UNIVERSITY OF CALIFORNIA SAN DIEGO

Towards Model-based Synergistic Learning for Robust Next-Generation MIMO Systems

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

in

Electrical Engineering
(Machine Learning and Data Science)

by

Aditya Sant

Committee in charge:

Professor Bhaskar D. Rao, Chair
Professor Jiawang Nie
Professor Piya Pal
Professor Rose Yu
Professor Xinyu Zhang

2024

The dissertation of Aditya Sant is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Undertaking, and completing, this Ph.D. journey has been a transformative process in me, instilling in various life lessons and learnings. I strongly believe that through all the interactions, projects, and grinding through new research territory, this has taught me the value of approaching any problem with the right mindset. By no means would it have been possible to complete this journey alone. The different mentors, guides, friends and family that have helped me throughout this journey are acknowledged here.

My journey would not have even begun had it not been for the opportunity to work in the Photonic Systems Lab at UCSD. I owe a ton of gratitude to Eduardo, Dr. Nikola Alic and Prof. Stojan Radic, who took a chance on me and decided to accept me as a researcher into the lab in the first year of my Ph.D.

After working in the Photonic Systems Lab for the first year, I decided to move into the area of wireless communication research. I approached Prof. Bhaskar Rao in the Summer of 2018, who took me in as a Ph.D. student in the DSP Lab. His mentorship carried me forward for the next six years to this Ph.D. completion. Prof. Rao, with over 40 years of experiences as an esteemed faculty at UCSD, has the remarkable ability to bring out the best in every one of his students. He has shown endless patience, from helping me get adjusted into a new research area, to letting me me take my time in understanding my topic as a researcher in the lab. Every research discussion with him, from the most structured meetings to the most open-ended brainstorming sessions has always given me extensive insights into my own work. Not only has Prof. Rao imparted immense technical wisdom to me as a researcher over my Ph.D. journey, he has instilled the ability to effectively present and express my work to the outside world. He is always extremely approachable and open to hearing new ways of conducting research, always encouraging me to follow my intuition in research, and helping me back up the work with strong mathematical rigor. The guidance of Prof. Rao has made this Ph.D. possible, even through the darkest of times, and no words can even begin to describe his contribution in helping me reach its fruition.

Having attended one of the top engineering departments in the world, I was extremely fortunate to learn under the some of the best professors in the field. I have enjoyed all my classes in the areas ranging from Photonics, to signal processing, and machine learning, under Profs. George Papen, Stojan Radic, Nuno Vasconcelos, Piya Pal, Paul Siegel, Gabriel Rebeiz, Alon Orlitsky, Charles Deledalle, Jose Unpingco, and Bhaskar Rao. Additionally all the seminars and informal discussions with various professors in the department have always ended up teaching me something new about the field.

I have learnt a lot about research methodology, writing codes and preparing technical documents through my Finnish collaborator and friend Markus Leinonen. He joint Prof. Rao's Lab in 2020 as a visiting

researcher for six months from the University of Oulu in Finland. We had a fruitful research collaboration over the next two years, which led to the publication of my very first journal paper. He unfortunately passed away in February 2023, but is fondly remembered as a friend, mentor, and collaborator, in that order.

The environment in the DSP Lab has been immensely enjoyable owing to both my research advisor and my labmates. Rohan and Govind, who parallely shared their research journey and milestones with me, were always there to help me with any problems I would face. I also learnt a lot and shared many pleasant memories with my other labmates Kuan-Lin and Hitesh. The work and contributions of the senior graduated students laid the foundation to build my own research. Most of the research projects in my Ph.D. were inspired by, and took forward the separate research works of David Ho and Zhillin Zhang. I would like to acknowledge the role of the all the graduated researchers from Prof. Rao's group, including Yacong, Yonghee, Jing, Elina, and David Wipf, who have directly, or indirectly (through their work), helped and guided me through my own research journey.

I would also like to acknowledge the ECE Office and all the other ancillary offices and staff at UCSD, including the International Students and Programs Office (ISPO), Division of Graduate Education and Postdoctoral Affairs (GEPA), who have streamlined the formalities and paperwork over different stages from beginning to end. The different staff members have always shown a lot of patience in answering any and all questions I would have related to the workflow and operations.

I am fortunate to have been given an opportunity to work with R&D teams in industry through two separate internship projects - at Nokia Bell Labs and Qualcomm Technologies. Both these experiences helped to widen my research perspective beyond academia and provided experience of working with larger teams and projects. I thank my mentor Dr. Harish Viswanathan at Nokia Bell Labs for guiding me through my internship project. I also am thankful for my friends Umair, Bibek and Sachin for so many memorable moments during my time in Murray Hill. My second internship at Qualcomm, although remote, was also a very key learning opportunity in the field of machine learning. I am grateful to my mentors Dr. Joseph Soriaga and Dr. Afshin Abdi for taking me into the team and guiding me through the entire internship, including following up on the research publication after the culmination of this work.

My friends in San Diego have been responsible for the most enjoyable times I have experienced in this city. I was fortunate to a explore a large chunk of the various hidden gems in and around the city with my friend Tharun, who introduced me to the city's contrasting cultures of hiking and surfing. A rich social dimension was provided by all my friends in San Diego, including Rohan, Sukanya, Govind, Anwesan, Tharun, Sheel, Aashi, Mustafa, Malhar, Pranav, Nadim, Ayush, Pulak, Mehmet, Sina, Uday, Jacob, Tim and Raghavendra. I always fondly remember the innumerable hours over coffee, dinners, ultimate frisbee,

festivals, birthdays, trips, hikes and so much more!

My friends from all over, including my undergradute college, and hometown of Pune are lifelong connections. I want to thank Bhavik, Samir, Judo, Kaustubh, Dane, Ankit, Shiney, Deokule, Deshmukh, Dave, Aroon, Adit, Chavan, Akash, Mahor, Utkarsh, Kevin, Abhishek, Sanjana, who continue to keep enriching my life.

My heart belongs to my mom, dad and younger brother, Varun. All of them have been my foundation and strength, helping me, guiding me, shaping me, and making me. Indeed I am the person they have made me today and to say they support me in everything I do would be a gross understatement.

All of my family back home in India, both old and new, have always supported and encouraged me in everything I decided to pursue. Even within the US, since I first moved to the Bay Area during my last year of my Ph.D., I have never felt away from home due to my extended family here.

Finally, I cannot thank one person enough for being my rock in this Ph.D. journey - my wife Avni. Her continuous support in helping me through all the ups and downs of my Ph.D. life has made it possible to navigate this journey without worrying about the outcome.

Chapter 2, in part, is a reprint, with permission, of the material as it appears in the papers: Aditya Sant, Markus Leinonen, and Bhaskar D. Rao. "Block-sparse Signal Recovery via General Total Variation Regularized Sparse Bayesian Learning." IEEE Transactions on Signal Processing 70 (2022): 1056-1071 and Aditya Sant, Markus Leinonen, and Bhaskar D. Rao. "General Total Variation Regularized Sparse Bayesian Learning for Robust Block-sparse Signal Recovery." ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021. The dissertation author was the primary investigator and author of these papers.. These works were supported in part by ONR Grant No. N00014-18-1-2038, NSF grant CCF-2124929 and the UCSD Center for Wireless Communications. The work of M. Leinonen has been financially supported in part by Walter Ahlström Foundation through Tutkijat Maailmalle program, Infotech Oulu, the Academy of Finland (grant 323698 and 319485), and Academy of Finland 6Genesis Flagship (grant 318927).

Chapter 3, in part, is a reprint, with permission, of the material as it appears in the papers: Aditya Sant, and Bhaskar D. Rao. "Insights into Maximum Likelihood Detection for One-bit Massive MIMO Communications", IEEE Transactions on Wireless Communications, (under review). The dissertation author was the primary investigator and author of these papers. This work was supported in part by ONR Grant No. N00014-18-1-2038, NSF grant CCF-2124929, NSF Grant CCF-2225617, and the UCSD Center for Wireless Communications.

Chapter 4, in part, is a reprint, with permission, of the material as it appears in the papers: Aditya

Sant, and Bhaskar D. Rao. "Regularized Neural Detection for One-Bit Massive MIMO Communication Systems." arXiv e-prints (2023): arXiv-2305 and Aditya Sant, and Bhaskar D. Rao. "Regularized Neural Detection for Millimeter Wave Massive MIMO Communication Systems with One-Bit ADCs." ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023. The dissertation author was the primary investigator and author of these papers.. These works were supported in part by ONR Grant No. N00014-18-1-2038, NSF grant CCF-2124929, NSF Grant CCF-2225617, and the UCSD Center for Wireless Communications.

Chapter 5, in part, is a reprint, with permission, of the material as it appears in the papers: Aditya Sant, and Bhaskar D. Rao. "Few-bit MIMO Detection Using DNN-aided Dequantization", which is to be submitted. The dissertation author was the primary investigator and author of these papers. This work was supported in part by ONR Grant No. N00014-18-1-2038, NSF grant CCF-2124929, NSF Grant CCF-2225617, and the UCSD Center for Wireless Communications.

VITA

2017	B. Tech & M. Tech in Electrical Engineering at Indian Institute of Technology Madras (minor in Physics), Chennai, India
2023	M. S. in Electrical Engineering (Machine Learning and Data Science), University of California San Diego, La Jolla, United States
2018-2024	Graduate Research Assistant, University of California San Diego, La Jolla, United States
2024	Ph. D. in Electrical Engineering, University of California San Diego, La Jolla, United States

PUBLICATIONS AND PATENTS

Aditya Sant, and Bhaskar D. Rao. "DOA Estimation in Systems with Nonlinearities for MmWave Communications." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

Aditya Sant, Markus Leinonen, and Bhaskar D. Rao. "General Total Variation Regularized Sparse Bayesian Learning for Robust Block-sparse Signal Recovery." ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021.

Aditya Sant, Markus Leinonen, and Bhaskar D. Rao. "Block-sparse Signal Recovery via General Total Variation Regularized Sparse Bayesian Learning." IEEE Transactions on Signal Processing 70 (2022): 1056-1071.

Aditya Sant, and Bhaskar D. Rao. "Regularized Neural Detection for One-Bit Massive MIMO Communication Systems." arXiv e-prints (2023): arXiv-2305.

Aditya Sant, and Bhaskar D. Rao. "Regularized Neural Detection for Millimeter Wave Massive MIMO Communication Systems with One-Bit ADCs." ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023.

Aditya Sant, and Bhaskar D. Rao. "Insights into Maximum Likelihood Detection for One-bit Massive MIMO Communications", IEEE Transactions on Wireless Communications, (under review).

Aditya Sant, and Bhaskar D. Rao. "Few-bit MIMO Detection Using DNN-aided Dequantization", (under prep)

Aditya Sant, Afshin Abdi, and Joseph Soriaga. "Deep Sequential Beamformer Learning for Multipath Channels in Mmwave Communication Systems." ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022.

Aditya Uday Sant, Afshin Abdi, and Joseph Binamira Soriaga. "Beamforming Multipath Wireless Channels using Neural Networks." U.S. Patent Application No. 17/938,313.

ABSTRACT OF THE DISSERTATION

Towards Model-based Synergistic Learning for Robust Next-Generation MIMO Systems

by

Aditya Sant

Doctor of Philosophy in Electrical Engineering

(Machine Learning and Data Science)

University of California San Diego, 2024

Professor Bhaskar D. Rao, Chair

As the demand for high-speed, reliable wireless communication among interconnected devices rises, the need for robust next-generation wireless MIMO systems becomes crucial. This dissertation is motivated by the need to address the challenges inherent in the development of such systems, specifically focusing on two key aspects: *(i)* Robust block-sparse mmWave channel modeling and *(ii)* Robust detection in the presence of few-bit MIMO systems. Central to this thesis is the integration of model-based methods with deep neural network (DNN)-aided approaches, leveraging the synergy between these two paradigms to enhance system performance and mitigate the impact of model inaccuracies and mismatches.

The first part of this dissertation focuses on the spatial modeling of mmWave channels, to capture the heterogeneous scattering behavior, through block-sparse signal recovery. Despite the promise of block-

sparse signal processing for channel modeling in the angular domain, a key challenge is block-patterned estimation without knowledge of block sizes and boundaries. This work propose a novel total variation sparse Bayesian learning (TV-SBL) method for block-sparse signal recovery under unknown block patterns. Unlike conventional approaches that employ block-promoting regularization on signal components, this method introduces two classes of *hyperparameter* regularizers for the SBL cost function inspired by total variation (TV) denoising. The first class relies on a conventional TV difference unit, allowing iterative SBL inference through convex optimization, thus facilitating the use of various numerical solvers. The second class integrates a region-aware TV penalty to penalize signal and zero blocks differently, thereby enhancing performance. An alternating optimization algorithm based on expectation-maximization is derived for computationally efficient parallel updates for both regularizer classes. Going beyond model-based methods, this work also presents a basis for extension to DNN-aided block-sparse signal recovery for 1-D and 2-D signals.

The second part of this dissertation focuses on designing detection algorithms for signal recovery in few-bit MIMO systems, beginning with a detailed analysis of one-bit MIMO systems. This begins by analyzing the smoothness and convexity of the one-bit likelihood function, based on the Gaussian CDF, for signal recovery. This culminates in an improved gradient descent (GD) algorithm for one-bit MIMO, and ensuing convergence analysis. The accelerated GD method is applied to one-bit MIMO recovery, further improving convergence. The analysis is extended to an effective surrogate function for the Gaussian CDF, i.e., the logistic regression (LR), explaining the enhanced performance when utilized as a surrogate likelihood. Constrained optimization, incorporating detection from a finite M-QAM constellation, is addressed by the introduction of a *learnable* Gaussian denoiser to project detected symbols onto the M-QAM subspace.

Another class of DNN-aided regularizers is proposed for one-bit MIMO, utilizing a regularized gradient descent update. A novel constellation-aware loss function is incorporated to tailor the DNN loss function to M-QAM symbol recovery. The key utility of a generalized DNN-aided GD update is for detection in mmWave channels, where there is a higher contrast in per-user channel powers, presenting a challenge for joint multi-user detection. Leveraging a general parametric DNN structure enables the development of a novel hierarchical detection training algorithm, ensuring network design for equitable detection in mmWave channels, where users with higher channel powers experience improved recovery performance.

The final part of this dissertation extends the research to two-bit MIMO detection. In few-bit MIMO systems like the two-bit MIMO receiver, the quantization noise exhibits distinct characteristics positioned between the fully saturated one-bit scenario and the independently additive noise observed in higher resolutions. However, limited research has focused on accurately characterizing this unique quantization noise profile. These properties of quantization noise, along with the constraints of optimization for MIMO

signal recovery, make DNNs ideally suited for the signal recovery. The DNN-augmented receiver algorithm developed attempts to learn this noise behavior to dequantize the signal, without the need for explicit analytical characterization, thereby enhancing signal recovery in few-bit MIMO systems.

Chapter 1

Introduction

Figure 1.1: Illustrating the different aspects of the next-generation of wireless systems.
Image source: https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/web-why-and-what-you-need-to-know-about-6G-in-2022.pdf

1.1 Next-generation wireless communication systems

In today's interconnected world, wireless systems have become an integral part of our daily lives, facilitating seamless connectivity for a myriad of applications like communication and data transfer, interconnected devices, augmented reality, autonomous vehicles, smart homes and factories, GPS and navigation, edge computing, and so many more [2–4]. The different verticals and applications for the next generation,

i.e. 6G, of wireless communications are illustrated in Fig. 1.1. From the early days of basic voice calls over wired links to the present era of high-speed data transfer and internet-of-things (IoT) connectivity, wireless communication technologies have undergone remarkable evolution. Today, the wireless channel, from WiFi routers to satellite communication, is present at some point or the other when we connect to the internet. The next generation of wireless communication promises to redefine the boundaries of connectivity and enables transformative applications across various domains. Building upon the foundations laid by its predecessors, 6G aims to deliver unprecedented performance in terms of data rates, latency, reliability, and energy efficiency [5,6]. This next generation of wireless communication will rely on two key aspects: *(i)* The mmWave spectrum, and *(ii)* Massive MIMO systems.

One of the key technologies that promises to revolutionize connectivity is the millimeter wave (mmWave) communication system. Operating in frequency bands above 24 GHz up to 100 GHz [7–10], mmWave communication offers significantly larger bandwidths compared to traditional microwave frequencies, paving the way for ultra-high data rates and low-latency transmission. The short wavelength of mmWave signals allows for the integration of large antenna arrays in compact form factors, facilitating beamforming and spatial multiplexing techniques to overcome path loss and propagation limitations inherent to higher frequencies. This is further elaborated next.

The massive Multiple-Input-Multiple-Output (massive MIMO) system design continues to play a pivotal role in shaping the future of wireless networks. By harnessing the power of advanced antenna arrays and beamforming techniques, massive MIMO offers the potential to significantly enhance spectral efficiency, coverage, and capacity, thereby enabling the deployment of ultra-dense networks and supporting a diverse range of applications [11–13]. Massive MIMO systems have several advantages over conventional wireless architectures. Firstly, massive MIMO significantly increases the spatial degrees of freedom, allowing for more efficient utilization of the available spectrum and improved spectral efficiency. Secondly, by exploiting the spatial diversity offered by a large number of antennas, massive MIMO enhances the resilience of wireless links to fading and interference, thereby improving the overall reliability and quality of service. Moreover, massive MIMO facilitates advanced beamforming techniques, enabling precise control over signal transmission and reception directions, which in turn enhances coverage, reduces interference, and enables seamless mobility support. As we embark on the journey towards 6G, the exploration and optimization of massive MIMO systems will be instrumental in realizing the full potential of next-generation wireless communication networks.

1.2 Towards robust MIMO Systems

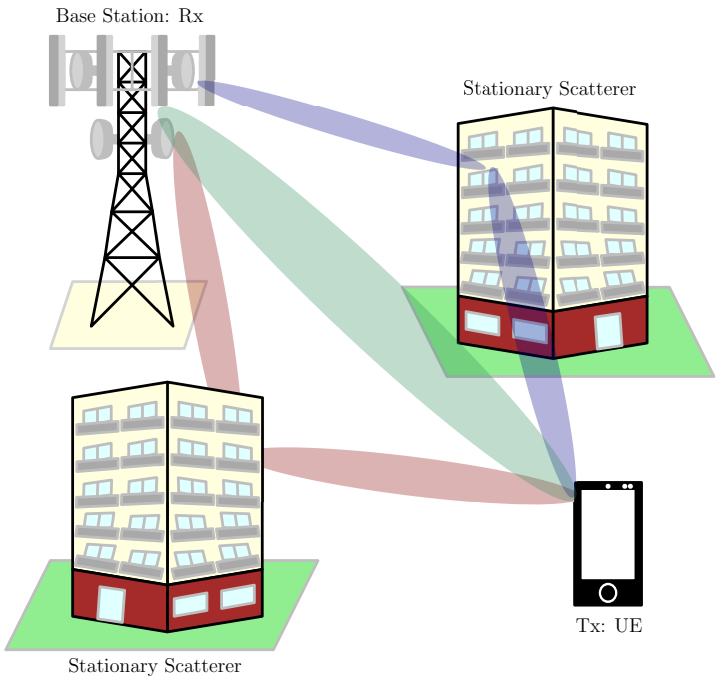


Figure 1.2: Clustered angular scattering for the uplink mmWave channel

Widescale deployment and adoption of the technologies for wireless communication systems is determined by the robustness of system modeling and receiver design. Accurate system modeling enables engineers to understand the behavior of wireless channels under diverse conditions, facilitating the design of robust communication systems capable of mitigating the challenges posed by fading, interference, and other impairments. A comprehensive examination of these aspects entails a detailed exploration of various components within wireless systems. This thesis endeavors to advance the field by introducing enhanced modeling techniques and receiver algorithms tailored for two specific verticals. In particular, we analyze *(i)* Robust block-sparse channel modeling for mmWave channels, and *(ii)* Receiver design for wireless systems with few-bit ADCs. The subsequent sections delve deeper into these topics.

