^2\lambda_{11} + r_1\lambda_{11}\Big(\mathbb{E}_{q(\mathbf{x})}[\mu_1] - \lambda_{11}^{-1}\boldsymbol{\lambda}_{12}^{\mathsf{T}}(\mathbb{E}_{q(\mathbf{r}_{\setminus 1})}[\mathbf{r}_{\setminus 1}] - \mathbb{E}_{q(\mathbf{x})}[\boldsymbol{\mu}_{\setminus 1}])\Big) + \text{const} \tag{4.94}$$

where (a) follows from the fact that the support of $q(r_i)$ are contained in $\mathcal{Q}^{-1}(y_i)$. From inspection, $q(r_1)$ is a truncated univariate Gaussian with

$$q(r_1) \sim \mathcal{TN}(m_1, \lambda_{11}^{-1}, \mathcal{Q}^{-1}(y_1)) \tag{4.95}$$

where

$$m_1 = \mathbf{h}_1^{\mathsf{T}}\,\mathbb{E}[\mathbf{x}] - \lambda_{11}^{-1}\boldsymbol{\lambda}_{12}^{\mathsf{T}}(\mathbb{E}_q[\mathbf{r}_{\setminus 1}] - \mathbf{H}_2\,\mathbb{E}_q[\mathbf{x}]). \tag{4.96}$$

Suppose the noise is AWGN with precision matrix $\mathbf{\Lambda} = \frac{2}{\sigma^2}\mathbf{I}$. Then $q(r_i)$ reduces to

$$q(r_i) \sim \mathcal{TN}(m_i, \frac{\sigma^2}{2}, \mathcal{Q}^{-1}(y_1)) \tag{4.97}$$

where

$$m_i = \mathbf{h}_i^{\mathsf{T}}\,\mathbb{E}[\mathbf{x}]. \tag{4.98}$$

The mean of r_i can be computed as

$$\mathbb{E}[r_i] = m_i + \frac{y_i \phi(s_i)}{\Phi(y_i s_i)} \sigma_i, \quad s_i = \frac{m_i}{\sigma_i}. \tag{4.99}$$

Expectation propagation

The optimal distribution is the marginal of a truncated multivariate gaussian.

$$q^*(r_i) = \int p(\mathbf{r}) \prod_{j \neq i} dr_i = p(r_i). \tag{4.100}$$

The elements of a truncated Gaussian are independent if the covariance matrix has zero off-diagonal entries, $\Sigma_{ij} = 0$ for $i \neq j$. In fact, for independent truncated Gaussians, the marginal distribution is simply a Gaussian $\mathcal{N}(\mu_i, \Sigma_{ii})$ truncated at $Q^{-1}(y)$ [149, Corollary 7],

Truncated Gaussian source

Assume $\mathbf{x}$ is a truncated Gaussian, the underlying Gaussian has pdf $f(\mathbf{x})$, then it has distribution

$$p(\mathbf{x}) = \frac{1}{Z} \mathbb{1}_A(\mathbf{x}) f(\mathbf{x}) \tag{4.101}$$

where $Z = Z(\boldsymbol{\mu}, \boldsymbol{\Sigma}, A)$.

Let the underlying Gaussian have pdf

$$f(\mathbf{x}) = \frac{1}{\pi^K} \exp\{-\|\mathbf{x}\|^2\} \tag{4.102}$$

i.e. the non-truncated $\mathbf{x} \in \mathbb{R}^{2K}$ has distribution $p(\mathbf{x}) \sim \mathcal{N}(\mathbf{0}, \frac{1}{2}\mathbf{I})$.

The joint density has the form

$$\log p(\mathbf{y}, \mathbf{r}|\mathbf{x})p(\mathbf{x}) \tag{4.103}$$

$$= \log \mathbb{1}_{Q^{-1}(\mathbf{y})}(\mathbf{r}) + \log p(\mathbf{r}|\mathbf{x}) + \log f(\mathbf{x}) + \log \mathbb{1}_A(\mathbf{x}) - \log Z. \tag{4.104}$$