1.2.1 Improved block-sparse signal modeling of mmWave channels

The paradigm of mmWave communication incorporates frequency bands of 24 GHz to 100 GHz [7–10]. Unlike the prevalent sub-6 GHz wireless channels characterized by rich scattering environments, the mmWave spectrum introduces a distinctive characteristic — its channel manifests a more specular behavior, marked by a notable reduction in the abundance of multipath components. The spatial domain analysis of the

mmWave channel reveals a pronounced block-sparse structure[14–16]. The scattering behavior is illustrated in Fig. 1.2, with different clustered multipath components. As seen from this figure, unlike traditional channels where reflections and scattering lead to a dense multipath profile, mmWave channels often exhibit a sparsity that is organized into distinct blocks of various sizes, originating from the heterogeneous scattering and limited propagation paths inherent to these higher-frequency bands [17–19]. Robust signal modeling capable of capturing these block-sparse components is essential for effective channel characterization and signal processing. Moreover, the ability of such modeling techniques to remain resilient to the diverse block sizes and structures is vital for their practical utility in mmWave systems. This will play a pivotal role in building reliable channel estimation algorithms for initial wireless access. To address this challenge, this work delves deeper into a specific class of signal recovery algorithms tailored for scenarios with unknown block structures, leveraging the principles of sparse Bayesian learning. This framework for block-sparse signal recovery is detailed in Chapter 2

1.2.2 MIMO detection for receivers using few-bit ADCs

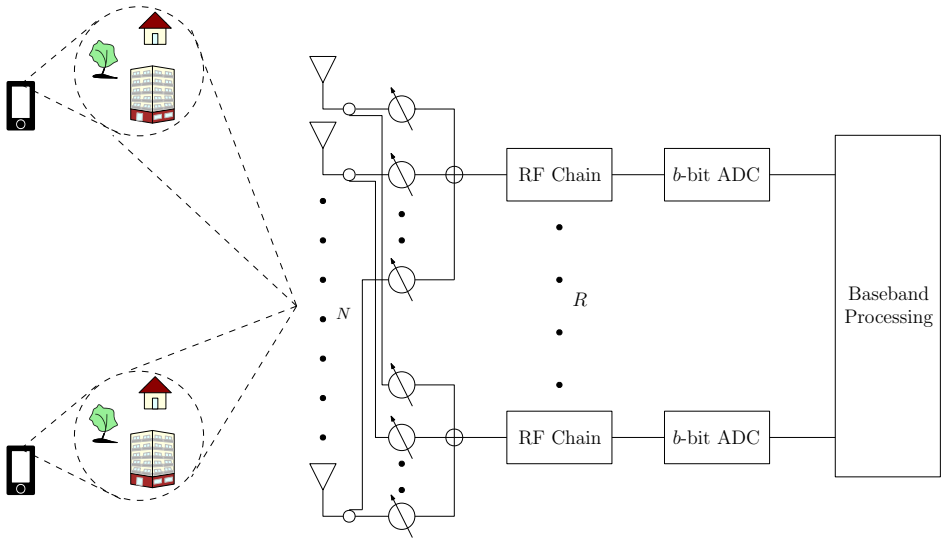


Figure 1.3: General MIMO receiver with b-bit ADC in the receiver chain

The practical deployment of wireless communication systems entails integrating various hardware components, such as power amplifiers, filters, local oscillators, and analog-to-digital converters (ADCs), each introducing its own set of nonlinearities that distort the received signal. This is illustrated via the general MIMO receiver chain blocks in Fig. 1.3. The work in this thesis focuses on examining the hardware-induced nonlinearities stemming from ADCs, particularly few-bit ADCs, which have garnered increasing research

attention [20–25]. As next-generation massive MIMO systems move towards larger number of antennas, the use of few-bit ADCs assist with widescale deployment due to reduced cost and complexity of operating the RF chain. However, few-bit ADCs bring in several hardware-induced nonlinearities, which need to be compensated at the receiver. To this end, efficient modeling of these systems, and the subsequent design of resilient data recovery algorithms at the receiver will play a key role in the design of robust MIMO systems.
One-bit ADCs: In recent years, one-bit ADCs have emerged as a particularly researched case within the realm of few-bit ADCs [1, 20, 26–36]. One-bit ADCs offer a radically simplified approach to analog-to-digital conversion, primarily relying on a single comparator. Unlike traditional multi-bit ADCs that quantize the analog signal into multiple levels, one-bit ADCs simply compare the input signal to a threshold voltage and output a single bit, thus significantly reducing hardware complexity. To fully unlock the potential of one-bit ADCs in next-generation wireless MIMO systems, it is necessary to conduct comprehensive signal modeling, and develop advanced signal processing algorithms specifically designed to accommodate the characteristics of one-bit quantization. The work in this thesis provides analytical insights into signal recovery for one-bit MIMO systems, as well as robust detector algorithm design for these systems, through Chapters 3-4. The analysis in Chapter 5 presents the detector design for few-bit MIMO receivers, particularly two-bit MIMO receivers.

Utilization of DNN-aided methods is particularly essential for building the next generation of robust wireless systems. However, the design of these systems is enhanced by utilizing the model-aided frameworks for signal generation. This is further elaborated in the next section.

1.3 Synergy of model-based and DNN-aided frameworks

Today, artificial intelligence (AI) has revolutionized our interaction with the information-centric world, and heralded in a new age of signal processing. Through AI technologies, we witness a sweeping revolution across diverse domains including computer vision, robotics, autonomous driving systems, natural language processing, medicine, and beyond [37–47]. Remarkable advancements such as the generative pre-trained transformer [48, 49] epitomize the profound strides made in this field, redefining the dynamics of human-information exchange.
Model-based learning: A pivotal class of learning techniques revolves around leveraging forward models for signal recovery. This framework enriches traditional algorithms by incorporating deep neural networks (DNNs) through a process known as algorithm unrolling, thereby enhancing signal recovery performance [50, 51]. In domains such as medical imaging and robotics, where the underlying forward model is clearly

defined, algorithm unrolling with deep neural networks (DNNs) has demonstrated enhanced performance over conventional model-based approaches [46]. These methods have played a crucial role in advancing AI-aided wireless system design, facilitating significant progress in the field. The subsequent discussion delves deeper into this.

1.3.1 The era of model-based AI for wireless MIMO systems

Artificial intelligence (AI) has led to several advancements in wireless communications, with DNN-aided frameworks improving multiple blocks of MIMO systems. These frameworks have significantly bolstered performance across various fronts, including enhanced beamforming [52–54], more robust detection algorithms [55–59], handling hardware nonlinearities [60], and optimized wireless access and resource management strategies [61–63], to name a few. In wireless communication systems, the forward model governing signal propagation and reception is often well-defined, facilitating the implementation of model-based approaches for system optimization. However, real-world wireless channels often exhibit dynamic and complex behaviors, leading to model mismatch. To address this challenge, synergistic learning using model-based approaches and DNN-aided methods presents a viable alternative. By leveraging DNNs to unroll algorithms and capture intricate patterns in wireless channel data, these methods enhance the adaptability and robustness of wireless communication systems in the face of model uncertainties. This synergy not only alleviates the impacts of model mismatch but also facilitates the extension of model-based approaches to scenarios where efficient modeling and scalability pose challenges. Within the scope of this thesis, we delve into the exploration of synergistic learning within two verticals of robust wireless systems introduced in Sec. 1.2: block-sparse signal modeling for mmWave wireless channels and the design of few-bit MIMO detection algorithms. This is elaborated further in the subsequent discussion.

1.3.2 Motivating synergistic learning for block-sparse mmWave channels

As discussed in Sec. 1.2.1, the non-homogeneous scattering in mmWave channels presents significant challenges in modeling a general block-sparse structure across scattered multipath clusters. Model-based approaches, such as specialized signal priors, offer a means to enforce structure on the recovered signal. However, the use of a general prior capable of capturing any block-sparse pattern and size proves unfeasible with solely model-based methods. Additionally, while tractable signal priors are effective in handling 1-D block-sparse signal structures, they often falter when scaling to 2-D scenarios. To overcome these limitations, the synergy between model-based algorithmic approaches and the integration of DNN-aided techniques

emerges as a promising solution. By leveraging the power of DNNs, these techniques augment model-based methods, enhancing their ability to capture the underlying signal structure with greater efficiency and adaptability.

1.3.3 Motivating synergistic learning for few-bit MIMO detection

In the realm of few-bit MIMO signal detection, the synergistic integration of model-based and DNN-aided methods emerges as a promising approach to address the inherent challenges. In practical communication systems, the quantization of signals to few bits often leads to high saturation levels, rendering the modeling of noise intractable with traditional methods. This poses a challenge for conventional receivers that heavily rely on accurate noise statistics for optimal performance [12, 64]. Furthermore, the transmitted symbols, drawn from finite constellation M-QAM sets, impose additional constraints on signal recovery within this constrained set. When relying solely on model-based approaches, accommodating these challenges becomes increasingly complex. However, by harnessing the complementary strengths of model-based algorithms and DNN-aided techniques, a promising avenue opens for enhanced detection performance.

1.4 Dissertation layout

The work in this dissertation independently looks at two facets of robust wireless system design, introduced in Sec. 1.2. The first part, namely Chapter 2 details the contributions to block-sparse signal recovery. The second part, i.e., Chapters 3 - 5 presents the contributions for detection algorithms for few-bit MIMO systems. The structure of the dissertation is summarized below to navigate through the research contributions.

In Chapter 2, the analysis delves into block-sparse signal processing from a broad perspective. A block-sparse recovery algorithm based on the framework of sparse Bayesian learning (SBL), i.e., total variation SBL (TV-SBL) is introduced. The approach introduces a novel class of regularizers to the SBL optimization framework to encourage inter-element coupling, resulting in signal recovery with block-sparse structure. Following this, the optimization of the regularized SBL loss function using different TV-based regularizers is analyzed, along with the resulting algorithm convergence. The chapter concludes by exploring the integration of Deep Neural Network (DNN)-aided techniques to augment block-sparse recovery, initially targeting 1-D block-sparse signals. Additionally, preliminary endeavors employing DNN-aided methods for the recovery of 2-D block-sparse signals are presented, laying the foundation for potential advancements in this direction.

Beginning with Chapter 3, the dissertation delves into robust detection in MIMO systems employing few-bit ADCs. This chapter primarily focuses on the detection for one-bit MIMO systems. Insights into the optimization algorithm for one-bit MIMO detection are provided, followed by a robust GD algorithm for signal recovery. Notably, the chapter introduces an innovative approach utilizing accelerated gradient descent (AGD) for optimizing the one-bit MIMO problem. Following this an analysis on the utility of surrogate likelihood measures is provided, analytically describing the advantages of this approach. The chapter also underscores the significance of projection to the M-QAM symbols in the detection algorithm, highlighting its role in enhancing detection performance. A novel DNN-aided receiver is presented, namely the A-PrOBNet, which unfolds the accelerated GD algorithm along with learnt projection step, showcasing the potential of integrating deep learning techniques into detection processes.

In Chapter 4, regularized GD, using a DNN-aided projection step, is presented as an alternative optimization framework for the one-bit MIMO detection problem introduced in Chapter 3. The regularized GD algorithm is unfolded using two distinct neural networks: *(i)* ROBNet, an unfolded DNN approach, and *(ii)* OBiRIM, a recurrent neural network approach, dedicated to implementing regularized GD-based detection. Following this, the chapter specifically addresses the challenges posed by employing GD-based joint detection in the mmWave channel, considering the channel's unique properties. In order to address these challenges, the mmW-ROBNet is introduced as a variant of the ROBNet tailored specifically to mmWave channels. This network is trained using a novel hierarchical detection algorithm that results in equitable detection of users based on the respective channel quality.

In Chapter 5 the detection approach is presented for few-bit MIMO systems, beyond one-bit. In particular, the chapter elucidates the differences in detection approaches for systems beyond one-bit ADCs. Notably, it explores the intricacies of detection through signal dequantization, a method unfeasible for one-bit MIMO systems. Moreover, the chapter underscores the role of system design in shaping detection performance, particularly emphasizing the trade-off between resolution and clipping inherent in designing ADC quantization levels. Subsequently, it delves into a subspace-based analysis of both quantized and unquantized signals, leading to the assessment of the probability of dequantization. A DNN-aided network is presented, i.e., the DQuantNet, to unfold the iterations of the alternating optimization, utilizing a learnt regularization step for the quantization noise. This chapter concludes with a discussion on the validity of dequantization-based detection for few-bit MIMO systems and possible extensions to this line of work.

In Chapter 6, the main contributions of this work are summarized, followed by a discussion of possible future directions of extending this research.

Chapter 2

Learning Block-sparse Structure for Clustered Angular Channels

2.1 Introduction

Solving an underdetermined system of linear equations has been heavily studied in the signal processing literature. In many signal acquisition and estimation tasks, the underlying signals are *sparse*. Compressed Sensing (CS) theory [65, 66] has extensively studied stability conditions, algorithms, and convergence for sparse signal estimation. Within the CS paradigm, one particular signal class is *block-sparse* signals where consecutive groups of elements are alternatively nonzero or zero. Block-sparse signal recovery has various applications in wireless communication, audio, and image processing. Our primary interest resides in mmWave channel estimation and modeling where the received signal consists of angular multipath components that impinge on the receive antenna as clustered rays (hence block-sparse) [14–16].

The key challenge with block-sparse recovery is to model inter-element dependency, in addition to the sparsity constraint. We address the most general class of block-sparse signals: block boundaries are *unknown* and the block sizes are, in general, *unequal*. In this setup, the number of possible block combinations involved in the search grows exponentially with the signal length. Imposing all signal boundaries without prior structure and knowledge would prevent scaling to large signal sizes and thus prove inefficient for recovery. On the other hand, employing a conventional CS algorithm with no block structure regularization would clearly underperform by not taking the full advantage of the signal structure. Thus, there is an inherent *trade-off* in incorporating a *prior* signal structure in block-sparse recovery under unknown block

boundaries: pre-defining the search grid enables accurate modeling at the cost of excessive complexity, whereas computationally tractable, flexible modeling of block-sparse priors may hamper the ability to recover arbitrary block structures. This calls for the design of a computationally efficient signal recovery algorithm that imposes an effective, yet *robust* prior to model the underlying non-uniform block-sparsity, which is the main focus of our paper.

2.1.1 Overview of Block-Sparse Signal Recovery

Block-sparse recovery methods have enforced block-patterned structures through block partitioning or efficient inter-element coupling. The early attempts assumed known block sizes and modified conventional CS algorithms to support block-sparse signal recovery. These include Group-Lasso [67], Group Basis Pursuit [68], Model-based CoSaMP [69], Block-OMP [70], and a block ℓ_2-norm based method [71]. Extending to the case of unknown block partitions, algorithms such as Struct-OMP in [72] and the method based on graphical models in [73] were developed. The approach in [73] uses the Boltzmann Machine model to capture inter-element dependencies; the method suffers from high complexity, inhibiting the scaling to large dictionaries and signal dimensions.

Apart from conventional CS approaches, Sparse Bayesian Learning (SBL) [74,75] has shown superior performance for block-sparse recovery, especially in multiple measurement vector (MMV) scenarios. The first work is [76], where the developed block Sparse Bayesian Learning (BSBL) algorithm assumes known block partitions and models temporal signal correlations in an MMV-SBL problem. The work in [77] proposes different optimization methods for the BSBL inference, including the extension to unknown block structures, which however does not follow an elegant optimization framework. Differently, [78] uses Bayesian compressive sensing and incorporates a spike-and-slab prior to model both block-patterned and individual sparsity. The inference in [78] relies on time-consuming Gibbs Sampler and Markov Chain Monte Carlo (MCMC) methods.

The Pattern-coupled SBL (PC-SBL) method [79] initiated a fresh view to incorporate block-sparse structures by coupling the underlying SBL *hyperparameters*. The Non-uniform Burst Sparsity algorithm in [80] improved on PC-SBL inference by using the Variational Bayesian Inference [81]. Coupled priors for block-sparse signal recovery are also used in the Extended-BSBL (EBSBL) method in [82]. Unlike the PC-SBL algorithm, EBSBL algorithm gives an equal weight to the neighbouring parameters, leading to performance superior to BSBL, but inferior to PC-SBL.

All the above coupled-priors-based algorithms incorporate a hyperprior in the SBL parameter space (see, e.g., Gamma hyperprior in [74]), which needs to be specifically tuned for a particular block-sparse

signal structure. Instead of such offline tuning, the hyperparameters could be estimated jointly with the signal. Such an approach is proposed in [83], where the signal and support random variables are modeled via a Gaussian-Bernoulli prior whose hyperparameters are also estimated from the observed data. Since exact update equations are no longer tractable for such models, the MCMC method is used for parameter estimation. Based on a clustered sparsity model, a hybrid clustered sparse prior is introduced in [84]. This framework extends PC-SBL [79] by considering the PC-SBL's coupling coefficient as a hyperparameter which is then adaptively adjusted based on the observed data.

2.1.2 Main Contribution: Total Variation Regularizers for SBL

We address compressed estimation of block-sparse signals in an MMV setup. We propose a novel SBL hyperparameter prior/regularizer to encourage block-patterned sparsity where the underlying sparsity profile is non-uniform and *unknown*, i.e., without any prior knowledge of block sizes and boundaries. To this end, we introduce two classes of *Total Variation* (TV) inspired regularizers that promote contiguous signal and zero regions by enforcing low TV in the hyperparameter space of SBL. The salient features of these regularizers, and the underlying optimization used, are given below:

- The first regularizer class, *Conventional TV Regularizers*, uses the standard TV of the form $T(\mathbf{x}) = \sum_i |x_i - x_{i-1}|$ (i.e., the absolute difference between two consecutive elements) and its variants utilizing sparsity promoting regularizers from the CS domain on the TV penalty. This enforces minimal hyperparameter variation in both the signal and zero regions of a block-sparse signal.

- The second regularizer class, *Region-aware Regularizers*, introduces more robust regularization that penalizes the signal and zero regions differently; this region-awareness stems from the incorporation of a nonlinear hyperparameter transformation $g(\cdot)$, thus creating a general version of the TV as $T(\mathbf{x}) = \sum_i |g(x_i) - g(x_{i-1})|$.

- A majorization-minimization approach is utilized to derive iterative convex solvers for the class of Conventional TV Regularizers. We have analyzed this framework separately in our prior work [85].

- We develop an iterative algorithm for the Expectation-Maximization (EM) based SBL inference using cyclic optimization, tailored for Region-aware Regularizers, and show its convergence properties. The procedure decouples the inference into parallel updates for each hyperparameter component, establishing a computationally efficient optimization method. We further unify the method to build a universal inference strategy for the TV-regularized SBL under both regularization classes.

Numerical experiments show that by inducing a soft, flexible TV prior, the proposed TV-regularized SBL method is *robust* to sparsity structure; the algorithm attains definitive recovery from strict block-sparsity to fully random sparsity in spite of the block-pattern-inducing prior. We additionally note that the block-sparse signal generation is inspired by the real-world applications like mmWave channel modeling. To this end, we assume that the block-sparse signals do not have excessively large amplitude fluctuations within each signal block; the developed TV-SBL accommodates such homogeneous signals for resilient performance.

To the best of our knowledge, this is the first work to apply a TV-type penalty in the hyperparameter space of SBL for the purpose of encouraging block-sparsity. The framework is quite general and allows for more exploration of a wide range of regularizers, especially those inspired by CS.

Organization: The paper is organized as follows. Section 2.2 formulates the block-sparse signal recovery problem through the SBL framework and introduces the two classes of TV regularization for block-sparsity. Section 2.3 presents a general optimization framework for the both regularizer classes. Section 2.4 devises a universal EM-based alternating optimization method, establishes its proof of convergence, and outlines its algorithmic implementation. Numerical experiments are shown in Section 2.5, and Section 5.6 provides concluding remarks.

Notations: Vectors are denoted by lower-case boldface letters ($\mathbf{a}$) and defined as column vectors $\mathbf{a} = [a_1 \cdots a_N]^{\mathrm{T}}$. Matrices are denoted by upper-case boldface letters ($\mathbf{A}$). The vector at ith column of matrix $\mathbf{A}$ is denoted as $\mathbf{a}_i$. Transpose and Hermitian transpose are denoted as $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$, respectively. The trace of matrix $\mathbf{A}$ is denoted as $\mathrm{Tr}(\mathbf{A})$. A diagonal matrix with diagonal entries $a_1, \ldots, a_N$ is denoted by $\mathrm{diag}(a_1, \ldots, a_N)$. The default norm for a vector $||\mathbf{a}||$ is the 2-norm, unless otherwise specified. $||\mathbf{A}||$ denotes the Frobenius norm of matrix $\mathbf{A}$, unless otherwise specified. The ℓ_0-"norm" $\|\mathbf{a}\|_0$ counts the number of nonzero entries of vector $\mathbf{a}$. The notation $(\cdot)^{(k)}$ is used to denote the value at the kth iteration. We use $I(\cdot)$ as a shorthand for the standard indicator function for positive real numbers $\mathbf{1}_{\{\mathbb{R}^+\}}(\cdot)$. The statistical quantities $p(\mathbf{x}|\mathbf{y})$, $\mathbb{E}[\cdot]$, etc. have their usual meaning.

2.2 Total Variation Regularizers for Block-Sparse Signal Recovery via SBL

2.2.1 Sparse Signal Recovery Problem

We consider a multiple measurement vector (MMV) problem where the objective is to simultaneously estimate L unknown source vectors $\mathbf{x}_l \in \mathbb{C}^N$ with a common *block-sparse* structure from a collection

of noisy linear measurements[1]

$$\mathbf{y}_l = \mathbf{A}\mathbf{x}_l + \mathbf{n}_l,\ l = 1, \ldots, L, \tag{2.1}$$

where $\mathbf{y}_l \in \mathbb{C}^M$ is a measurement vector at time instant l, $\mathbf{A} \in \mathbb{C}^{M \times N}$ is a fixed and known measurement matrix (or a basis matrix or dictionary), and $\mathbf{n}_l \sim \mathcal{CN}(\mathbf{0}, \lambda \mathbf{I})$ is a noise vector, independent of $\mathbf{x}_l$. Source vectors and noise vectors are assumed to be independent and identically distributed (i.i.d.) across the time instants.

Because of the block-sparse structure, $\mathbf{x}_l$ consists of successive signal blocks and zero blocks. The same sparsity pattern is shared among the collection of vectors $\{\mathbf{x}_l\}_{l=1}^L$, but the magnitudes of nonzero elements are drawn from an i.i.d. Gaussian distribution. Thus, the signal ensemble $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_L]$ is *block-row-sparse*. We consider the most general (and challenging) block-sparse inference task: both the block locations and (possibly non-uniform) sizes are *unknown*.

Prior to introducing our novel SBL-based method for block-sparse recovery, we provide a brief overview of the general SBL framework for sparse signal recovery.

2.2.2 SBL with Generalized Cost Function

SBL is inspired by the Automatic Relevance Determination mechanism from neural networks [86, 87], providing the means for relatively weighing the importance of different network weights, which could be sparse. There are various advantages for choosing SBL as a sparse signal recovery method for the MMV case:

- The M-SBL [88] parameter estimation abstracts each row of $\mathbf{X}$ by a single (hyper)parameter (γ_i), reducing the number of parameters to be estimated from NL to N compared to CS approaches.
- SBL falls into the category of *correlation-aware* methods which show superior sparse recovery performance [89].
- It has been shown in [75, 90] that the global minimum for the SBL cost function leads to maximally sparse solutions. Additionally, the number of local minima can be bounded above [75]; this bound is typically lower than that for conventional sparse recovery algorithms.

[1]Since our framework is not application-specific, we call each unknown $\mathbf{x}_l$ a source for convenience. The considered MMV setting arises in, e.g., wireless communications: 1) direction of arrival estimation of clustered multipath components at mmWave frequencies, where the block-sparse $\mathbf{x}_l$ represents a vector of beam space angular components of the channel for time instant l, and 2) a user activity detection and channel estimation problem in multi-antenna communications under uncorrelated fading channels with spatially correlated activity patterns, where the block-sparse $\mathbf{x}_l$ is the channel vector associated with N users at time instant l.

- SBL shows great promise for sparse signal recovery under correlated sources and ill-conditioned dictionaries [91, 92]. Such dictionaries (e.g., FFT bases) are often encountered in specific signal recovery applications, like wireless channel estimation.

We now describe the SBL inference. With an additive Gaussian noise model (2.1), the likelihood of the observations $p(\mathbf{y}_l|\mathbf{x}_l)$ is given by the Gaussian likelihood $\mathcal{CN}(\mathbf{A}\mathbf{x}_l, \lambda\mathbf{I})$. The SBL framework [88] assumes a prior distribution on $\mathbf{x}_l$; for each signal $\mathbf{x}_l \in \mathbb{C}^N$, $l = 1, \ldots, L$, this is characterized by a parametric Gaussian distribution $\mathcal{CN}(\mathbf{0}, \boldsymbol{\Gamma})$ as

$$p(\mathbf{x}_l; \boldsymbol{\gamma}) = \tfrac{1}{|\pi\boldsymbol{\Gamma}|}\exp\left(-\mathbf{x}_l^{\mathrm{H}}\boldsymbol{\Gamma}^{-1}\mathbf{x}_l\right), \tag{2.2}$$

where $\boldsymbol{\gamma} = [\gamma_1 \cdots \gamma_N]^{\mathrm{T}} \in \mathbb{R}_+^N$ is a vector of *hyperparameters*, adjusting the variance of each signal component $x_{l,i}$, $i = 1, \ldots, N$, and $\boldsymbol{\Gamma} \triangleq \mathrm{diag}(\gamma_1, \ldots, \gamma_N)$. The hyperparameter values $\boldsymbol{\gamma}$ reflect the sparsity profile of the block-row-sparse $\mathbf{X}$; a suitable prior on $\boldsymbol{\gamma}$ can lead $\mathbf{x}_l$ to model many interesting sparse priors, e.g., Gaussian scale mixtures [74, 93, 94].

The posterior density $p(\mathbf{x}_l|\mathbf{y}_l; \boldsymbol{\gamma})$ is also Gaussian as $\mathcal{CN}(\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}}, \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}})$, where

$$\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}} = \lambda^{-1}\boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}}\mathbf{A}^{\mathrm{H}}\mathbf{y}_l,\ \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}} = \left(\lambda^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{A} + \boldsymbol{\Gamma}^{-1}\right)^{-1}. \tag{2.3}$$

For a given $\boldsymbol{\gamma}$, the estimate of each signal $\{\mathbf{x}_l\}_{l=1}^L$ is formed as $\hat{\mathbf{x}}_{l,\mathrm{SBL}} = \boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}}$ according to (2.3). Following [88], the hyperparameter estimation is done through Type-II Maximum a Posterior (MAP) estimation over $\boldsymbol{\gamma}$ as

$$\begin{aligned}\boldsymbol{\gamma}^* &= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\mathrm{argmax}}\ \log\, p(\boldsymbol{\gamma}|\mathbf{y}_1, \ldots, \mathbf{y}_L) \\ &= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\mathrm{argmin}}\ L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^L \mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l - \log p(\boldsymbol{\gamma}),\end{aligned} \tag{2.4}$$

where $\boldsymbol{\Sigma}_{\mathbf{y}} = \lambda\mathbf{I} + \mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}$ is the measurement model covariance matrix and $\log p(\boldsymbol{\gamma})$ is the *hyperprior* on $\boldsymbol{\gamma}$. The expression in (2.4) is the generalized *MMV-SBL cost function*.

Two important remarks regarding (2.4) are in order.

1. *Role of the prior:* A majority of works consider so-called *non-informative* (i.e., uniform) prior $\log p(\boldsymbol{\gamma})$, which corresponds to type-II Maximum Likelihood estimation above. However, in signal processing tasks, where one possesses some *prior information* of the structure of signal $\mathbf{X}$, an appropriate choice of an *informative* prior of hyperparameter $\boldsymbol{\gamma}$ can significantly improve the inference performance [95–97]. We address the latter by incorporating the knowledge of the underlying block-sparse signal structure into the hyperprior $\log p(\boldsymbol{\gamma})$.

2. *Minimization strategies:* The SBL cost function (2.4) is, in general, *non-convex* in $\boldsymbol{\gamma}$ due to the concave term $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$; convexity of $\log p(\boldsymbol{\gamma})$ depends on the prior. Different minimization strategies for (2.4) have been proposed, along with their convergence guarantees. To this end, we develop and discuss specific optimization strategies tailored for our novel SBL method in Sec. 2.3.

2.2.3 SBL with Novel TV-based Regularizers

Regarding the MMV-SBL cost function (2.4), this section presents our novel SBL priors, i.e., *hyperparameter regularizers*, that will encourage block-sparse signal recovery. We introduce two regularization classes inspired by **Total Variation (TV)** and various CS-based sparse regularizers. TV has been used extensively in image processing for regularization and denoising [98–100]. TV regularization has also been used for group-sparse recovery along with the sparsity-inducing regularizers (like ℓ_1-norm penalty) in such image recovery tasks [101–106]. These approaches enforce structure by working directly on signal $\mathbf{x}_l$ – a more challenging and less efficient approach for the complicated block-sparsity problem.

Differently, our approach relies on enforcing special structures on the hyperparameter vector $\boldsymbol{\gamma}$. It is noteworthy that imposing the regularizer on the hyperparameters rather than the source vectors $\mathbf{x}_l$ is an important distinction and also key to the success of our approach. Block-sparse recovery algorithms have been developed introducing priors on the hyperparameter space (see Sec. 2.1.1). The BSBL [77] algorithm is tailored to fixed block sizes, thus limiting its utility for more flexible block-sparse recovery. The algorithms using rigid SBL hyperparameter coupling [79,80], although highly effective for flexible block-sparse recovery, are herein shown to be sensitive to recovering signals with both block-sparse and isolated sparse components; these stronger priors are biased to block structures in the underlying signal. We aim to bridge this gap by introducing a softer TV-inspired prior $p(\boldsymbol{\gamma})$ on the hyperparameter space. This circumvents excessive coupling bias on the signal and makes the proposed method extremely *robust* and capable of accommodating both block-sparse and isolated sparse components. Empirical evidence for our claims is provided through the simulation results for different classes of block-sparse signals in Sec. 2.5.

Consider the MMV-SBL cost function (2.4). We denote the hyperprior as $\tilde{\beta}\,T(\boldsymbol{\gamma}) \triangleq -\log p(\boldsymbol{\gamma})$, where $\tilde{\beta}$ is a non-negative weighting parameter controlling the emphasis on the prior, and $T(\cdot)$ is a general TV-type penalty of vector $\boldsymbol{\gamma}$, which will become explicit in sequel. Accordingly, we will minimize the *TV-regularized MMV-SBL cost function*

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\; L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l + \tilde{\beta}\,T(\boldsymbol{\gamma}). \tag{2.5}$$

Regardless of the form of $T(\cdot)$, we refer to our proposed method collectively as **TV-SBL**.

Recall from (2.4) that $\tilde{\beta}$ in (2.5) serves as the interface on how much weight the data $\mathbf{Y}$ carries in (the optimization over) the posterior distribution versus the weight carried by the prior information about the signal structure, captured by $T(\boldsymbol{\gamma})$. This brings us to motivate the choice of an appropriate $T(\cdot)$. The first thing to stress is that the purpose of the regularization $T(\cdot)$ is not to promote sparse solutions per se. Namely, in the canonical part of the SBL cost function in (2.5), the log det term $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$ takes the role of imposing a sparsity penalty via the principles of automatic relevance determination. Thus, the role of $T(\cdot)$ is to *augment* the SBL's objective to encourage *special* sparse structures, herein, block-sparse structures.

Conventional CS techniques induce sparsity via minimization of the ℓ_0-"norm" $||\mathbf{x}||_0 = \sum_{i=1}^{N} I(|x_i|)$. Many surrogate measures have been used, the most common one being the ℓ_1-norm in CS [107, 108]. We harness this indicator function perspective to promote block-sparsity. By extending this element-counting function to a *block-counting function*, we introduce two novel classes of regularizers $T(\cdot)$: *(i) Conventional TV Regularization* and *(ii) Region-aware Regularization using TV*. We introduce three analytical TV-SBL regularizers – Linear TV, Log TV and DoL TV – with the first two belonging to the conventional TV regularization class, and the third to the region-aware regularization class. We elaborate further below.

Conventional TV Regularization

Our initial approach defines a block (either signal or zero block) as a region with constant γ_i's, analogous to the BSBL algorithm in [76], yet without the knowledge of the true block distribution. We define the related regularizer using the block-counting function that is equal to TV on $\boldsymbol{\gamma}$, i.e.,

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} I(|\gamma_i - \gamma_{i-1}|). \tag{2.6}$$

Using equal variances for the entries within a block, the measure (2.6) counts the number of edges in the underlying block structure. Note that even if the regularizer (2.6) tends to enforce equal values of γ_i's for the consecutive nonzero entries, this does not stringently translate to restraining the corresponding signal entries $x_{l,i}$ to have equal magnitudes. This is the main asset in regularizing the hyperparameters, not the signal magnitudes. Thus, while simple, the imposed hierarchical prior structure admits significant boost in recovering various block-sparse signals, as demonstrated by the numerical results in Sec. 2.5.

Armed with the ideal measure (2.6), we can use *tractable measures* developed in CS on the TV input variable $|\gamma_i - \gamma_{i-1}|$ to identify appropriate block structures. CS theory has developed many regularizers that are monotonically increasing and concave on the positive orthant to promote sparsity. We investigate two such surrogate functions, introducing our two conventional TV regularizers addressed in this paper.

Linear TV

The linear TV regularizer is equivalent to the ℓ_1 penalty in CS and is given by the form

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |\gamma_i - \gamma_{i-1}|. \tag{2.7}$$

As stated earlier, TV has also been used in different signal processing applications to preserve edges and enforce local smoothness. We use this convex regularizer to enforce a block structure in the recovered signal. In addition to the signal regions, this penalty is found to "denoise" the zero regions more effectively than the unregularized SBL algorithm [85].

Log TV

Another widely used regularizer in CS is $\sum_{i=1}^{N} \log(|x_i|+\epsilon)$, where ϵ is a positive stability parameter. This log-sum regularizer employs an iterative reweighted ℓ_1 minimization algorithm and has been shown to yield superior recovery [108, 109]. Utilizing this regularizer for block-sparsity, the log TV regularizer is given by

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} \log(|\gamma_i - \gamma_{i-1}| + \epsilon). \tag{2.8}$$

As in the CS literature, the log TV based approach is found to be more effective than the linear TV, using the MM optimization framework (see Sec. 2.3.1) [85]. This is due to its better resemblance to ℓ_0-"norm" [108], allowing more signal variance differences within a block and restraining small (faulty) signal estimate components to emerge.

The block-counting function (2.6) enforces blocks quite rigidly through constant γ_i values, irrespective of a block being a zero or nonzero block. A more flexible block-counting function, explained next, incorporates this crucial difference between zero and nonzero blocks.

Region-aware Regularization using TV (RAR TV)

We introduce a new terminology here, namely, zero and signal *regions*. Inspired by a pictorial representation of block-sparse signals (e.g., block-sparse angular regions for mmWave channels), a signal region – as a broader description of a signal block – consists of a contiguous sequence of nonzero entries, which are not necessarily associated with the same γ_i value. We have empirically observed that there exists a dissimilarity in the recovery of zero and signal regions of the sparse vector $\mathbf{x}_l$. We modify our original block-counting

function (2.6) to incorporate this differentiation as

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |I(\gamma_i) - I(\gamma_{i-1})|. \tag{2.9}$$

The block-counting function (2.9), unlike (2.6), will minimize the number of transitions between a signal region and a zero region, while allowing arbitrary variation of signal magnitudes within the blocks of the signal region. Further, this will enforce strict contiguous zero blocks; hence the name *Region-aware Regularization (RAR)*. This differential treatment of the zero and nonzero regions translates to a nonlinear scaling of the hyperparameter components. More explicitly, we use a tractable relaxation of (2.9) as

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |g(\gamma_i) - g(\gamma_{i-1})|, \tag{2.10}$$

where $g(\cdot)$ is a non-decreasing concave function in the positive quadrant, which (relatively) amplifies lower γ_i values while scaling down the higher values corresponding to large signal components. We term the penalty in (2.10) as the *Difference of Functions (DoF)* regularizer. We use this form, with its concave non-decreasing behavior, for the optimization analysis of TV-SBL in Sec. 2.4. Prior to this, we introduce one specific realization of $g(\cdot)$ for the DoF penalty, which will be used to illustrate the utility of the RAR strategy.

Difference of Logs (DoL) TV

Following the above arguments for the utility of the $\log(\cdot)$ transform in CS as surrogate measures of the ℓ_0-"norm", we propose a novel *Difference of Logs (DoL)* TV penalty as

$$T(\boldsymbol{\gamma}) = \sum_{i=2}^{N} |\log(\gamma_i) - \log(\gamma_{i-1})|. \tag{2.11}$$

Here, the role of the $\log(\cdot)$ function[2] is to nonlinearly scale the hyperparameter vector $\boldsymbol{\gamma}$ to differentially treat the zero and signal regions, which is the main feature required for $g(\cdot)$, as motivated above. We remark that other tractable surrogate functions $g(\cdot)$ that better resemble the indicator function $I(\cdot)$ in (2.9) are worth studying as future work.