Given $q(\mathbf{r})$ then we have the following expression

$$\log q(\mathbf{x}) = \mathbb{E}_{q(\mathbf{r})}[\log p(\mathbf{y}, \mathbf{r}|\mathbf{x})p(\mathbf{x})] \tag{4.105}$$

$$= \mathbb{E}_{q(\mathbf{r})}\big[\log \mathbb{1}_{Q^{-1}(\mathbf{y})}(\mathbf{r}) + \log p(\mathbf{r}|\mathbf{x}) + \log f(\mathbf{x}) + \log \mathbb{1}_A(\mathbf{x})\big] + \text{const} \tag{4.106}$$

$$\stackrel{a}{=} \mathbb{E}_{q(\mathbf{r})}\big[\log p(\mathbf{r}|\mathbf{x}) + \log f(\mathbf{x}) + \log \mathbb{1}_A(\mathbf{x})\big] + \text{const} \tag{4.107}$$

$$= \log \mathbb{1}_A(\mathbf{x}) + \mathbf{x}^\mathsf{T}\mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\,\mathbb{E}_{q(\mathbf{r})}[\mathbf{r}] - \frac{1}{2}\mathbf{x}^\mathsf{T}(2\mathbf{I} + \mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{H})\mathbf{x} + \text{const} \tag{4.108}$$

$$= \log \mathbb{1}_A(\mathbf{x}) + \mathbf{x}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\mu} - \frac{1}{2}\mathbf{x}^\mathsf{T}\boldsymbol{\Lambda}\mathbf{x} + \text{const}. \tag{4.109}$$

From inspection, $q(\mathbf{x})$ is a truncated Gaussian with $q(\mathbf{x}) = \mathcal{TN}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}, A)$ where

$$\boldsymbol{\mu} = \boldsymbol{\Lambda}^{-1}\mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\widehat{\mathbf{r}} \tag{4.110}$$

$$\boldsymbol{\Lambda} = 2\mathbf{I} + \mathbf{H}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{H}. \tag{4.111}$$

We convert this to a fully factorized distribution $q(\mathbf{x}) = \prod_i q(x_i)$. Suppose A is the Cartesian product $[a, b]^{2K}$, then $q(x_i)$ becomes a truncated Gaussian with

$$q(x_i|m_i, \lambda_{i,i}^{-1}, [a, b]) \tag{4.112}$$

where

$$m_i = \mu_i - \lambda_{i,i}^{-1}\boldsymbol{\lambda}_{i,\setminus i}^\mathsf{T}(\mathbb{E}_q[\mathbf{x}_{\setminus i}] - \boldsymbol{\mu}_{\setminus i}) \tag{4.113}$$

and $\boldsymbol{\lambda}_{i,\setminus i}^\mathsf{T} = [\lambda_{i,i}, \lambda_{i,i-1}, \lambda_{i,i+1}, \lambda_{i,2K}]$.

4.10.2 Residue module architecture

Table 4.1 describes the architecture of $h(\cdot)$ in (4.66). The last column indicates the input/output dimension in parentheses and the dimension of the weight matrix for the dense layers in brackets.

Table 4.1: Architecture of the residue module, $2K = 8$, $M^{1/2} = 2$.

layer name	output size	MLP
input	-	$(8, 2)$
batch normalization	16	-
dense layer	48	$[16 \times 48]$
batch normalization	48	-
RELU	48	-
dense layer	48	$[48 \times 48]$
batch normalization	48	-
RELU	48	-
dense layer	16	$[48 \times 16]$
output	-	$(8, 2)$

Chapter 5

Concluding Remarks

5.1 Summary and contributions

In chapter 2, we proposed a novel antithetic dithered 1-bit receiver architecture to reduce signal distortion. Efficient channel estimation algorithms were developed to exploit the induced negative correlated noise for improved estimation performance. We illustrated that both linear and nonlinear estimators can benefit from negative correlation. We provided a rigorous analysis of a low complexity nonlinear estimator for channel estimation. In the process, we developed a generalized statistical framework to analyze correlated quantized output arising from this generalized linear model. We formalized the approximation technique used in this work as a special case of the more general pseudo maximum likelihood method. A parameter expanded EM (PX-EM) algorithm applied to such a system was shown to exhibit fast convergence, possessing an upper bounded convergence guarantee and a graceful monotonic estimation performance over a large SNR range. Stochastic Gibbs sampling algorithms were constructed to evaluate truncated multivariate normal distributions and to implement an asymptotically exact data augmentation algorithm for comparison.