Remark. *One feature of interest here is that the penalty* $|\log(\gamma_i) - \log(\gamma_{i-1})|$ *in* (2.11) *alternatively reduces the TV in the hyperparameter space using the ratio* $\frac{\gamma_i}{\gamma_{i-1}} \to 1$. *The analysis of the TV reduction by matching*

[2]Some of our experiments (not provided herein) have shown that the concave square root penalty $g(\gamma_i) = \sqrt{\gamma_i}$ is another potential variant for (2.10), performing similar to the $\log(\cdot)$ penalty.

Table 2.1: Comparison of Key Properties for TV-SBL Regularizers

Property	**Linear TV** (2.7)	**Log TV** (2.8)	**DoL TV** (2.11)
Block-counting function	Conventional TV (2.6)	Conventional TV (2.6)	Region-aware TV (2.9)
Analogous CS regularizer	ℓ_1-norm regularization	log-sum regularization	No conventional CS regularizer
Block-sparse enforcement	Minimize γ_i variation	Minimize γ_i variation	Minimize γ_i zero–nonzero transition
Block type symmetry	Invariant to block type	Invariant to block type	Differential zero–nonzero γ_i regularizer
Convex optimization solution	Possible through MM	Possible through MM	Not possible (EM-based optimization used)
Performance	Weakest TV-SBL regularizer	Stronger than Linear TV	Strongest TV-SBL regularizer

the ratio to 1 warrants further elaboration and discussion, and it is left for future work.

Remark. *The value of regularization weight parameter β has clearly a significant impact on the TV-SBL's performance and needs thus appropriate adjustment. As for any regularization-based method, having a systematic way of defining the optimal β is elusive. Pragmatically, if the estimate of $\boldsymbol{\gamma}$ and, consequently, the estimate of $\mathbf{x}_l$ tend to exclusively show isolated sparse components for a block-sparse $\mathbf{X}$, increasing β promotes the blocks to emerge. On the contrary, if the estimate vectors are non-sparse with overly smooth envelopes, β should be decreased. We remark that configuring β could also be incorporated into a learning framework. Specific tuning of β is outside of the scope of this work.*

Through this section, we have introduced three novel block-sparsity-inducing regularizers: (2.7), (2.8), and (2.11). The key properties highlighting the differences between these regularizers are summarized in Table 2.1. We now move on to analyze the optimization of the TV-regularized MMV-SBL cost function (2.5) under each regularizer.

2.3 Optimization Framework for TV-SBL

SBL optimization has been extensively studied in various contexts. The original approach was proposed by Tipping [74], using fixed point iteration. The most popular algorithm for SBL optimization is the Expectation-Maximization (EM) algorithm, as used in [75, 76, 88]. Another common approach is the Majorization-Minimization (MM) framework [85,110] that relaxes the non-convex optimization problem (2.4) into a sequence of convex optimization problems. Other strategies include the iterative reweighted ℓ_1 and ℓ_2 methods [93] and Generalized Approximate Message Passing (GAMP) [92].

The fundamental difference in the underlying block-counting measures between the Conventional

TV in (2.6) and RAR TV in (2.9) classifies the optimization procedures for the TV-SBL cost (2.5) under their respective labels.

1. **Conventional TV Optimization:** The linear and log TV penalties in (2.7) and (2.8) can be tackled by convex optimization tools after appropriately *relaxing* the problem (2.5), e.g., using the MM framework. We address such MM-based optimization in Sec. 2.3.1.

2. **RAR TV Optimization:** The DoL penalty in (2.11) precludes the use of convex solvers, as will be discussed in Sec. 2.3.2. We approach this by splitting the overall optimization into two sequential sub-problems, and combining the solutions. We use the EM-based inference to generate closed-form update expressions. The EM preliminaries are presented in Sec. 2.3.3 and our overall optimization procedure is detailed in Sec. 2.4.

Both these frameworks reformulate the original TV-SBL cost (2.5), to be handled by different optimization tools.

2.3.1 Conventional TV-SBL through Convex Optimization

There are many options for minimizing the general SBL objective function. We apply the majorization-minimization (MM) approach and derive an iterative algorithm for minimizing the TV-SBL cost.

Linear TV

he TV-SBL optimization (2.4) for the Linear TV regularizer in (2.7) is

$$\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\, L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L} \mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l + \beta \sum_{i=2}^{N}|\gamma_i - \gamma_{i-1}|. \tag{2.12}$$

Using the MM technique similar to [110], we majorize the concave term $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$ and solve iteratively a sequence of convex optimization problems. We majorize (i.e., linearize) $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$ by its first-order Taylor approximation at point $\boldsymbol{\Gamma}^{(j)}$, i.e.,

$$\begin{aligned}&\log|\lambda\mathbf{I} + \mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}| \le \log|\lambda\mathbf{I} + \mathbf{A}\boldsymbol{\Gamma}^{(j)}\mathbf{A}^{\mathrm{H}}| + \\ &\operatorname{Tr}\big((\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)})^{-1}\mathbf{A}\mathbf{A}^{\mathrm{H}}[\boldsymbol{\Gamma} - \boldsymbol{\Gamma}^{(j)}]\big),\end{aligned} \tag{2.13}$$

where the superscript j denotes the MM iteration index. Using (2.13), at iteration j, we end up with solving the convex problem

$$\begin{aligned}\boldsymbol{\gamma}^{(j+1)} = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\ L\operatorname{Tr}\left(\left(\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}\right)^{-1}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}\right)\\ &+\textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l+\beta\sum_{i=2}^{N}|\gamma_i-\gamma_{i-1}|,\end{aligned} \tag{2.14}$$

and then updating $\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}$ using the newly obtained $\boldsymbol{\gamma}^{(j+1)}$.

Log TV

The TV-SBL optimization (2.4) for the Log TV regularizer in (2.8) is

$$\begin{aligned}\boldsymbol{\gamma}^{*} = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\ L\log|\boldsymbol{\Sigma}_{\mathbf{y}}|+\textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l\\ &+\beta\textstyle\sum_{i=2}^{N}\log(|\gamma_i-\gamma_{i-1}|+\epsilon).\end{aligned} \tag{2.15}$$

Similar to the Linear TV case above, we apply the MM approach for (2.15). Besides majorizing the $\log|\boldsymbol{\Sigma}_{\mathbf{y}}|$ term via (2.13), we majorize the concave Log TV penalty (2.8) by its first-order Taylor approximation at points $(\gamma_i^{(j)}-\gamma_{i-1}^{(j)})$, $i=2,\ldots,N$, i.e.,

$$\log(|\gamma_i-\gamma_{i-1}|+\epsilon)\leq\log(|\gamma_i^{(j)}-\gamma_{i-1}^{(j)}|+\epsilon)+\tfrac{|\gamma_i-\gamma_{i-1}|}{|\gamma_i^{(j)}-\gamma_{i-1}^{(j)}|+\epsilon}. \tag{2.16}$$

Thus, at iteration j, we solve the convex problem

$$\begin{aligned}\boldsymbol{\gamma}^{(j+1)} = &\underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\ L\operatorname{Tr}\left(\left(\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}\right)^{-1}\mathbf{A}\boldsymbol{\Gamma}\mathbf{A}^{\mathrm{H}}\right)\\ &+\textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l+\beta\sum_{i=2}^{N}\frac{1}{|\gamma_i^{(j)}-\gamma_{i-1}^{(j)}|+\epsilon}|\gamma_i-\gamma_{i-1}|,\end{aligned} \tag{2.17}$$

followed by updating $\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}$ using the newly obtained $\boldsymbol{\gamma}^{(j+1)}$.

Convex Solver Implementation of TV-SBL

Any convex optimization package can be implemented to solve (2.14) and (2.17). Algorithm 8 presents the implementation of TV-SBL via the widely used CVX optimization package [111] to facilitate easy adoption and experimentation. One key step is to handle the matrix inverse in $\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l$ through the Schur's complement equivalence [112, Appendix A5.5] by introducing the Hermitian symmetric matrix variables $\mathbf{Z}_{\mathrm{cvx},l}\in\mathbb{S}^{M\times M}$, $l=1,\ldots,L$.

Algorithm 1: CVX Solver for TV-SBL

Input: $\mathbf{A}, \mathbf{Y}, \boldsymbol{\gamma}^{(0)}$, λ, β, ϵ, $j_{\max}$
Output: $\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}}$ $(\forall\, l \in \{1,\ldots,L\}$

for $j = 0$ **to** $j_{\max} - 1$ **do**
  Evaluate $[\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}]^{-1} = (\lambda \mathbf{I} + \mathbf{A}\boldsymbol{\Gamma}^{(j)}\mathbf{A}^{\mathrm{H}})^{-1}$;
  CVX variables: $\boldsymbol{\gamma}_{\mathrm{cvx}} \in \mathbb{R}^N$, $\mathbf{Z}_{\mathrm{cvx},l} \in \mathbb{S}^{M\times M}$, $l = 1,\ldots,L$;
  minimize: $L\,\mathrm{Tr}\big[\big(\boldsymbol{\Sigma}_{\mathbf{y}}^{(j)}\big)^{-1}\mathbf{A}\boldsymbol{\Gamma}_{\mathrm{cvx}}\mathbf{A}^{\mathrm{H}}\big] + \sum_{l=1}^{L}\mathrm{Tr}(\mathbf{Z}_{\mathrm{cvx},l}) + \beta\, T(\boldsymbol{\gamma}_{\mathrm{cvx}})$;
  subject to: $\boldsymbol{\gamma}_{\mathrm{cvx}} \succeq \mathbf{0}$, $\begin{bmatrix} \mathbf{Z}_{\mathrm{cvx},l} & (\mathbf{y}_l\mathbf{y}_l^{\mathrm{H}})^{1/2} \\ (\mathbf{y}_l\mathbf{y}_l^{\mathrm{H}})^{1/2} & \lambda\mathbf{I} + \mathbf{A}\boldsymbol{\Gamma}_{\mathrm{cvx}}\mathbf{A}^{\mathrm{H}} \end{bmatrix} \succeq \mathbf{0}$ $(\forall\, l \in \{1,\ldots,L\})$;
  $\boldsymbol{\gamma}^{(j+1)} \leftarrow \boldsymbol{\gamma}_{\mathrm{cvx}}$;
end
Evaluate $\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}}$ $(\forall\, l \in \{1,\ldots,L\})$ using $\boldsymbol{\gamma}^{(j_{\max})}$ in (2.3);

2.3.2 Difficulty in Majorizing DoL TV Penalty

Consider the DoL penalty in (2.11). The term $|\log(\gamma_i) - \log(\gamma_{i-1})|$ is concave in γ_i for $\gamma_i > \gamma_{i-1}$ and convex in γ_i for $\gamma_i < \gamma_{i-1}$. Thus, unlike the linear and log TV in (2.12) and (2.15) respectively, we cannot find a universal majorizer for the DoL function. In fact, this holds true for any DoF TV penalty in (2.10) with a concave function $g(\cdot)$. Until further notice, we proceed with the DoF regularizer (2.10) while keeping in mind that its special case $g(\cdot) = \log(\cdot)$ realizes the DoL in (2.11).

The TV-SBL optimization (2.5) under the DoF TV penalty in (2.10) reads as

$$\begin{aligned}\boldsymbol{\gamma}^* = \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\operatorname{argmin}}\, & L\log|\boldsymbol{\Sigma}_{\mathbf{y}}| + \textstyle\sum_{l=1}^{L}\mathbf{y}_l^{\mathrm{H}}\boldsymbol{\Sigma}_{\mathbf{y}}^{-1}\mathbf{y}_l \\ & + \tilde{\beta}\,\textstyle\sum_{i=2}^{N}|g(\gamma_i) - g(\gamma_{i-1})|.\end{aligned} \tag{2.18}$$

We solve (2.18) with the EM method. In particular, as it will be seen in Sec. 2.4, this converts the minimization (2.18) into a form that can be solved efficiently using cyclic optimization. Before elaborating further, we describe the general EM procedure used in the TV-SBL optimization.

2.3.3 TV-SBL Inference through Expectation-Maximization

As opposed to a single-shot minimization (2.18), the EM algorithm finds the estimate of the hyperparameter vector $\boldsymbol{\gamma}$ iteratively. We follow the EM-SBL framework, introduced in [75] and extended to MMV in [88]. Using the standard EM theory [113], $\mathbf{Y}$ is the observation variable, $\mathbf{X}$ is the hidden variable and $\boldsymbol{\gamma}$ is the unknown parameter to be estimated. Each of the variables $\mathbf{Y}, \mathbf{X}$, and $\boldsymbol{\gamma}$ is as determined in (2.1) and (2.2).

E-step

Treating $(\mathbf{Y}, \mathbf{X})$ as the complete data and using the Markov chain $\boldsymbol{\gamma} \to \mathbf{X} \to \mathbf{Y}$, the joint distribution gives

$$\log p(\mathbf{X}, \mathbf{Y}, \boldsymbol{\gamma}) = \log p(\mathbf{Y}|\mathbf{X}) + \log p(\mathbf{X}|\boldsymbol{\gamma}) + \log p(\boldsymbol{\gamma}). \tag{2.19}$$

The Q-function is evaluated by averaging out the hidden variable $\mathbf{X}$ as

$$\begin{aligned} Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{(k)}) &= \mathbb{E}_{\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}}\left[\log p(\mathbf{Y}, \mathbf{X}, \boldsymbol{\gamma})\right] \\ &= \mathbb{E}_{\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}}\left[\log p(\mathbf{Y}|\mathbf{X}) + \log p(\mathbf{X}|\boldsymbol{\gamma}) + \log p(\boldsymbol{\gamma})\right] \\ &= \mathbb{E}_{\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}}\left[\log p(\mathbf{Y}|\mathbf{X}) + \log \textstyle\prod_{l=1}^{L} p(\mathbf{x}_l|\boldsymbol{\gamma})\right] \\ &\quad + \log p(\boldsymbol{\gamma}), \end{aligned} \tag{2.20}$$

where the expectation is with respect to the posterior $p(\mathbf{X}|\mathbf{Y};\boldsymbol{\gamma}^{(k)}) = \prod_{l=1}^{L} p(\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k)})$. Removing the terms that do not depend on $\boldsymbol{\gamma}$, and using the result from [88] to evaluate the posterior expectation, we write the E-step as

$$\begin{aligned} Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{(k)}) &= \textstyle\sum_{i=1}^{N}\Big[- L\log\gamma_i - \sum_{l=1}^{L}\frac{E_{l,i}^{(k)}}{\gamma_i}\Big] + \log p(\boldsymbol{\gamma}) \\ &= L\Big\{\textstyle\sum_{i=1}^{N}\Big[-\frac{E_i^{(k)}}{\gamma_i} - \log\gamma_i\Big] - \frac{\tilde{\beta}}{L}T(\boldsymbol{\gamma})\Big\}, \end{aligned} \tag{2.21}$$

where, using (2.3), we defined the quantities for iteration k as

$$\begin{aligned} E_i^{(k)} &= \frac{1}{L}\textstyle\sum_{l=1}^{L} E_{l,i}^{(k)} \\ &= \frac{1}{L}\textstyle\sum_{l=1}^{L}\Big[\big[\boldsymbol{\mu}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k)}}(i)\big]^2 + \boldsymbol{\Sigma}_{\mathbf{x}|\mathbf{y};\boldsymbol{\gamma}^{(k)}}(i,i)\Big]. \end{aligned} \tag{2.22}$$

Remark. *For the subsequent analysis, we use the regularization normalized by the number of snapshots, i.e.,* $\beta \triangleq \tilde{\beta}/L$.

Specializing (2.21) to the DoF TV regularizer in (2.10), we re-define the E-step cost function as

$$J^{(k)}(\boldsymbol{\gamma}) \triangleq \textstyle\sum_{i=1}^{N} f_i^{(k)}(\gamma_i) + \beta\,\sum_{i=2}^{N}|g(\gamma_i) - g(\gamma_{i-1})|, \tag{2.23}$$

where, for brevity, we introduced a function

$$f_i^{(k)}(\gamma_i) \triangleq \frac{E_i^{(k)}}{\gamma_i} + \log\gamma_i. \tag{2.24}$$

$$
\begin{aligned}
J^{(k)}(\boldsymbol{\gamma}) &= \sum_{i\in\text{even}} \Big[f_i^{(k)}(\gamma_i) + \beta \ \big(\ \big|g(\gamma_i) - g(\gamma_{i-1})\big| + \big|g(\gamma_i) - g(\gamma_{i+1})\big| \ \big) \Big] + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i) \\
&= \sum_{i\in\text{odd}} \Big[f_i^{(k)}(\gamma_i) + \beta \ \big(\ \big|g(\gamma_i) - g(\gamma_{i-1})\big| + \big|g(\gamma_i) - g(\gamma_{i+1})\big| \ \big) \Big] + \sum_{i\in\text{even}} f_i^{(k)}(\gamma_i).
\end{aligned} \tag{2.26}
$$

M-step

$$
\begin{aligned}
\boldsymbol{\gamma}^{(k+1)} &= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\text{argmax}}\, Q(\boldsymbol{\gamma}|\boldsymbol{\gamma}^{(k)}) \\
&= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\text{argmin}}\, J^{(k)}(\boldsymbol{\gamma}) \\
&= \underset{\boldsymbol{\gamma}\succeq\mathbf{0}}{\text{argmin}}\, \textstyle\sum_{i=1}^{N} f_i^{(k)}(\gamma_i) + \beta \sum_{i=2}^{N} |g(\gamma_i) - g(\gamma_{i-1})|.
\end{aligned} \tag{2.25}
$$

Using EM, we have converted the original TV-SBL optimization from (2.18) to iterative updates as per (2.25). It is important to note that besides $J^{(k)}(\boldsymbol{\gamma})$ being non-convex in $\boldsymbol{\gamma}$, the TV term introduces *coupling* of the hyperparameters, preventing updates of $\{\gamma_i\}_{i=1}^N$ in parallel. Consequently, the computational complexity of vector optimization in (2.25) may become excessive for high-dimensional signal setups. Therefore, in the next section, we propose an alternating optimization framework that enables solving the M-step (2.25) efficiently via element-wise parallel updates.

2.4 Alternating Optimization for Solving the M-Step of the EM-based TV-SBL

Our alternating optimization procedure for TV-SBL, detailed in this section, has been inspired by the coordinate descent optimization methods [114–118]. We extend these methods to handle our EM-based TV-SBL framework described in Sec. 2.3.3. First, the detailed derivation is carried out for the M-step of DoF TV-SBL in (2.25) (for which the MM approach is not applicable). The section ends with establishing *a universal EM-based TV solver*, providing alternative solvers to the linear TV optimization (2.12) and log TV optimization (2.15) apart from their MM solutions in Sec. 2.3.1.

2.4.1 Alternating Optimization and Convergence

In order to use the techniques from coordinate descent to solve (2.25), we first reformulate the cost function $J^{(k)}(\boldsymbol{\gamma})$ in (2.23). We begin by expressing the TV part $T(\boldsymbol{\gamma})$ as summations over the **even and odd indices**, as shown in (2.26). The form (2.26) rewrites the M-step optimization (2.25) as a summation over two disjoint sets of the elements of $\boldsymbol{\gamma}$, i.e., the even and odd index elements of $\boldsymbol{\gamma}$. This provides an ideal setting to use alternating optimization and separately optimize over the even and odd indices. For this

alternating optimization procedure, we define the following function in relation to (2.26):

$$\begin{aligned} &F_i^{(k)}(\gamma_i;\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) \triangleq f_i^{(k)}(\gamma_i)+ \\ &\beta\ \left(\left|g(\gamma_i)-g(\gamma_{i-1}^{(k)})\right|+\left|g(\gamma_i)-g(\gamma_{i+1}^{(k)})\right|\right), \end{aligned} \tag{2.27}$$

where $\gamma_{i-1}^{(k)}$ and $\gamma_{i+1}^{(k)}$ are fixed quantities at EM iteration k.

The following theorem presents our proposed EM-based alternating optimization method for DoF TV-SBL along with its convergence properties. In particular, the following update equations form the M-step (see (2.25)) tailored to TV-SBL.

Theorem 1. *For $F_i^{(k)}(\gamma_i;\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)})$ defined by (2.27), if the alternating optimization at iteration k is carried out as*

$$\begin{aligned} \gamma_i^{(k+1)} &= \underset{\gamma_i>0}{\operatorname{argmin}}\, F_i^{(k)}(\gamma_i;\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}),\ \forall i\ \textit{even} \\ \gamma_i^{(k+1)} &= \underset{\gamma_i>0}{\operatorname{argmin}}\, F_i^{(k)}(\gamma_i;\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}),\ \forall i\ \textit{odd}, \end{aligned} \tag{2.28}$$

the cost function (2.23) decreases over each iteration k, i.e.,

$$J^{(k)}(\boldsymbol{\gamma}^{(k+1)}) \leq J^{(k)}(\boldsymbol{\gamma}^{(k)}). \tag{2.29}$$

Proof. The proof is given in Appendix 2.A. □

By means of Theorem 1, we decouple the general DoF TV penalty and provide N separate update equations corresponding to each hyperparameter element γ_i. Moreover, by using this two-step alternating optimization over the even and odd indices of $\boldsymbol{\gamma}$, the M-step cost function (2.25) decreases over each iteration as per (2.29), which subsequently results in convergence to a local minimum of problem (2.25).

Owing to the generality of the cost function $J^{(k)}(\boldsymbol{\gamma})$ in terms of the choice of concave function $g(\cdot)$, it may not always be possible to find closed-form solutions to the minimization steps in (2.28). However, the following corollary asserts that for a convergent algorithm determined through (2.29), it is sufficient to obtain only a decrease in the optimization steps.

Corollary 1.1. *If the updated parameters $\boldsymbol{\gamma}^{(k+1)}$ satisfy*

$$\begin{aligned} &F_i^{(k)}(\gamma_i^{(k+1)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) \leq F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}),\ \forall i\ \textit{even}, \\ &F_i^{(k)}(\gamma_i^{(k+1)};\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}) \leq F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}), \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall i\ \textit{odd}, \end{aligned}$$

then this is sufficient to imply (2.29).

Proof. The proof follows from the steps similar to those for Theorem 1. In particular, the inequalities (2.38) and (2.40) do not require the obtained $\gamma_i^{(k+1)}$, $\forall i$, in (2.28) to be strict minimizers. □

Corollary 1 implies that we can use different optimization methods for (2.28), ranging from gradient descent to MM, to approximately solve the M-step of the TV-SBL. Having described the alternating optimization recipe for the TV-SBL, we now present its specific iterative algorithmic implementation.

2.4.2 Algorithm Implementation: Segment-wise Parallel Updates

This section elaborates the steps to solve each optimization sub-problem in (2.28) over the even and odd indices. We exploit the nature of the absolute value function $|\cdot|$ of the DoF regularizer to break each minimization here into individual segments, optimize these separately, and then combine the results. Hence, we refer to our method as the *Segment-wise Parallel Update* algorithm. We begin with the update equation for the even indices in (2.28); the optimization for the odd indices will follow analogously.

We start by introducing quantities that order the neighbouring elements for each $\gamma_i^{(k)}$:

$$\gamma_{i,\max}^{(k)} \triangleq \max\big(\gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}\big),\ \gamma_{i,\min}^{(k)} \triangleq \min\big(\gamma_{i-1}^{(k)}, \gamma_{i+1}^{(k)}\big). \tag{2.30}$$

Using (2.30), the optimization problem for the even indices in (2.28) reads as

$$\begin{aligned}\gamma_i^{(k+1)} &= \underset{\gamma_i>0}{\operatorname{argmin}}\, F_i^{(k)}(\gamma_i; \gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}) \\ &= \underset{\gamma_i>0}{\operatorname{argmin}} \Big[f_i^{(k)}(\gamma_i) + \beta\big(\ \big|g(\gamma_i) - g(\gamma_{i,\max}^{(k)})\big| \\ &\qquad\qquad + \big|g(\gamma_i) - g(\gamma_{i,\min}^{(k)})\big|\ \big)\Big],\quad \forall i \text{ even}.\end{aligned} \tag{2.31}$$

The objective function of (2.31) is differentiable, except at points $\gamma_i \in \{\gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}\}$. To overcome the non-differentiability, we first define three mutually exclusive and collectively exhaustive *segments* on positive reals ($\mathbb{R}_+$):

$$\begin{aligned}\hat{\Gamma}_i &= \big\{\gamma_i \mid \gamma_i > \gamma_{i,\max}^{(k)} \geq 0\big\}, \\ \tilde{\Gamma}_i &= \big\{\gamma_i \mid \gamma_{i,\max}^{(k)} \geq \gamma_i \geq \gamma_{i,\min}^{(k)} \geq 0\big\}, \\ \bar{\Gamma}_i &= \big\{\gamma_i \mid \gamma_{i,\min}^{(k)} > \gamma_i \geq 0\big\},\end{aligned} \tag{2.32}$$

wherein this function is continuous and differentiable. Using (2.32), the optimization problem (2.31) is

equivalent to

$$\gamma_i^{(k+1)} = \underset{\gamma_i \in \left\{\hat{\gamma}_i^{(k)}, \tilde{\gamma}_i^{(k)}, \bar{\gamma}_i^{(k)}\right\}}{\operatorname{argmin}} F_i^{(k)}(\gamma_i; \gamma_{i,\max}^{(k)}, \gamma_{i,\min}^{(k)}), \ \forall i \text{ even}, \tag{2.33}$$

where the three "*candidate*" solutions $\{\hat{\gamma}_i^{(k)}, \tilde{\gamma}_i^{(k)}, \bar{\gamma}_i^{(k)}\}$ are found by separate *segment-wise optimization* problems

$$\begin{aligned}
\hat{\gamma}_i^{(k)} &\triangleq \underset{\gamma_i \in \hat{\Gamma}_i}{\operatorname{argmin}} f_i^{(k)}(\gamma_i) + 2\beta g(\gamma_i) - \beta\big[g(\gamma_{i,\max}^{(k)}) + g(\gamma_{i,\min}^{(k)})\big], \\
\tilde{\gamma}_i^{(k)} &\triangleq \underset{\gamma_i \in \tilde{\Gamma}_i}{\operatorname{argmin}} f_i^{(k)}(\gamma_i) + \beta\big[g(\gamma_{i,\max}^{(k)}) - g(\gamma_{i,\min}^{(k)})\big], \\
\bar{\gamma}_i^{(k)} &\triangleq \underset{\gamma_i \in \bar{\Gamma}_i}{\operatorname{argmin}} f_i^{(k)}(\gamma_i) - 2\beta g(\gamma_i) + \beta\big[g(\gamma_{i,\max}^{(k)}) + g(\gamma_{i,\min}^{(k)})\big].
\end{aligned} \tag{2.34}$$

The constrained problems in (2.34) present our general approach to solve the TV-SBL problem for any general DoF TV penalty (2.10). However, these might be difficult to solve, especially through closed-form expressions. Fortunately, for certain controlled cases, like the DoL TV penalty (2.11), we can solve them equivalently, yet more efficiently, through the following two-step procedure: an *unconstrained update* followed by *projection* to the relevant segment.

Step 1: Unconstrained update

$$\begin{aligned}
\hat{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} f_i^{(k)}(\gamma_i) + 2\beta g(\gamma_i) \\
\tilde{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} f_i^{(k)}(\gamma_i) \\
\bar{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} f_i^{(k)}(\gamma_i) - 2\beta g(\gamma_i).
\end{aligned} \tag{2.35}$$

Step 2: Segment-wise projection

$$\begin{aligned}
\hat{\gamma}_i^{(k)} &= \max\big(\gamma_{i,\max}^{(k)}, \hat{\alpha}_i^{(k)}\big) \\
\tilde{\gamma}_i^{(k)} &= \min\big(\gamma_{i,\max}^{(k)}, \max\big\{\gamma_{i,\min}^{(k)}, \tilde{\alpha}_i^{(k)}\big\}\big) \\
\bar{\gamma}_i^{(k)} &= \max\big(0, \min\big\{\gamma_{i,\min}^{(k)}, \bar{\alpha}_i^{(k)}\big\}\big).
\end{aligned} \tag{2.36}$$

Elaborate justification of this segment-wise parallel update optimization strategy is presented in Sec. 2.4.3.

As we mentioned earlier, the steps above are used to update the hyperparameter values in parallel, obtaining $\gamma_i^{(k+1)}$ for the even indices. Then, these updated values are used in the optimization over the odd indices through the optimization steps analogous to (2.30), (2.36), and (2.33).

Algorithm 2: Universal EM-Based Segment-wise Parallel Update Algorithm for TV-SBL

Input: $\mathbf{A}, \mathbf{Y}, \boldsymbol{\gamma}^{(0)}$, λ, β

Output: $\boldsymbol{\gamma}^{(k_{\max})}$, $\boldsymbol{\mu}^{(k_{\max})}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k_{\max})}}$, $\forall l$

for $k = 0$ **to** k_{max} **do**
- Evaluate $\boldsymbol{\Sigma}^{(k)}_{\mathbf{x}|\mathbf{y}}$ and $\boldsymbol{\mu}^{(k)}_{\mathbf{x}_l|\mathbf{y}_l;\boldsymbol{\gamma}^{(k)}}$, $\forall l$ using (2.3);
- Evaluate $E_i^{(k)}$ using (2.22);
- Evaluate $\hat{\alpha}_i^{(k)}$, $\tilde{\alpha}_i^{(k)}$, and $\bar{\alpha}_i^{(k)}$, $\forall i$ using Table 2.2;
- **for** $i \in$ *even indices* **do**
  - Evaluate $\gamma^{(k)}_{i,\max}$ and $\gamma^{(k)}_{i,\min}$ using (2.30);
  - Evaluate $\hat{\gamma}_i^{(k)}$, $\tilde{\gamma}_i^{(k)}$, $\bar{\gamma}_i^{(k)}$ using (2.36);
  - Obtain $\gamma_i^{(k+1)}$ using (2.33);
- **for** $i \in$ *odd indices* **do**
  - Evaluate $\gamma^{(k+1)}_{i,\max}$ and $\gamma^{(k+1)}_{i,\min}$ using (2.30);
  - Evaluate $\hat{\gamma}_i^{(k)}$, $\tilde{\gamma}_i^{(k)}$, $\bar{\gamma}_i^{(k)}$ using (2.36);
  - Obtain $\gamma_i^{(k+1)}$ using (2.33);

2.4.3 Unifying TV-SBL using Alternating Optimization

We now specialize this framework and set up a *universal* TV-SBL solver to handle all the introduced TV regularizers: the linear TV (2.7), the log TV (2.8), and the DoL TV (2.11).

To begin, it is clear that for a convex cost (2.31), the segment-wise algorithm – following the steps (2.35), (2.36), and (2.33) – converges to a local minimum of (2.31), enjoying the convergence properties of Theorem 1, i.e., the TV-SBL M-step cost (2.25) decreases over each iteration. Since the DoL TV-SBL cost function (2.31) is non-convex, it is not immediately clear whether the unconstrained updates (2.35) in conjunction with the segment-wise projections in (2.36) work. To this end, Appendix 2.B shows the segment-wise parallel updates for the DoL TV cost to be equivalent to (2.31), i.e., this optimization strategy finds a local minimum of the M-step cost (2.25). Thus, the DoL TV-SBL optimization, via the controlled nature of the DoL TV, adheres to the convergence results of Theorem 1, *despite the non-convexity*. Appendix 2.B also derives the closed-form solutions to the unconstrained updates (2.35) for the DoL TV.

We derive an EM-based solution to the TV-SBL with the linear TV penalty (2.7) and the log TV penalty (2.8) as follows. For each penalty, we majorize the involved non-convex TV-SBL cost $F_i^{(k)}(\gamma_i; \gamma^{(k)}_{i,\max}, \gamma^{(k)}_{i,\min})$ in (2.31) by a convex surrogate. Applying the segment-wise algorithm on the majorized convex cost function

Table 2.2: TV-SBL: Unconstrained updates (2.35) for the linear, log, and DoL TV penalties

TV Penalty	$\hat{\alpha}_i^{(k)}$	$\tilde{\alpha}_i^{(k)}$	$\bar{\alpha}_i^{(k)}$
$\lvert\gamma_i - \gamma_{i-1}\rvert$	$\sqrt{\frac{E_i^{(k)}}{q_i^{(k)}+2\beta}}$	$\sqrt{\frac{E_i^{(k)}}{q_i^{(k)}}}$	$\sqrt{\frac{E_i^{(k)}}{\max(\epsilon_0, q_i^{(k)}-2\beta)}}$
$\log(\lvert\gamma_i - \gamma_{i-1}\rvert + \epsilon)$	$\sqrt{\frac{E_i^{(k)}}{q_i^{(k)}+\beta(a_i^{(k)}+b_i^{(k)})}}$	$\sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)}+\beta(b_i^{(k)}-a_i^{(k)})\right)}}$	$\sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)}-\beta(a_i^{(k)}+b_i^{(k)})\right)}}$
$\lvert\log(\gamma_i) - \log(\gamma_{i-1})\rvert$	$\frac{E_i^{(k)}}{1+2\beta}$	$E_i^{(k)}$	$\frac{E_i^{(k)}}{1-2\beta}$

ensures the convergence of (2.25) according to Corollary 1.1, i.e., the TV-SBL M-step cost (2.25) decreases over each iteration. The closed-form solutions to the unconstrained updates (2.35) for the linear and log TV under this majorization approach are derived in Appendix 2.C.1 and 2.C.2, respectively.

Finally, our **universal EM-based TV-SBL algorithm** exploiting the segment-wise parallel updating is summarized in Algorithm 12; the required unconstrained update rules for (2.35) for each TV-penalty type are summarized in Table 2.2.

2.4.4 Algorithm Complexity

Since the E-Step of TV-SBL is the same as the original M-SBL, the complexity of TV-SBL is similar to M-SBL. The main computational burden at each iteration of our proposed EM-based TV-SBL, as summarized in Algorithm 12, is the computation of the matrix inverse in the E-step in evaluating $\mathbf{\Sigma}_{\mathbf{x}_l|\mathbf{y}_l}^{(k)}$, $\forall l$, via (2.3). The number of floating-point operations is of order $\mathcal{O}(M^2N)$ per iteration k. It has been shown in [88] that the overall covariance computation can be done independent of the number of snapshots L. Thus, the overall order of the entire algorithm is $\mathcal{O}(k_{\max}M^2N)$, where $k_{\max}$ is the (maximum) number of iterations taken. For the M-step update (2.25) using the TV-SBL regularizers (2.7), (2.8), and (2.11), we have an $\mathcal{O}(1)$ parallel update procedure for each of the hyperparameter components $\boldsymbol{\gamma}_i$ (see Algorithm 12); thus faster computation than the E-step update (2.22), leaving the overall TV-SBL complexity as $\mathcal{O}(k_{\max}M^2N)$.

2.5 Numerical Results

We provide numerical results for the block-sparse signal recovery using the proposed TV-SBL algorithm, implemented via Algorithm 12. For the MMV setup (2.1), we consider a signal of length $N = 300$ with $M = 30$ measurements and $L = 5$ snapshots. We form the dictionary $\mathbf{A} \in \mathbb{R}^{M\times N}$ by first drawing its elements from a Gaussian distribution, and then normalizing the columns as $\|\cdot\|_2 = 1$. The signal ensemble $\mathbf{X}$ contains K nonzero rows and each nonzero element is drawn from $\mathcal{N}(0, 1/K)$. We consider three classes of sparse signal distributions:

1. *Homogeneous* block-sparse signal with total sparsity $K = 20$, composed of 4 blocks of length 5 each;

2. *Random* sparse signal with $K = 15$ randomly placed nonzero components, which are thus mostly isolated;

3. *Hybrid* sparse signal with total sparsity $K = 20$, composing 3 blocks of length 5, and 5 isolated components.

Each entry of noise signal $\mathbf{n}_l$ is generated from $\mathcal{N}(0, \lambda)$ with noise variance λ chosen so that the signal-to-noise ratio (SNR), $10\log_{10}\left(\frac{\mathbb{E}[||\mathbf{A}\mathbf{x}_l||^2]}{\mathbb{E}[||\mathbf{n}_l||^2]}\right)$, varies from -5 to 25 dB. All expectations $\mathbb{E}[\cdot]$ are evaluated over 200 Monte Carlo trials. We assess the performance with the following two metrics.

1. *Estimation accuracy* is measured by the normalized mean square error (NMSE), $\mathbb{E}\left[\frac{||\hat{\mathbf{X}}-\mathbf{X}||^2}{||\mathbf{X}||^2}\right]$, where $\hat{\mathbf{X}}$ is the estimated source matrix.

2. *Support recovery* is evaluated using the F_1-Score [119], $\mathrm{F}_1 = 2\,\mathbb{E}\left[\frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}\right]$, where precision $= \frac{\text{tp}}{\text{tp+fa}}$, recall $= \frac{\text{tp}}{\text{tp+mis}}$, tp is the number of true positives, fa is the number of false alarms, and mis is the number of misdetections. The F_1-Score is a balanced support recovery metric as it penalizes misdetections and false alarms equally. $\mathrm{F}_1 = 1$ implies perfect recovery.