In chapter 3, we studied the feasibility of using feed forward networks to learn the nonlinear relationship of the 1-bit MIMO model purely from data. We developed feed forward neural network (FFNN) based soft-detectors and provide an analysis of the performance these detectors, address and partially resolve the issues that prevent the networks from reaching the ideal maximum likelihood (ML) performance limit. Next we turned to the simpler symbol detection problem and assess the performance issues by analyzing the performance of several state of the art FFNN architectures. We demonstrated that the performance limit cannot be overcome by modern deep learning architectures and training techniques, and provided possible explanations to this behavior. We conjectured that the extreme nonlinearity of the likelihood function makes the task of learning difficult, sufficient inductive bias must be given to the neural network to reduce the difficulty of learning, and direct mappings from data to approximate posterior estimates might be inferior to architectures based on iterative methods.

In chapter 4, we evaluated existing model based approaches, identified the shortcomings of gradient methods that rely on the exact likelihood and determined the attributes that made the sigmoid approximation robust in the high SNR regime. We focused on a set of algorithms based on variational methods and proposed a structured deep learning detector based on stochastic variational inference. Stochastic estimate of the mean field update was introduced to reduce complexity of the algorithm. Damping was added to further improve the performance of mean field inference (MFI). Parallel processing was proposed to reduce inference time. The proposed PSMFNet contains few parameters and can be trained efficiently using standard deep learning techniques. In numerical experiments, the proposed detector was shown to outperform existing methods that do not employ a second candidate search step.

5.2 Future work

In Chapter 3, we have seen empirically that there is a limit to what a feed-forward rectifier network can learn in a supervised setting. Thus a naive approach of employing dense networks will not produce an optimal result in general.

Two aspects of the problem require refinement. The first and foremost is that we have a known generative model, and none of this information is given to the network. This information is contained in the likelihood, and fortunately the likelihood is smooth thus gradient of the likelihood can be computed to indicate the direction of further improvement. The approximated likelihood can be used to compute the gradient.

As an attempt to tackle this challenge, we may take a page from recent literature [112, 150], where the optimizing algorithm or inference procedure is learned via deep learning. When implemented using a recurrent architecture, the proposed approaches are in essence trying to identify an instance within the class of iterative algorithms that is most suited to the problem at hand. Precisely, our goal is find the maximizer of the likelihood:

$$\theta^* = \arg\max_{\theta \in \Theta} f(\theta) \tag{5.1}$$

Instead of performing simple gradient ascent to evolve the parameters, we strive to learn an optimizer g that produces the following sequence of parameter updates

$$\theta_{t+1} = \theta_t + g_t(\nabla f(\theta_t), \phi). \tag{5.2}$$

The update rule g can be modeled with recurrence such that it can make decisions based on current and past gradients.

Another dimension that deserves attention is the role that symmetry or more generally endowing specific structure of the network have in reducing the complexity of the

problem we are learning. In [151], the authors reveal that the highly successful convolutional network can be viewed as a hand crafted prior. The main benefit is that gradient updates of the parameters quickly arrived at the solution for natural images, whereas it resisted bad solutions. [53] noted that the interaction between users can be framed as self-attention where a user's signal can be explained by interference from other users and the observed received signal. From this insight, the self-attention mechanism can be employed to produce higher quality state vectors.

In Chapter 4, we restricted our attention to variational methods as the basis for the structured deep learning algorithm. A promising direction is the development of inference methods via the use of graph neural networks (GNNs) [152, 153]. Note that the 1-bit wireless communication model can be represented as a factor graph. Belief propagation can be used to develop a message passing algorithm over the factor graph. The main challenge is the requirement to model high order dependencies. GNNs have been used to learn message passing algorithms over simple factor graphs with pairwise potentials [154, 155]. However when higher order potentials are involved, specialized factor nodes [156], tensor decomposition [157] are required. In another case, the use of a belief propagation neural network is confined to niche applications [158]. A GNN that can be broadly applied to factor graphs with high order potentials will have immense potential.

Bibliography

[1] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.

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

[3] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian, "Analog-to-digital converters," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 69–77, 2005.

[4] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on selected areas in communications*, vol. 17, no. 4, pp. 539–550, 1999.

[5] B. Murmann, "ADC performance survey 1997-2021." [Online]. Available: http://web.stanford.edu/ murmann/adcsurvey.html.

[6] T. Sundstrom, B. Murmann, and C. Svensson, "Power dissipation bounds for high-speed nyquist analog-to-digital converters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 3, pp. 509–518, 2008.

[7] J. Singh, S. Ponnuru, and U. Madhow, "Multi-gigabit communication: The ADC bottleneck," in *2009 IEEE International Conference on Ultra-Wideband*. IEEE, 2009, pp. 22–27.

[8] A. Mezghani and J. A. Nossek, "On ultra-wideband MIMO systems with 1-bit quantized outputs: Performance analysis and input optimization," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*. IEEE, 2007, pp. 1286–1289.

[9] C. Risi, D. Persson, and E. G. Larsson, "Massive mimo with 1-bit adc," *arXiv preprint arXiv:1404.7736*, 2014.

[10] J. Mo and R. W. Heath, "Capacity analysis of one-bit quantized mimo systems with transmitter channel state information," *IEEE transactions on signal processing*, vol. 63, no. 20, pp. 5498–5512, 2015.

[11] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "One-bit massive MIMO: Channel estimation and high-order modulations," in *Communication Workshop (ICCW), 2015 IEEE International Conference on.* IEEE, 2015, pp. 1304–1309.

[12] Y. Li, C. Tao, G. Seco-Granados, A. Mezghani, A. L. Swindlehurst, and L. Liu, "Channel estimation and performance analysis of one-bit massive MIMO systems," *IEEE Transactions on Signal Processing*, 2017.

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

[14] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, "The expressive power of neural networks: A view from the width," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6232–6240.

[15] B. Hanin and M. Sellke, "Approximating continuous functions by relu nets of minimal width," 2017.

[16] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[21] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

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

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

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

[27] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[28] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[29] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, 2014, pp. 2933–2941.

[30] L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou, "Empirical analysis of the hessian of over-parametrized neural networks," *arXiv preprint arXiv:1706.04454*, 2017.

[31] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in neural information processing systems*, 2016, pp. 586–594.

[32] A. J. Bray and D. S. Dean, "Statistics of critical points of gaussian fields on large-dimensional spaces," *Physical review letters*, vol. 98, no. 15, p. 150201, 2007.

[33] D. Soudry and Y. Carmon, "No bad local minima: Data independent training error guarantees for multilayer neural networks," *arXiv preprint arXiv:1605.08361*, 2016.

[34] M. Belkin, D. J. Hsu, and P. Mitra, "Overfitting or perfect fitting? risk bounds for classification and regression rules that interpolate," in *Advances in Neural Information Processing Systems*, 2018, pp. 2300–2311.

[35] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[36] M. S. Advani and A. M. Saxe, "High-dimensional dynamics of generalization error in neural networks," *arXiv preprint arXiv:1710.03667*, 2017.

[37] T. Liang and A. Rakhlin, "Just interpolate: Kernel" ridgeless" regression can generalize," *arXiv preprint arXiv:1808.00387*, 2018.

[38] X. Dou and T. Liang, "Training neural networks as learning data-adaptive kernels: Provable representation and approximation benefits," *arXiv preprint arXiv:1901.07114*, 2019.

[39] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

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

[41] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[42] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *arXiv preprint arXiv:1707.07980*, 2017.

[43] S. Dörner, S. Cammerer, J. Hoydis, and S. Ten Brink, "Deep learning based communication over the air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2017.

[44] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, "OFDM-autoencoder for end-to-end learning of communications systems," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2018, pp. 1–5.

[45] E. Balevi and J. G. Andrews, "One-bit OFDM receivers via deep learning," *IEEE Transactions on Communications*, 2019.

[46] E. Balevi, A. Doshi, and J. G. Andrews, "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2079–2090, 2020.

[47] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive mimo system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.

[48] O. Shental and J. Hoydis, "" machine LLRning": Learning to softly demodulate," *arXiv preprint arXiv:1907.01512*, 2019.

[49] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[50] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2017, pp. 1–5.

[51] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 584–588.

[52] 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, 2020.

[53] K. Pratik, B. D. Rao, and M. Welling, "RE-MIMO: Recurrent and permutation equivariant neural MIMO detection," *arXiv preprint arXiv:2007.00140*, 2020.

[54] L. V. Nguyen, D. H. Nguyen, and A. L. Swindlehurst, "DNN-based detectors for massive MIMO systems with low-resolution ADCs," IEEE, pp. 1–6, 2021.

[55] J. Singh, O. Dabeer, and U. Madhow, "On the limits of communication with low-precision analog-to-digital conversion at the receiver," *IEEE Transactions on Communications*, vol. 57, no. 12, 2009.