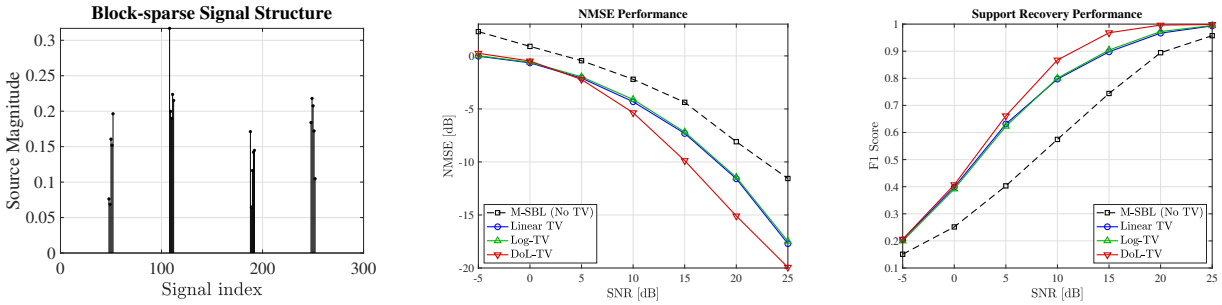
Remark. *For ease of comparison, we form a support estimate for an algorithm by preserving the K largest rows of $\hat{\mathbf{X}}$ while setting the rest to zero. In practice, the support could also be estimated using a fixed threshold. The choice for setting this fixed threshold is not static and we use empirical methods to set the best minimum threshold based on the signal values.*

Remark. *For implementation convenience, we first run the M-SBL algorithm [88] to convergence and use the obtained $\boldsymbol{\gamma}$ values as a warm start (i.e., initialize $\boldsymbol{\gamma}^{(0)}$) for all subsequent block-sparse recovery algorithms.*

2.5.1 Performance of Different TV Penalties

We first compare the relative performance of the different TV-based SBL regularizers (2.7), (2.8), and (2.11) for the homogeneous block-sparse scenario (see Fig. 2.1(a)). As evident in Figs. 2.1(b) and 2.1(c), all three TV-SBL regularizers outperform the M-SBL algorithm [88] which imposes no prior on $\boldsymbol{\gamma}$. Both conventional TV regularizers (linear TV and log TV) have very similar performance in terms of the NMSE and support recovery[3]. However, the region-aware DoL TV clearly outperforms the conventional TV

[3]As opposed to the EM-based approaches herein, our results in [85] show that the log TV considerably outperforms the linear TV when optimized via their MM approaches (2.14) and (2.17), respectively.



(a) (b) (c)

Figure 2.1: Performance of TV-SBL under the different TV penalties for $N = 300$, $M = 30$, and $L = 5$: (a) Homogeneous block-sparsity (4 blocks of length 5), (b) NMSE, and (c) Support recovery.

regularizers. This performance boost reinforces the strategy to selectively penalize signal and zero regions in block-sparse recovery.

Remark. *Based on the analysis in Sec. 2.5.1, we use the DoL TV version of the TV-SBL for all subsequent simulations.*

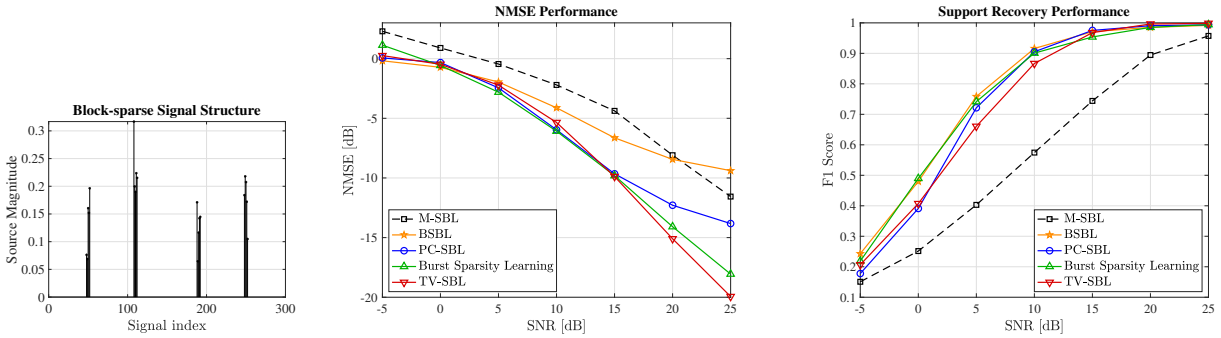
2.5.2 Comparison with Benchmark Algorithms

We investigate all the three signal classes and compare the performance of our TV-SBL algorithm (using DoL TV) against the following SBL-based block-sparse recovery algorithms: *(i)* BSBL [77], *(ii)* PC-SBL [79], and *(iii)* Burst Sparsity Learning [80]. The M-SBL algorithm [88] is used as a reference to show recovery without regularization.

Remark. *In order to assess the robustness of each algorithm to the changes in block patterns, the parameters of each algorithm were empirically tuned for the homogeneous block-sparse signal, and then left unchanged for random and hybrid sparse signals. This tuning is crucial for all hyperparameter-coupling based algorithms, including our TV-SBL, because it would be evidently optimal to disable the coupling (i.e., set $\beta = 0$ for TV-SBL) for random sparse signals. Thus, we simulate a practical signal recovery scenario where the parameter tuning might not be viable.*

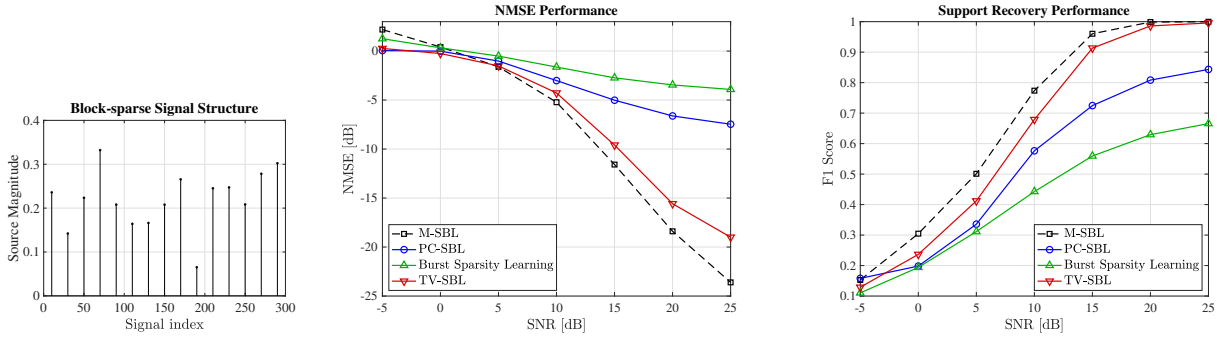
Homogeneous block-sparse signals

As seen in Fig. 2.2, all algorithms, unsurprisingly, outperform M-SBL. BSBL, being provided block size and boundary information *apriori*, attains the best F_1-Score (Fig. 2.2(c)). However, as the NMSE (Fig. 2.2(b)) illustrates, we need regularization/coupling in addition to fixed block partitioning to recover the signal effectively. TV-SBL shows superior recovery performance at high SNR values; however it is outperformed by



(a) (b) (c)

Figure 2.2: Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 30$, and $L = 5$.



(a) (b) (c)

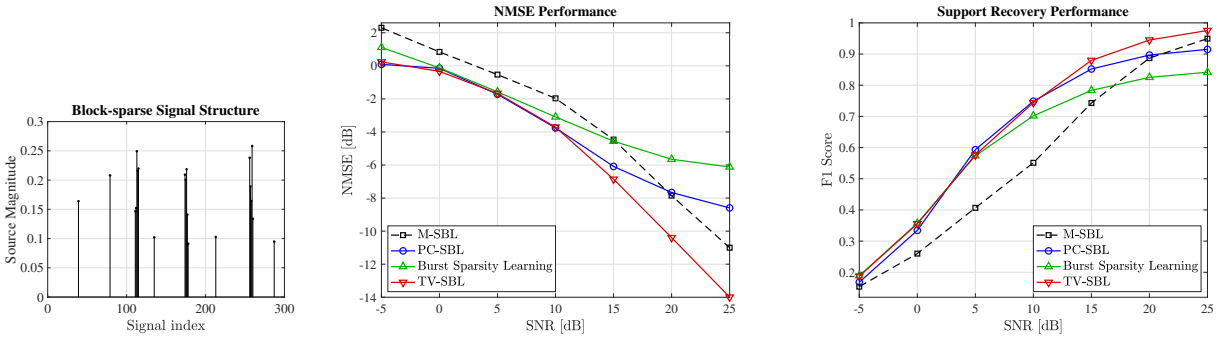
Figure 2.3: Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 30$, and $L = 5$: (a) Isolated sparsity (15 blocks of length 1), (b) NMSE, and (c) Support recovery.

Burst Sparsity Learning at lower SNR owing to the softer prior imposed. TV-SBL induces hyperparameter coupling implicitly through the DoL TV prior (2.11). This, combined with its region-aware nature, illustrates the utility of TV-SBL for block-sparse recovery.

There is an important caveat to explicit coupling-based priors like PC-SBL and Burst Sparsity Learning: *searching for blocks even when there are none.* We now demonstrate that a softer prior for block-sparsity in TV-SBL enjoys increased flexibility to a block structure, showing TV-SBL's utility beyond homogeneous block-sparsity.

Sparse signals

Fig. 2.3(a) represents the extreme scenario for the block-sparse algorithms: the block size is one. First, as seen in Figs. 2.3(b) and 2.3(c), M-SBL achieves the best performance, as expected, establishing a justified performance bound. TV-SBL significantly outperforms the coupling-based algorithms, while being comparable to M-SBL. Explicit hyperparameter coupling in PC-SBL and Burst Sparsity Learning biases the



(a) (b) (c)

Figure 2.4: Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 30$, and $L = 5$: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.

algorithms to block structures, and thus renders them ineffective for isolated sparsity. Using a softer prior, TV-SBL supports block-sparsity without such excessive bias; it is remarkably adept at isolated sparsity as well. This motivates us to analyze a hybrid signal, containing both isolated and block-patterned components, which is typically the case in practice.

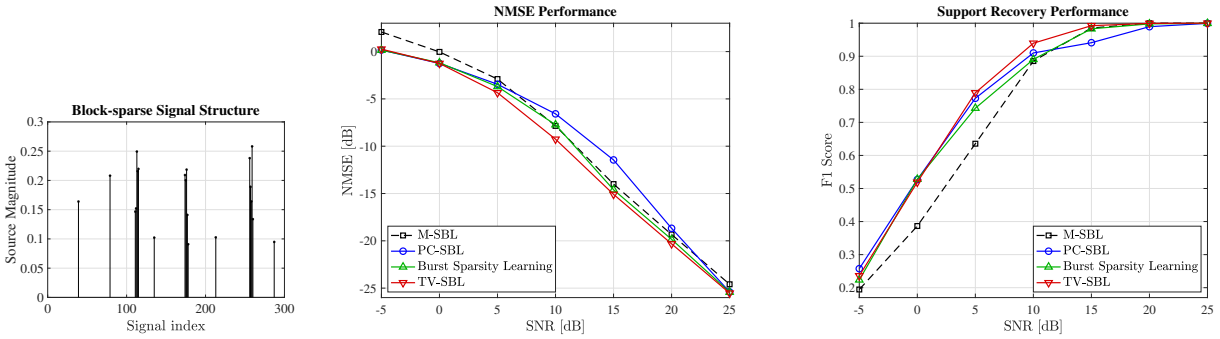
Hybrid sparse signals

The hybrid block structure in Fig. 2.4(a) is a representative of a practical scenario for, e.g., multiple-input multiple-output (MIMO) wireless channels. As seen in Figs. 2.4(b) and 2.4(c), TV-SBL outperforms all the other algorithms. The soft prior of TV-SBL flexibly accommodates both block-patterned and isolated components, whereas PC-SBL and Burst Sparsity Learning, imposing explicit hyperparameter coupling, are sensitive to isolated components in the underlying signal. This reinforces the fact that TV-SBL presents a *balanced* approach to block-sparse signal recovery; it does not compromise on sparse signal recovery at the cost of block-sparse signal recovery.

2.5.3 Effect of Compression Ratio

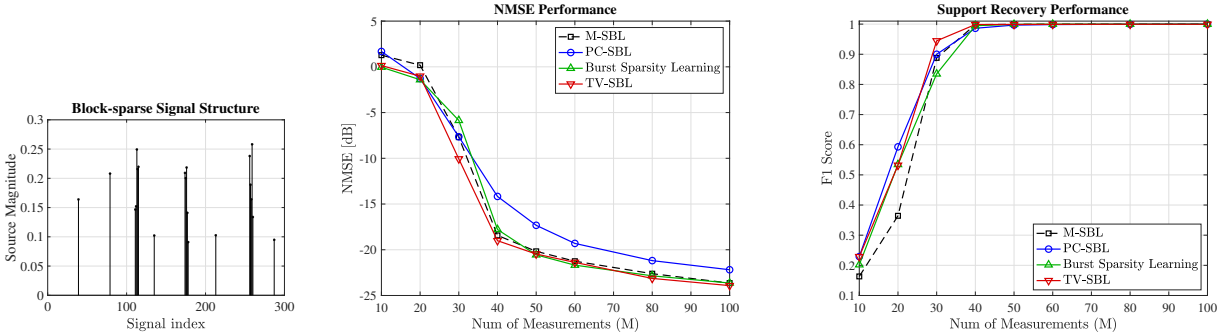
We highlight the effect of increasing the compression ratio M/N for the hybrid sparse signals in Fig. 2.5. We increase the number of measurements by 50 % from that in Fig. 2.4, i.e., from $M = 30$ to $M = 45$, for the SNR analysis. As observed in Figs. 2.5(b) and 2.5(c), the relative gap between the algorithms reduces. Here too, similar to Fig. 2.4, TV-SBL slightly outperforms the other algorithms. The improved performance for all algorithms is now attributed to the increase in the number of measurements, which reduces the effect of a block-sparsity prior. This is reinforced by analyzing the system performance for varying the number of measurements M at SNR 20 dB, as seen in Fig. 2.6. As seen from the results in Figs. 2.6(b) and

2.6(c), the performance saturates as we increase the number of measurements. This is in line with the general estimation theory in that the impact of a prior is overridden by having increased amount of data to perform the inference. However, it may not always be feasible to acquire more measurements in a fixed data acquisition setup. Nonetheless, for a scenario with a limited number of measurements and snapshots and varying block-sparse structures, our proposed TV-SBL algorithm represents a *robust* regularization-enforced SBL method for *general-patterned block-sparse signal recovery*.



(a) (b) (c)

Figure 2.5: Performance of TV-SBL (DoL TV) versus the benchmark algorithms for $N = 300$, $M = 45$, and $L = 5$: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.

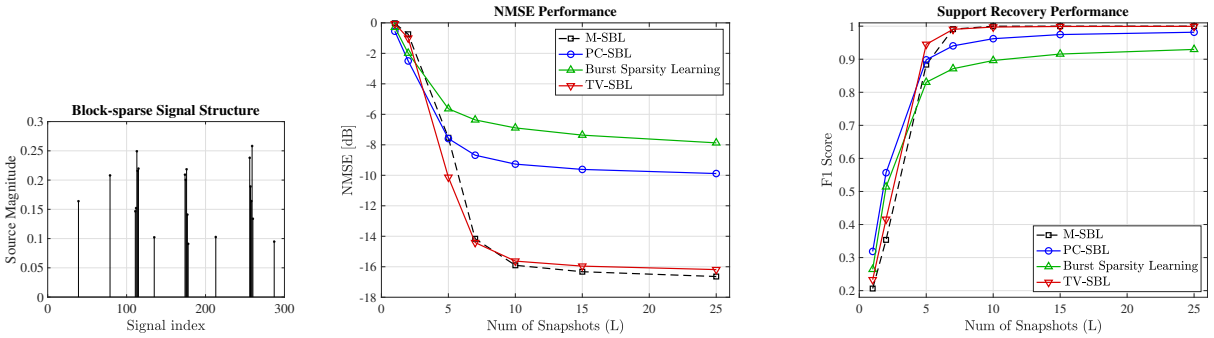


(a) (b) (c)

Figure 2.6: Performance of TV-SBL (DoL TV) versus the benchmark algorithms with varying number of measurements for $N = 300$, $L = 5$ at SNR $= 20$ dB: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.

2.5.4 Effect of Snapshots

We analyze the performance of the TV-SBL algorithm as we vary the number of snapshots L in Fig. 2.7. The NMSE (Fig. 2.7(a)) and support recovery (Fig. 2.7(b)) are evaluated at SNR 20 dB for the hybrid block-sparse signal structure (Fig. 2.7(a)). The performance plots illustrate that by increasing the number



(a) (b) (c)

Figure 2.7: Performance of TV-SBL (DoL TV) versus the benchmark algorithms with varying number of snapshots for $N = 300$, $M = 30$ at SNR $= 20$ dB: (a) Hybrid sparsity (3 blocks of length 5 and 5 blocks of length 1), (b) NMSE, and (c) Support recovery.

of snapshots, the performance of M-SBL improves significantly, as is expected from the reduced dependency on the prior. It is interesting to note that while the algorithms using rigid coupling, i.e., PC-SBL and Burst Sparsity Learning saturate at a lower NMSE, the softer prior of the TV-SBL is able to follow the M-SBL algorithm, thereby showing the flexibility of the chosen prior.

2.6 Conclusions and Future Work

We proposed TV-based hyperparameter regularizers for SBL to perform robust block-sparse signal recovery under unknown block patterns. We introduced a new SBL algorithm, TV-SBL, with two perspectives to handle block regularization: 1) the conventional TV regularization (linear TV and log TV) and 2) the RAR using TV (DoL TV). We showed that, after appropriate relaxations, the conventional TV regularization reduces to a sequence of convex optimization problems, enabling a multitude of numerical solvers. The majority of our analysis, algorithms, and experiments focused on the DoL TV regularizer, motivated by its superior performance reported herein. We developed an EM-based alternating optimization solution algorithm, which has universal applicability to all the introduced TV regularizers. The soft TV prior of the TV-SBL – especially when incorporating the region-aware DoL TV – presents a novel balanced perspective on handling block-sparsity. The numerical results show the TV-SBL algorithm to be an efficient method in recovering sparse signals with both block-patterned and isolated components, proving immensely useful for practical signal recovery systems, like mmWave channel estimation with non-uniform sparse scattering.

Chapter 2, in part, is a reprint of the material as it appears in Aditya Sant, Markus Leinonen, and Bhaskar D. Rao. "Block-sparse Signal Recovery via General Total Variation Regularized Sparse Bayesian Learning." IEEE Transactions on Signal Processing 70 (2022), and in part, a reprint of the material as

it appears in Aditya Sant, Markus Leinonen, and Bhaskar D. Rao. "General Total Variation Regularized Sparse Bayesian Learning for Robust Block-sparse Signal Recovery." ICASSP 2021-2021. The dissertation author was the primary investigator and author of these papers.

Appendices

2.A Proof of Theorem 1

Using (2.26) and (2.27), the right-hand side of (2.29), $J^{(k)}(\boldsymbol{\gamma}^{(k)})$, is written as

$$J^{(k)}(\boldsymbol{\gamma}^{(k)}) = \sum_{i\in\text{even}} F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i^{(k)}). \tag{2.37}$$

Given an initial value for $J^{(k)}(\boldsymbol{\gamma}^{(k)})$ in (2.37), by implementing the first step of the algorithm (2.28), i.e., updating the even elements in $\boldsymbol{\gamma}^{(k+1)}$, we get

$$\begin{aligned} &\sum_{i\in\text{even}} F_i^{(k)}(\gamma_i^{(k+1)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i^{(k)}) \\ &\leq J^{(k)}(\boldsymbol{\gamma}^{(k)}). \end{aligned} \tag{2.38}$$

The inequality (2.38) expresses the reduction in the cost function $J^{(k)}(\boldsymbol{\gamma})$ by only minimizing over the even indices of $\boldsymbol{\gamma}$. We now show the same for the odd indices, corresponding to the second step of the algorithm (2.28), finally leading to (2.29).