[56] J. Choi, J. Mo, and R. W. Heath, "Near maximum-likelihood detector and channel estimator for uplink multiuser massive MIMO systems with one-bit ADCs," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 2005–2018, 2016.

[57] J. García, J. Munir, K. Roth, and J. A. Nossek, "Channel estimation and data equalization in frequency-selective MIMO systems with one-bit quantization," *arXiv preprint arXiv:1609.04536*, 2016.

[58] S. P. Lipshitz, R. A. Wannamaker, and J. Vanderkooy, "Quantization and dither: A theoretical survey," *J. Audio Eng. Soc*, vol. 40, no. 5, pp. 355–375, 1992. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=7047

[59] S. R. Norsworthy, R. Schreier, and G. C. Temes, *Delta-sigma data converters: theory, design, and simulation*. Wiley-IEEE Press, 1996.

[60] B. Widrow, I. Kollar, and M.-C. Liu, "Statistical theory of quantization," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 353–361, Apr 1996.

[61] A. Pollok, Y. Chen, D. Haley, and L. M. Davis, "Quantization noise mitigation via parallel ADCs," *IEEE Signal Processing Letters*, vol. 21, no. 12, pp. 1491–1495, Dec 2014.

[62] J. G. McMichael, S. Maymon, and A. V. Oppenheim, "Exploiting cross-channel quantizer error correlation in time-interleaved analog-to-digital converters," in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov 2011, pp. 525–529.

[63] P. Diaconis and S. Holmes, "Stein's method: expository lectures and applications," in *Institute of Mathematical Statistics, Lecture Notes-Monograph Series (Book 46)*. IMS, 2004.

[64] C. Mollén, J. Choi, E. G. Larsson, and R. W. Heath, "One-bit ADCs in wideband massive MIMO systems with OFDM transmission," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 3386–3390.

[65] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Transactions on Communication Technology*, vol. 12, no. 4, pp. 162–165, 1964.

[66] I.-G. Lee, J. Son, E. Choi, and S.-K. Lee, "Fast automatic gain control employing two compensation loop for high throughput MIMO-OFDM receivers," in *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*. IEEE, 2006, pp. 4–pp.

[67] M. Egozcue, L. F. Garcia, and W.-K. Wong, "On some covariance inequalities for monotonic and non-monotonic functions," *Journal of Inequalities in Pure and Applied Mathematics*, vol. 10, no. 3, pp. 1–7, 2009.

[68] H. White, "Maximum likelihood estimation of misspecified models," *Econometrica: Journal of the Econometric Society*, pp. 1–25, 1982.

[69] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*, E. Parzen, K. Tanabe, and G. Kitagawa, Eds. New York, NY: Springer New York, 1998, pp. 199–213.

[70] C. Gourieroux, A. Monfort, and A. Trognon, "Pseudo maximum likelihood methods: Theory," *Econometrica: Journal of the Econometric Society*, pp. 681–700, 1984.

[71] D. J. Olive, "Applied robust statistics," *Preprint M-02-006*, 2008.

[72] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[73] C. Liu and D. B. Rubin, "The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence," *Biometrika*, vol. 81, no. 4, pp. 633–648, 1994.

[74] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[75] C. J. Wu, "On the convergence properties of the EM algorithm," *The Annals of statistics*, pp. 95–103, 1983.

[76] C. Liu, D. B. Rubin, and Y. N. Wu, "Parameter expansion to accelerate EM: The PX-EM algorithm," *Biometrika*, vol. 85, no. 4, pp. 755–770, 1998.

[77] J. S. Liu and Y. N. Wu, "Parameter expansion for data augmentation," *Journal of the American Statistical Association*, vol. 94, no. 448, pp. 1264–1274, 1999.

[78] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation," *Journal of the American statistical Association*, vol. 82, no. 398, pp. 528–540, 1987.

[79] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[80] P. Damien and S. G. Walker, "Sampling truncated normal, beta, and gamma densities," *Journal of Computational and Graphical Statistics*, vol. 10, no. 2, pp. 206–215, 2001.

[81] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for joint MIMO channel estimation and signal detection," *arXiv preprint arXiv:1907.09439*, 2019.

[82] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *arXiv preprint arXiv:1906.04610*, 2019.

[83] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," *arXiv preprint arXiv:2002.02750*, 2020.