Let us rewrite the left-hand side of (2.38) equivalently as

$$\begin{aligned} &\sum_{i\in\text{even}} F_i^{(k)}(\gamma_i^{(k+1)};\gamma_{i-1}^{(k)},\gamma_{i+1}^{(k)}) + \sum_{i\in\text{odd}} f_i^{(k)}(\gamma_i^{(k)}) = \\ &\sum_{i\in\text{odd}} F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}) + \sum_{i\in\text{even}} f_i^{(k)}(\gamma_i^{(k+1)}). \end{aligned} \tag{2.39}$$

Since the second step of the algorithm (2.28) minimizes $\sum_{i\in\text{odd}} F_i^{(k)}(\gamma_i;\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)})$ over the odd coefficients, by using the equivalence in (2.39), the second step results in

$$\begin{aligned} J^{(k)}(\boldsymbol{\gamma}^{(k+1)}) \leq &\sum_{i\in\text{odd}} F_i^{(k)}(\gamma_i^{(k)};\gamma_{i-1}^{(k+1)},\gamma_{i+1}^{(k+1)}) \\ &+ \sum_{i\in\text{even}} f_i^{(k)}(\gamma_i^{(k+1)}), \end{aligned} \tag{2.40}$$

where $J^{(k)}(\boldsymbol{\gamma}^{(k+1)})$ represents the M-step cost function value after completing one iteration cycle in the EM algorithm.

Combining (2.38), (2.39), and (2.40) results in $J^{(k)}(\boldsymbol{\gamma}^{(k+1)}) \leq J^{(k)}(\boldsymbol{\gamma}^{(k)})$ in (2.29), which completes the proof.

2.B Segment-wise Optimization for DoL TV Penalty

Using the definition of $f_i^{(k)}(\gamma_i)$ in (2.24), we rewrite (2.35) as

$$\begin{aligned}
\hat{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} F_1^{(k)}(\gamma_i) \triangleq \tfrac{E_i^{(k)}}{\gamma_i} + (1+2\beta)\log\gamma_i, \\
\tilde{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} F_0^{(k)}(\gamma_i) \triangleq \tfrac{E_i^{(k)}}{\gamma_i} + \log\gamma_i, \\
\bar{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} F_2^{(k)}(\gamma_i) \triangleq \tfrac{E_i^{(k)}}{\gamma_i} + (1-2\beta)\log\gamma_i.
\end{aligned} \tag{2.41}$$

These can be solved by equating the derivative to zero, generating the unconstrained update expressions as

$$\hat{\alpha}_i^{(k)} = \frac{E_i^{(k)}}{1+2\beta}, \quad \tilde{\alpha}_i^{(k)} = E_i^{(k)}, \quad \bar{\alpha}_i^{(k)} = \frac{E_i^{(k)}}{1-2\beta}. \tag{2.42}$$

We choose the regularization parameter $\beta < 0.5$ to ensure positive minima. Next, we show the validity of the two-step strategy in (2.35) and (2.36), separately for each three segments $\hat{\Gamma}_i$, $\bar{\Gamma}_i$, and $\tilde{\Gamma}_i$ given by (2.32).

Segment $\gamma_i \in \hat{\Gamma}_i$: The derivative of the cost function $F_1^{(k)}(\gamma_i)$ in (2.41) is

$$\frac{\partial}{\partial\gamma_i} F_1^{(k)}(\gamma_i) = -\frac{E_i^{(k)}}{\gamma_i^2} + (1+2\beta)\frac{1}{\gamma_i}. \tag{2.43}$$

It is evident that $\frac{\partial F_1^{(k)}(\gamma_i)}{\partial\gamma_i} > 0$ for $\gamma_i > \hat{\alpha}_i^{(k)}$, where $\hat{\alpha}_i^{(k)}$ is defined in (2.42). Additionally, if $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$, then, by the definition of segment $\hat{\Gamma}_i$ in (2.32), we must have $\hat{\alpha}_i^{(k)} < \gamma_{i,\max}^{(k)}$. This implies that if $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$, then $\hat{\alpha}_i^{(k)} < \gamma_i$, $\forall \gamma_i \in \hat{\Gamma}_i$. Using the fact that (2.43) is positive for $\gamma_i > \hat{\alpha}_i^{(k)}$, $F_1^{(k)}(\gamma_i)$ is continuously increasing in the segment $\hat{\Gamma}_i$, if $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$. Thus, the minimum of $F_1^{(k)}(\gamma_i)$ under the constraint set $\gamma_i \in \hat{\Gamma}_i$ is located either at the stationary point $\hat{\alpha}_i^{(k)}$ (when $\hat{\alpha}_i^{(k)} \in \hat{\Gamma}_i$) or at the lowest value of the segment (when $\hat{\alpha}_i^{(k)} \notin \hat{\Gamma}_i$). This leads to the solution of the form (2.36) as

$$\operatorname*{argmin}_{\gamma_i \in \hat{\Gamma}_i} F_1^{(k)}(\gamma_i) = \max\big(\gamma_{i,\max}^{(k)}, \hat{\alpha}_i^{(k)}\big). \tag{2.44}$$

Segment $\gamma_i \in \bar{\Gamma}_i$: The derivative of the cost function $F_2^{(k)}(\gamma_i)$ in (2.41) is

$$\frac{\partial}{\partial \gamma_i} F_2^{(k)}(\gamma_i) = -\frac{E_i^{(k)}}{\gamma_i^2} + (1-2\beta)\frac{1}{\gamma_i}. \tag{2.45}$$

It is evident that $\frac{\partial F_2^{(k)}(\gamma_i)}{\partial \gamma_i} < 0$ for $0 < \gamma_i < \bar{\alpha}_i^{(k)}$, where $\bar{\alpha}_i^{(k)}$ is defined in (2.42). By the definition of segment $\bar{\Gamma}_i$ in (2.32), if $\bar{\alpha}_i^{(k)} \notin \bar{\Gamma}_i$, then $\bar{\alpha}_i^{(k)} > \gamma_{i,\min}^{(k)}$; this implies $\bar{\alpha}_i^{(k)} > \gamma_i$, $\forall \gamma_i \in \bar{\Gamma}_i$. Using the fact that (2.45) is negative for $0 < \gamma_i < \bar{\alpha}_i^{(k)}$, $F_2^{(k)}(\gamma_i)$ is continuously decreasing in the segment $\bar{\Gamma}_i$ if $\bar{\alpha}_i^{(k)} \notin \bar{\Gamma}_i$. Following the logic similar to segment $\hat{\Gamma}_i$, the solution is of the form (2.36), i.e.,

$$\underset{\gamma_i \in \bar{\Gamma}_i}{\operatorname{argmin}}\ F_2^{(k)}(\gamma_i) = \max\big(0,\, \min\big\{\gamma_{i,\min}^{(k)},\, \bar{\alpha}_i^{(k)}\big\}\big) \tag{2.46}$$

Segment $\gamma_i \in \tilde{\Gamma}_i$: The derivative of the cost function $F_0^{(k)}(\gamma_i)$ in (2.41) is

$$\frac{\partial}{\partial \gamma_i} F_0^{(k)}(\gamma_i) = -\frac{E_i^{(k)}}{\gamma_i^2} + \frac{1}{\gamma_i}. \tag{2.47}$$

It is evident that *(i)* $\frac{\partial F_0^{(k)}(\gamma_i)}{\partial \gamma_i} < 0$ for $0 < \gamma_i < \tilde{\alpha}_i^{(k)}$, and *(ii)* $\frac{\partial F_0^{(k)}(\gamma_i)}{\partial \gamma_i} > 0$ for $\gamma_i > \tilde{\alpha}_i^{(k)}$, where $\tilde{\alpha}_i^{(k)}$ is defined in (2.42). By the definition of segment $\tilde{\Gamma}_i$ in (2.32), if $\tilde{\alpha}_i^{(k)} \notin \tilde{\Gamma}_i$, then either $\tilde{\alpha}_i^{(k)} < \gamma_{i,\min}^{(k)}$ or $\tilde{\alpha}_i^{(k)} > \gamma_{i,\max}^{(k)}$. If $\tilde{\alpha}_i^{(k)} < \gamma_{i,\min}^{(k)}$, then $F_0^{(k)}(\gamma_i)$ is increasing in $\tilde{\Gamma}_i$; in this case, $\tilde{\gamma}_i^{(k)} = \gamma_{i,\min}^{(k)}$. Analogously, if $\tilde{\alpha}_i^{(k)} > \gamma_{i,\max}^{(k)}$, then $\tilde{\gamma}_i^{(k)} = \gamma_{i,\max}^{(k)}$. Putting these conditions together, we obtain the solution in (2.36) as

$$\underset{\gamma_i \in \tilde{\Gamma}_i}{\operatorname{argmin}}\ F_0^{(k)}(\gamma_i) = \min\Big(\gamma_{i,\max}^{(k)},\, \max\Big\{\gamma_{i,\min}^{(k)},\, \tilde{\alpha}_i^{(k)}\Big\}\Big). \tag{2.48}$$

2.C Expression for unconstrained updates

2.C.1 Unconstrained Updates in (2.35) for Linear TV

We begin by reformulating the M-step (2.25) to correspond to the linear TV penalty in (2.7) as

$$J^{(k)}(\boldsymbol{\gamma}) = \textstyle\sum_{i=1}^{N} \left[\frac{E_i^{(k)}}{\gamma_i} + \log \gamma_i\right] + \beta \sum_{i=2}^{N} |\gamma_i - \gamma_{i-1}|. \tag{2.49}$$

We majorize $\log \gamma_i$ by its first-order Taylor approximation at point $\gamma_i^{(k)}$, $i = 1, \ldots, N$. Consequently, (2.49) is majorized as

$$\hat{J}^{(k)}(\boldsymbol{\gamma}) = \textstyle\sum_{i=1}^{N} \left[\frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i\right] + \beta \sum_{i=2}^{N} |\gamma_i - \gamma_{i-1}|, \tag{2.50}$$

where $q_i^{(k)} = \frac{1}{\gamma_i^{(k)}+\epsilon}$ and ϵ is a small stability parameter. Following the steps similar to Sec. 2.4.1 and 2.4.2, we obtain the majorized cost function equivalent to (2.31) as

$$\begin{aligned}\hat{F}_i^{(k)}(\gamma_i;\gamma_{i,\max}^{(k)},\gamma_{i,\min}^{(k)}) &= \frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i \\ &+\beta\ \left(\left|\gamma_i-\gamma_{i,\max}^{(k)}\right| + \left|\gamma_i-\gamma_{i,\min}^{(k)}\right|\ \right),\end{aligned} \tag{2.51}$$

Accordingly, using the definition of $f_i^{(k)}(\gamma_i)$ in (2.24), we write the updates in (2.35) for linear TV as

$$\begin{aligned}\hat{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\ \tfrac{E_i^{(k)}}{\gamma_i} + (q_i^{(k)}+2\beta)\gamma_i \\ \tilde{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\ \tfrac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i \\ \bar{\alpha}_i^{(k)} &= \text{argmin}_{\gamma_i\in\mathbb{R}}\ \tfrac{E_i^{(k)}}{\gamma_i} + (q_i^{(k)}-2\beta)\gamma_i,\end{aligned} \tag{2.52}$$

For the first two problems in (2.52), differentiating with respect to γ_i and equating to zero, we get $\hat{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{q_i^{(k)}+2\beta}}$ and $\tilde{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{q_i^{(k)}}}$. For the third problem in (2.52), it is possible that $q_i^{(k)} - 2\beta < 0$, i.e., the cost function is unbounded below and theoretically, $\bar{\alpha}_i^{(k)} = \infty$. In a practical implementation, we account for the negativity $q_i^{(k)}-2\beta < 0$ via setting a large value $\bar{\alpha}_i^{(k)}$. Thus, the update rule becomes $\bar{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{\max(\epsilon_0, q_i^{(k)}-2\beta)}}$, where ϵ_0 is a small number.

2.C.2 Unconstrained Updates in (2.35) for Log TV

We begin by formulating the M-step (2.25) with respect to the log TV penalty in (2.8) as

$$J^{(k)}(\boldsymbol{\gamma}) = \textstyle\sum_{i=1}^N\left[\frac{E_i^{(k)}}{\gamma_i} + \log\gamma_i\right] + \beta\sum_{i=2}^N \log(|\gamma_i-\gamma_{i-1}|+\epsilon_1). \tag{2.53}$$

Besides majorizing $\log\gamma_i$ similar to the linear TV case in Appendix 2.C.1, we majorize $\log(|\gamma_i-\gamma_{i-1}|+\epsilon_1)$ by its first-order Taylor approximation at point $(\gamma_i^{(k)}-\gamma_{i-1}^{(k)})$, $i = 2,\ldots,N$. Consequently, (2.53) is majorized as

$$\tilde{J}^{(k)}(\boldsymbol{\gamma}) = \textstyle\sum_{i=1}^N\left[\frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i\right] + \beta\sum_{i=2}^N c_i^{(k)}|\gamma_i-\gamma_{i-1}|, \tag{2.54}$$

where $q_i^{(k)} = \frac{1}{\gamma_i^{(k)}+\epsilon}$ and $c_i^{(k)} = \frac{1}{|\gamma_i^{(k)}-\gamma_{i-1}^{(k)}|+\epsilon_1}$.

Following the steps similar to Sec. 2.4.1 and 2.4.2, we obtain the majorized cost function having the form similar to (2.31):

$$\begin{aligned}\tilde{F}_i^{(k)}(\gamma_i;\gamma_{i,\max}^{(k)},\gamma_{i,\min}^{(k)}) &= \frac{E_i^{(k)}}{\gamma_i} + q_i^{(k)}\gamma_i \\ &+\beta\ \left(\ a_i^{(k)}\left|\gamma_i-\gamma_{i,\max}^{(k)}\right| + b_i^{(k)}\left|\gamma_i-\gamma_{i,\min}^{(k)}\right|\ \right),\end{aligned} \tag{2.55}$$

where $a_i^{(k)} = \frac{1}{|\gamma_i^{(k)} - \gamma_{\max}^{(k)}| + \epsilon_1}$ and $b_i^{(k)} = \frac{1}{|\gamma_i^{(k)} - \gamma_{\min}^{(k)}| + \epsilon_1}$. Observing the similarity to (2.51), the unconstrained updates (2.35) for log TV become

$$\begin{aligned}
\hat{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} \frac{E_i^{(k)}}{\gamma_i} + \big(q_i^{(k)} + \beta(a_i^{(k)} + b_i^{(k)})\big)\gamma_i \\
\tilde{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} \frac{E_i^{(k)}}{\gamma_i} + \big(q_i^{(k)} + \beta(b_i^{(k)} - a_i^{(k)})\big)\gamma_i \\
\bar{\alpha}_i^{(k)} &= \operatorname{argmin}_{\gamma_i \in \mathbb{R}} \frac{E_i^{(k)}}{\gamma_i} + \big(q_i^{(k)} - \beta(a_i^{(k)} + b_i^{(k)})\big)\gamma_i.
\end{aligned} \tag{2.56}$$

Based on the same logic as for the linear TV updates, the solutions to these unconstrained updates are $\hat{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{q_i^{(k)} + \beta(a_i^{(k)} + b_i^{(k)})}}$, $\tilde{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)} + \beta(b_i^{(k)} - a_i^{(k)})\right)}}$, and $\bar{\alpha}_i^{(k)} = \sqrt{\frac{E_i^{(k)}}{\max\left(\epsilon_0, q_i^{(k)} - \beta(a_i^{(k)} + b_i^{(k)})\right)}}$.

Chapter 3

Insights into One-bit MIMO Detection

3.1 Introduction

The advent of massive MIMO communications has brought in a new era of high speed communication systems and interconnected devices [3, 4]. However, one of the key challenges facing massive MIMO deployment is the ensuing system cost and complexity. In this context, the use of high-resolution and high-speed Analog-to-Digital Converters (ADCs) significantly contributes to the overall cost and power consumption within the MIMO communication system [23, 120]. Addressing these challenges, accompanied by several advances in algorithm design and machine learning techniques, research into robust communication system design is being developed on the backbone of few-bit or low resolution ADCs [20–25]. A specific type of low-resolution ADC, the one-bit ADC, has attracted notable attention in communication system design and sensing due to its simple design and ease of implementation.

Advances in DNN technologies have enabled robust detector designs for these few-bit MIMO receivers. The overall application of DNNs to wireless communication systems has greatly improved receiver performance and robustness. The general parametric structure of DNNs, coupled with their advantage as universal functional approximators [121, 122], makes these an integral part of the future of robust wireless communication, exploited for a variety of applications from beamformer design [52–54], channel estimation [123–125] as well as end-to-end detection [55–59]. In this work, we begin by analyzing the detection process for one-bit MIMO receivers. Following this, a robust detector utilizing a DNN-aided unfolded network is

developed for multi-user one-bit massive MIMO systems.

3.1.1 Prior work

One-bit MIMO was first used for sensing and channel estimation algorithms [26–28]. Going beyond this, the main focus of one-bit MIMO has been on receiver design. One-bit MIMO data detection gained a valuable advance with the application of Bussgang's theorem to linearize the input-output relation [126]. Through means of this relation, a class of linear receivers was developed for detection from one-bit data [29–31]. Several works utilized this linearization to characterize the one-bit system and evaluate the overall system performance and capacity [127–129]. Additional robust model-based detectors improving on the Bussgang linear detectors have also been proposed in some works [32, 33].

However, today, the state-of-the-art class of receivers utilizes the nonlinear optimization of the likelihood function. The one-bit maximum likelihood (ML) optimization was derived using the Gaussian cumulative distribution function (CDF) [34]. Utilizing this formulation the work in [35] introduced a near maximum likelihood (n-ML) detector based on a two step iterative algorithm - gradient descent (GD) followed by projection onto the unit sphere. Other works applying the Gaussian CDF likelihood formulation have also been used extending this idea [130, 131]. However, one of the limitations of applying the GD iteration on the Gaussian CDF is its numerical instability at high signal-to-noise ratio (SNR) values [132]. One of the approaches to address this was through a surrogate function for the Gaussian CDF, i.e., the logistic regression (LR). The authors in [1] designed the detector, the OBMNet, as an unfolded DNN, implementing the GD algorithm for this approximate likelihood. Both the n-ML algorithm as well as the OBMNet were limited in performance due to the sub-optimal projection step onto the M-QAM constellation. The work in [20] improved on the OBMNet by introducing a learnable M-QAM projection over the GD iterations. The resulting unfolded DNN, the FBM-DetNet, is the current state-of-the-art detector for one-bit MIMO systems.

Extending the model-based methods to learning-based methods, DNNs have also been used to design robust detectors for one-bit MIMO receivers. The OBMNet [1] and FBM-DetNet[20] were implemented as unfolded DNNs, learning the parameters for the GD methods and the M-QAM projection, respectively. The work in [133] utilized model-based unrolling to learn the GD algorithm with a generalized Newton update. Other DNNs for one-bit detection, not relying on the likelihood framework have also been developed [134–137]. The general parametric structure of DNNs can also enhance the GD update step by enforcing a general regularization at the end of each GD iteration. The framework in [36] utilizes two unique networks

- an unfolded DNN, the ROBNet, as well as a recurrent network, the OBiRIM, to implement a regularized GD algorithm. The mmWave extension of the regularized GD, i.e., the mmW-ROBNet [138], demonstrates the utility of the regularized GD framework for mmWave channels. Here, the regularized framework, along with a novel hierarchical training strategy is able to generate equitable user performance for the mmWave one-bit MIMO receiver.