[84] Y.-S. Jeon, N. Lee, S.-N. Hong, and R. W. Heath, "One-bit sphere decoding for uplink massive MIMO systems with one-bit adcs," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4509–4521, 2018.

[85] D. Barber and F. Agokov, "The IM algorithm: a variational approach to information maximization," *Advances in Neural Information Processing Systems*, vol. 16, p. 201, 2004.

[86] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 297–304.

[87] E. Zehavi, "8-PSK trellis codes for a rayleigh channel," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, 1992.

[88] G. Böcherer, F. Steiner, and P. Schulte, "Bandwidth efficient and rate-matched low-density parity-check coded modulation," *IEEE Transactions on Communications*, vol. 63, no. 12, pp. 4651–4665, 2015.

[89] A. Martinez, A. G. i Fabregas, G. Caire, and F. M. Willems, "Bit-interleaved coded modulation revisited: A mismatched decoding perspective," *IEEE Transactions on Information Theory*, vol. 55, no. 6, pp. 2756–2765, 2009.

[90] G. Böcherer, "Achievable rates for shaped bit-metric decoding," *arXiv preprint arXiv:1410.8075*, 2014.

[91] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, 1982.

[92] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE transactions on information theory*, vol. 44, no. 3, pp. 927–946, 1998.

[93] X. Li and J. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *Electronics Letters*, vol. 34, no. 10, pp. 942–943, 1998.

[94] S. Ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEU Int. J. Electron. Commun*, vol. 54, no. 6, pp. 389–398, 2000.

[95] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. ten Brink, "Trainable communication systems: Concepts and prototype," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5489–5503, 2020.

[96] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[97] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[98] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1501–1510.

[99] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[100] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[101] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Advances in Neural Information Processing Systems*, 2017, pp. 6594–6604.

[102] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[103] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6924–6932.

[104] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[105] B. C. Ross, "Mutual information between discrete and continuous data sets," *PloS one*, vol. 9, no. 2, 2014.

[106] G. V. Moustakides and K. Basioti, "Training neural networks for likelihood/density ratio estimation," *arXiv preprint arXiv:1911.00405*, 2019.

[107] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[108] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. De Freitas, "Predicting parameters in deep learning," in *Advances in neural information processing systems*, 2013, pp. 2148–2156.

[109] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi, "Learning feed-forward one-shot learners," in *Advances in neural information processing systems*, 2016, pp. 523–531.

[110] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[111] J. Marino, Y. Yue, and S. Mandt, "Iterative amortized inference," *arXiv preprint arXiv:1807.09356*, 2018.

[112] P. Putzky and M. Welling, "Recurrent inference machines for solving inverse problems," *arXiv preprint arXiv:1706.04008*, 2017.

[113] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser MIMO systems," *IEEE Transactions on Communications*, vol. 61, no. 4, pp. 1436–1449, 2013.

[114] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[115] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014.

[116] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[117] L. V. Nguyen and D. H. Nguyen, "Linear receivers for massive MIMO systems with one-bit ADCs," *arXiv preprint arXiv:1907.06664*, 2019.

[118] S. Wang, Y. Li, and J. Wang, "Multiuser detection in massive MIMO with quantized phase-only measurements," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 4576–4581.

[119] ——, "Multiuser detection for uplink large-scale MIMO under one-bit quantization," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 4460–4465.

[120] C.-K. Wen, C.-J. Wang, S. Jin, K.-K. Wong, and P. Ting, "Bayes-optimal joint channel-and-data estimation for massive MIMO with low-precision ADCs," *IEEE Transactions on Signal Processing*, vol. 64, no. 10, pp. 2541–2556, 2015.

[121] Z. Zhang, X. Cai, C. Li, C. Zhong, and H. Dai, "One-bit quantized massive MIMO detection based on variational approximate message passing," *IEEE Transactions on Signal Processing*, vol. 66, no. 9, pp. 2358–2373, 2017.

[122] L. V. Nguyen, A. L. Swindlehurst, and D. H. Nguyen, "SVM-based channel estimation and data detection for one-bit massive MIMO systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2086–2099, 2021.

[123] L. V. Nguyen, D. T. Ngo, N. H. Tran, A. L. Swindlehurst, and D. H. Nguyen, "Supervised and semi-supervised learning for MIMO blind detection with low-resolution ADCs," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2427–2442, 2020.

[124] Y.-S. Jeon, S.-N. Hong, and N. Lee, "Supervised-learning-aided communication framework for MIMO systems with low-resolution ADCs," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7299–7313, 2018.

[125] ——, "Blind detection for MIMO systems with low-resolution ADCs using supervised learning," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[126] K. Gao, N. Estes, B. Hochwald, J. Chisum, and J. N. Laneman, "Power-performance analysis of a simple one-bit transceiver," in *2017 Information Theory and Applications Workshop (ITA)*. IEEE, 2017, pp. 1–10.

[127] Y. Li and R. Zemel, "Mean-field networks," *arXiv preprint arXiv:1410.5884*, 2014.

[128] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference." *Journal of Machine Learning Research*, vol. 14, no. 5, 2013.

[129] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[130] S. R. Bowling, M. T. Khasawneh, S. Kaewkuekool, and B. R. Cho, "A logistic approximation to the cumulative normal distribution," *Journal of Industrial Engineering and Management*, vol. 2, no. 1, pp. 114–127, 2009.

[131] D. K. W. Ho and B. D. Rao, "Antithetic dithered 1-bit massive MIMO architecture: Efficient channel estimation via parameter expansion and PML," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2291–2303, 2019.

[132] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, no. 1, pp. 25–37, 2000.

[133] P. Baqué, T. Bagautdinov, F. Fleuret, and P. Fua, "Principled parallel mean-field inference for discrete random fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5848–5857.

[134] P. Krähenbühl and V. Koltun, "Parameter learning and convergent inference for dense random fields," in *International Conference on Machine Learning*. PMLR, 2013, pp. 513–521.

[135] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari, "Composite objective mirror descent." in *COLT*. Citeseer, 2010, pp. 14–26.

[136] M. J. Wainwright and M. I. Jordan, *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[137] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 267–282, 2007.

[138] N. D. Campbell, K. Subr, and J. Kautz, "Fully-connected CRFs with non-parametric pairwise potential," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1658–1665.

[139] R. Frostig, S. I. Wang, P. Liang, and C. D. Manning, "Simple MAP inference via low-rank relaxations." in *NIPS*, 2014, pp. 3077–3085.

[140] V. Vineet, J. Warrell, and P. H. Torr, "Filter-based mean-field inference for random fields with higher-order terms and product label-spaces," *International Journal of Computer Vision*, vol. 110, no. 3, pp. 290–307, 2014.

[141] M. Martinez and R. Stiefelhagen, "Taming the cross entropy loss," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 628–637.

[142] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[143] K. Kawaguchi and L. Kaelbling, "Elimination of all bad local minima in deep learning," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 853–863.

[144] S. Liang, R. Sun, J. D. Lee, and R. Srikant, "Adding one neuron can eliminate all bad local minima," *Advances in Neural Information Processing Systems*, vol. 31, pp. 4350–4360, 2018.

[145] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[146] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[147] G. Tucker, A. Mnih, C. J. Maddison, D. Lawson, and J. Sohl-Dickstein, "Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models," *arXiv preprint arXiv:1703.07370*, 2017.

[148] W. Grathwohl, D. Choi, Y. Wu, G. Roeder, and D. Duvenaud, "Backpropagation through the void: Optimizing control variates for black-box gradient estimation," *arXiv preprint arXiv:1711.00123*, 2017.

[149] W. C. Horrace, "Some results on the multivariate truncated normal distribution," *Journal of multivariate analysis*, vol. 94, no. 1, pp. 209–221, 2005.

[150] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.

[151] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[152] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[153] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.

[154] A. Scotti, N. N. Moghadam, D. Liu, K. Gafvert, and J. Huang, "Graph neural networks for massive mimo detection," *arXiv preprint arXiv:2007.05703*, 2020.

[155] K. Yoon, R. Liao, Y. Xiong, L. Zhang, E. Fetaya, R. Urtasun, R. Zemel, and X. Pitkow, "Inference in probabilistic graphical models by graph neural networks," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 868–875.

[156] V. G. Satorras and M. Welling, "Neural enhanced belief propagation on factor graphs," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 685–693.

[157] Z. Zhang, F. Wu, and W. S. Lee, "Factor graph neural network," *arXiv preprint arXiv:1906.00554*, 2019.

[158] J. Kuck, S. Chakraborty, H. Tang, R. Luo, J. Song, A. Sabharwal, and S. Ermon, "Belief propagation neural networks," *arXiv preprint arXiv:2007.00295*, 2020.