Although different strategies for detection of one-bit received signals have been proposed, no work, to the best of the authors' knowledge, comprehensively looks at the properties and convergence for the recovery algorithms. Bridging this gap, this work aims to generate useful insights into the ML framework for the one-bit MIMO receivers.

3.1.2 Contributions of this work

Through this work, we endeavor to bridge the gap between theory and algorithm design for the one-bit MIMO receiver. In particular, we enumerate the following contributions.

1. *Characterizing ML optimization:* We characterize the properties of the CDF-based likelihood, namely, the convexity, smoothness and the nature of the solution space. Different from prior works, this analysis enables structured algorithm design as well as convergence analysis.

2. *Stabilizing CDF-based GD update:* Implementing the GD update for the CDF-based likelihood is shown to run into computational instabilities. Utilizing the properties of the CDF, a robust approximation of the gradient is implemented, preserving the first order properties of the CDF (necessary for GD).

3. *Introduce accelerated GD update:* This stabilized GD update is utilized in the design of an accelerated GD algorithm for faster convergence. To the best of the authors' knowledge, this is the first work to utilize AGD in signal recovery for one-bit MIMO receivers. The convergence of the algorithms is analyzed using the properties of the likelihood function.

4. *Analysis of robust CDF surrogate, LR:* Prior works have demonstrated the utility of the logistic regression (LR) as an effective surrogate to the CDF for the one-bit likelihood [1, 20, 132]. The insights from the CDF-based likelihood are extended to explaining the improved performance of the LR-based likelihood.

5. *DNN-aided Gaussian denoising:* In order to address the constrained optimization over the M-QAM symbols, we extend and generalize the quantization-based M-QAM projection from [20]. To this end, we expound the role of the M-QAM projection step and develop a general *learnable* two-tier projection

framework for robust M-QAM symbol recovery. This framework is implemented as an unfolded DNN referred to as the A-PrOBNet.

Organization: This manuscript is organized as follows - Sec. 3.2 introduces the system model, and formulation of the one-bit likelihood optimization. Sec. 3.3 analyzes the different properties of the CDF-based likelihood. This section also introduces the improved GD algorithm and the AGD algorithm, as well as the related convergence analysis for these algorithms. Sec. 3.4 analyzes the surrogate functions for the CDF-based likelihood, in particular, the LR function. Sec. 3.5 introduces the general Gaussian denoising for projection onto the M-QAM symbol space. Sec. 4.6 provides the experimental validation and Sec. 5.6 provides concluding remarks and future directions.

Notation: The abbreviation ML is used for maximum likelihood, as opposed to machine learning. The latter has not been abbreviated wherever utilized. We use lower-case boldface letters $\mathbf{a}$ and upper case boldface letters $\mathbf{A}$ to denote complex valued vectors and matrices respectively. The notation $\mathfrak{Re}(\cdot)$ and $\mathfrak{Im}(\cdot)$ denote the real and imaginary parts, respectively. The operation $(\cdot)^{\mathrm{T}}$ denotes the transpose of the array or matrix. Unless otherwise specified, all scalar functions like $\tanh(\cdot)$ or $\mathrm{sign}(\cdot)$, when applied to arrays or matrices, imply element-wise operation. The diagonalization operator, denoted by $\mathrm{diag}(\cdot)$, when applied to an array $\mathbf{a}$, creates a diagonal matrix with the diagonal entries given by $\mathbf{a}$. The notation $\mathbf{x}^{(t)}$ is used to denote the value of the variable $\mathbf{x}$ at iteration t of the algorithm. For the DNN training, the size of the training set is given by N_{train} and the notation $\hat{\mathbf{x}}_{n,\text{train}}$ denotes the n^{th} sample from this set. Unless otherwise specified, the norm $||\cdot||$ represents the ℓ_2-norm for a vector or matrix.

3.2 System Model and general one-bit likelihood

Through this section the multi-user uplink MIMO model is introduced, with one-bit ADCs at the base station (BS) receiver. This is followed by the formulation of the ML optimization that forms the basis of the detection algorithm.

3.2.1 One-bit MIMO system model

The Rayleigh fading channel with block flat-fading, as used in most past works, e.g. [12, 139] is utilized here. The K single antenna users transmit to a multi-antenna base-station (BS) with N receive antennas. The MIMO channel $\bar{\mathbf{H}} \in \mathbb{C}^{N\times K}$ consists of i.i.d entries drawn from $\mathcal{CN}(0,1)$. This work assumes perfect unquantized channel state information (CSI) at the BS.

As a part of the multi-user uplink, the k^{th} user transmits the signal $\bar{x}_k$ drawn from the M-QAM constellation. The multi-user transmitted signal is $\bar{\mathbf{x}} = \begin{bmatrix} \bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K \end{bmatrix}^{\mathrm{T}}$. The unquantized received signal at the BS is given by

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

where $\bar{\mathbf{n}}$ is the AWCGN with noise variance depending on the system signal-to-noise ratio (SNR) $\rho = \frac{\mathbb{E}(||\bar{\mathbf{H}}\bar{\mathbf{x}}||^2)}{\mathbb{E}(||\bar{\mathbf{n}}||^2)}$. The transformed signal due to the one-bit quantization is given by

$$\bar{\mathbf{y}} = \operatorname{sign}\big(\mathfrak{Re}(\bar{\mathbf{r}})\big) + j \operatorname{sign}\big(\mathfrak{Im}(\bar{\mathbf{r}})\big). \tag{3.2}$$

In order to express the algorithm design as a function of real-valued inputs, we convert the received signal and the observed channel matrix into real-valued forms as

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{H}}) & -\mathfrak{Im}(\bar{\mathbf{H}}) \\ \mathfrak{Im}(\bar{\mathbf{H}}) & \mathfrak{Re}(\bar{\mathbf{H}}) \end{bmatrix}, \ \mathbf{x} = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{x}}) \\ \mathfrak{Im}(\bar{\mathbf{x}}) \end{bmatrix}, \\ \mathbf{r} &= \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{r}}) \\ \mathfrak{Im}(\bar{\mathbf{r}}) \end{bmatrix}, \ \mathbf{y} = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{y}}) \\ \mathfrak{Im}(\bar{\mathbf{y}}) \end{bmatrix}, \ \mathbf{n} = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{n}}) \\ \mathfrak{Im}(\bar{\mathbf{n}}) \end{bmatrix}. \end{aligned} \tag{3.3}$$

Thus, the modified received one-bit signal at the BS is

$$\mathbf{y} = \operatorname{sign}(\mathbf{H}\mathbf{x} + \mathbf{n}). \tag{3.4}$$

The detection algorithm recovers the M-QAM transmitted symbols $\mathbf{x}$ from the one-bit received data $\mathbf{y}$.

3.2.2 Signal detection - Maximum likelihood framework

As stated earlier, the signal detection is formulated as a likelihood optimization problem. The maximum likelihood (ML) problem for one-bit MIMO has been derived in [34]. We alternatively opt for minimizing the negative log-likelihood expression for ease of subsequent analysis. This is given by

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \underset{\mathbf{x} \in \mathcal{M}^{2K}}{\operatorname{argmin}} \sum_{i=1}^{2N} -\log \Phi\big(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}\big), \tag{3.5}$$

where $\Phi(\cdot)$ is the Gaussian cumulative distribution function (CDF) for $\mathcal{N}(0,1)$ and $\mathcal{M}^{2K}$ represents the set of the $2K$-dimensional vectors, consisting of the real-valued representation (see eq. (3.3)) of the K-dimensional

vectors of M-QAM symbols. In addition, the vector $\mathbf{h}_i$ denotes the i^{th} row of $\mathbf{H}$.

Remark. *Since the factor $\sqrt{2\rho}$ is a positive scalar and does not affect the convergence of the optimization over the constrained set $\mathcal{M}^{2K}$, we can eliminate this factor for ease of representation. Thus, for all subsequent expressions and analysis, the likelihood is expressed as a general function by the form $\mathcal{L} = \sum_i f(y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x})$*

In order to delve deeper into the analysis of the likelihood function, and algorithm development, we consider two key features with respect to the optimization (3.5).

Generalization of likelihood

In order to understand the broader class of likelihood functions, including all surrogate measures, a general likelihood formulation is presented as

$$\mathcal{L}(\mathbf{x}) = \sum_{i=1}^{2N} \zeta(y_i \mathbf{h}^{\mathrm{T}} \mathbf{x}). \tag{3.6}$$

The scalar function $\zeta(\cdot)$ can take different values, depending on the exact or surrogate value of the likelihood. Based on this, we follow two separate lines of analysis

- By substituting the CDF, we attain the original likelihood expression (3.5). We provide detailed analysis into the CDF-based likelihood expression in Sec. 3.3.
- We can also substitute appropriate surrogates for the CDF-based likelihood to overcome the limitations of the former. This is elaborated in more detail in Sec. 3.4.

The general gradient $\nabla_{\mathbf{x}}$ and Hessian $\mathcal{H}_{\mathbf{x}}$ expressions will be utilized in the analysis later. For the general likelihood $\zeta(\cdot)$, these expressions are given as

$$\nabla_{\mathbf{x}} = \mathbf{G}^{\mathrm{T}} \, \zeta'(\mathbf{G}\mathbf{x}) \tag{3.7a}$$

$$\mathcal{H}_{\mathbf{x}} = \mathbf{H}^{\mathrm{T}} \, \mathrm{diag}(\zeta''(\mathbf{G}\mathbf{x})) \, \mathbf{H}, \tag{3.7b}$$

where $\mathbf{G} = \mathrm{diag}(\mathbf{y})\mathbf{H}$ and the $\mathrm{diag}(\cdot)$ operator notation for both matrices and arrays is explained in Sec. 5.1.

Constrained vs unconstrained optimization

Since the transmitted symbols are drawn from an M-QAM constellation, a constrained optimization is performed over the set of M-QAM symbols. However, for understanding the properties of the likelihood

framework and development of robust recovery algorithms, unconstrained optimization over the entire set $\mathbb{R}^{2K}$ is initially considered. Specifically, we analyze

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \underset{\mathbf{x}\in\mathbb{R}^{2K}}{\operatorname{argmin}} \sum_{i=1}^{2N} -\log \Phi\big(y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}\big). \tag{3.8}$$

The CDF-based likelihood and the different CDF surrogates will first be analyzed via the unconstrained optimization framework (3.8) in Sec. 3.3-3.4. Constrained optimization over $\mathcal{M}^{2K}$ is then detailed in Sec. 3.5.

3.3 Insights into the cdf-based one-bit likelihood

This section begins with the analysis of the CDF-based likelihood. This is followed by the design a robust approximate GD algorithm and accelerated GD algorithm, along with the convergence analysis for both.

Substituting the CDF-based likelihood for the general expressions (3.6)-(3.7) gives

$$\zeta(z) = -\log \Phi(z) \tag{3.9a}$$

$$\zeta'(z) = -\frac{\phi(z)}{\Phi(z)} \tag{3.9b}$$

$$\zeta''(z) = \frac{\phi(z)}{\Phi(z)}\Big(z + \frac{\phi(z)}{\Phi(z)}\Big), \tag{3.9c}$$

where $\phi(\cdot)$ is the probability density function (PDF) of the standard normal distribution $\mathcal{N}(0,1)$. Utilizing this in (3.6) and evaluating the gradient gives the GD update, as derived in [35],

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \alpha^{(t)}\, \mathbf{G}^{\mathrm{T}} \frac{\phi(\mathbf{G}\mathbf{x})}{\Phi(\mathbf{G}\mathbf{x})}, \tag{3.10}$$

where $\alpha^{(t)}$ is the step size at the t^{th} iteration.

One of the limitations of applying unconstrained GD to the CDF-based likelihood function is the evaluation of the gradient (3.9b) (see Sec. 3.3.2 for details). We construct a more robust GD algorithm to overcome these limitations.

3.3.1 Characterizing the CDF-based likelihood

The various properties of the CDF-based one-bit likelihood function, useful for deriving the different GD-based algorithms and analyzing the convergence properties, are enumerated. The following inequalities are used in this analysis.

Lemma 1. *For the scalar argument $z < 0$, we can bound the CDF-gradient $\zeta'(z)$, given by* (3.9b)*, by the following bounds*

$$-z < \frac{\phi(z)}{\Phi(z)} < -z - \frac{1}{z} \tag{3.11}$$

Proof. If $z > 0$, then the following holds for the CDF,

$$\frac{1}{\sqrt{2\pi}}\exp\Big(-\frac{z^2}{2}\Big)\frac{z}{z^2+1} < 1 - \Phi(z) < \frac{1}{\sqrt{2\pi}}\exp\Big(-\frac{z^2}{2}\Big)\frac{1}{z}. \tag{3.12}$$

Rearranging the terms above gives the inequalities in (3.11). □

We now enumerate the various characteristic properties of the cdf-based likelihood function.

Convexity

The Hessian for the CDF-based likelihood (3.9c) is substituted in (3.7b). This can be separately analyzed for both positive and negative arguments. For $z > 0$ the expression (3.7b) is always positive. For $z < 0$, lemma 1 is utilized to show

$$0 < \zeta''(z) < 1. \tag{3.13}$$

Since each element of the matrix $\mathrm{diag}(\zeta''(\mathbf{Gx}))$ in (3.7b) is always positive, for the CDF-based likelihood, the Hessian is positive semi-definite (PSD). Therefore, the CDF-based likelihood is a convex function of $\mathbf{x}$.

Solution space

Having showed the convexity of the likelihood, the minimizer is now analyzed. We define the set $\mathcal{X}_1 = \{\mathbf{x}\,|\mathbf{y}_i\mathbf{h}_i^{\mathrm{T}}\mathbf{x} > 0, \forall\, i = 1, 2, \ldots, 2N\}$. Consider two cases for the minimizer, based on the SNR at the BS receiver.

- Case 1: Separable solution - There exists at least one finite $\mathbf{x}$, for which $\mathbf{y}_i\mathbf{h}_i^{\mathrm{T}}\mathbf{x} > 0$, $\forall\, i = 1, 2, \ldots, 2N$. This corresponds to operating in a high SNR regime.
- Case 2: Non-separable solution - There exists no $\mathbf{x}$, such that $\mathbf{y}_i\mathbf{h}_i^{\mathrm{T}}\mathbf{x} > 0$, $\forall\, i = 1, 2, \ldots, 2N$, i.e., $\mathcal{X}_1$ is a null set. This corresponds to low SNR operation.

In Case 1, the Gaussian CDF approaches 1 asymptotically. Thus the minimum value of the likelihood (3.6) cannot be attained by any finite $\mathbf{x}$. The high-SNR saturation of the performance the GD algorithms is analyzed by operating in this case, as seen in later sections.

For Case 2, it is evident that the optimal value $\mathbf{x}^*$ for minimizing the likelihood is bounded. The significance of analyzing this case is presented after the smoothness analysis of the likelihood (see Remark 3.3.1).

Smoothness

The function $\mathcal{L}(\mathbf{x})$ is β-smooth if

$$\mathcal{L}_\beta(\mathbf{x}) = \frac{\beta}{2}\,||\mathbf{x}||^2 - \mathcal{L}(\mathbf{x}) \tag{3.14}$$

is convex [140]. Utilizing (3.7b) and (3.9c), the Hessian for $\mathcal{L}_\beta(\mathbf{x})$, $\mathcal{H}^\beta_{\mathbf{x}}$, is given by

$$\mathcal{H}^\beta_{\mathbf{x}} = \beta\,\mathbf{I} - \mathbf{H}^{\mathrm{T}}\,\mathrm{diag}(\zeta''(\mathbf{G}\mathbf{x}))\,\mathbf{H}. \tag{3.15}$$

In order to show the Hessian to be PSD, consider any vector $\mathbf{z} \in \mathbb{R}^{2K}$. We have

$$\begin{aligned}
\mathbf{z}^{\mathrm{T}}\mathcal{H}^\beta_{\mathbf{x}}\mathbf{z}^{\mathrm{T}} &= \beta\,\mathbf{z}^{\mathrm{T}}\mathbf{I}\mathbf{z} - \mathbf{z}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\,\mathrm{diag}(\zeta''(\mathbf{G}\mathbf{x}))\,\mathbf{H}\mathbf{z} \\
&> \beta\,\mathbf{z}^{\mathrm{T}}\mathbf{I}\mathbf{z} - \mathbf{z}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{z} \\
&= \beta\,||\mathbf{z}||^2 - ||\mathbf{H}\mathbf{z}||_2^2 \\
&\geq ||\mathbf{z}||^2(\beta - ||\mathbf{H}||_2^2),
\end{aligned} \tag{3.16}$$

where $||\mathbf{H}||_2$ is the ℓ_2-norm of the matrix $\mathbf{H}$. The inequality (3.13) and the Cauchy-Schwartz inequality are utilized above. The Hessian is PSD if

$$\beta \geq ||\mathbf{H}||_2^2. \tag{3.17}$$

Thus, the cdf-based likelihood is a smooth function with the smoothness parameter β lower bounded by $||\mathbf{H}||^2$. Based on this, we have the following:

- The smoothness parameter thus depends on the chosen channel matrix. This captures the dimensionality of the problem, i.e., the number of users and MIMO antennas.
- The optimal step size for the improved GD method $\alpha^{(t)}$ is given by $1/\beta$. If the number of users or antenna elements increases, the optimal step size reduces.

Remark. *Note that this smoothness characterization is valid for the likelihood, irrespective of the solution being drawn from Case 1 or Case 2. The optimal value* $\mathbf{x}^*$ *is bounded for Case 2; hence the existing results for smooth functions [140] can be applied to this case. For Case 1 however, the choice of GD parameters and subsequent convergence analysis in the absence of a finite minimizer warrants explicit analysis. This case presents the high-SNR saturation regime of receiver algorithm. Thus, for the remainder of this work, all subsequent analysis and algorithm design is performed from the perspective of operating under Case 1.*

3.3.2 Improved Gradient Descent for log-CDF likelihood

One of the limitations of applying GD to the CDF-based likelihood (3.9a) is the evaluation of the gradient (3.9b) for large negative arguments. The Gaussian CDF quickly decreases to zero for negative values of z and thus the numerical evaluation of the gradient runs into instabilities due to inability of capturing such low values within floating point precision. Any approximation of computing this gradient expression must preserve the descent direction, not just address the numerical instabilities. For this reason, regularizing the denominator of (3.9b) by a scalar constant is not a feasible solution.

A key observation here is that the gradient computation of (3.9b) does not necessarily require the computation of the individual PDF and CDF terms $\phi(z)$ and $\Phi(z)$, respectively; only the ratio of the two terms is essential. The core principle to improve robustness for the CDF-based GD algorithm thus involves a numerically efficient method to evaluate the ratio $\zeta'(z) = \phi(z)/\Phi(z)$, for large negative arguments z.

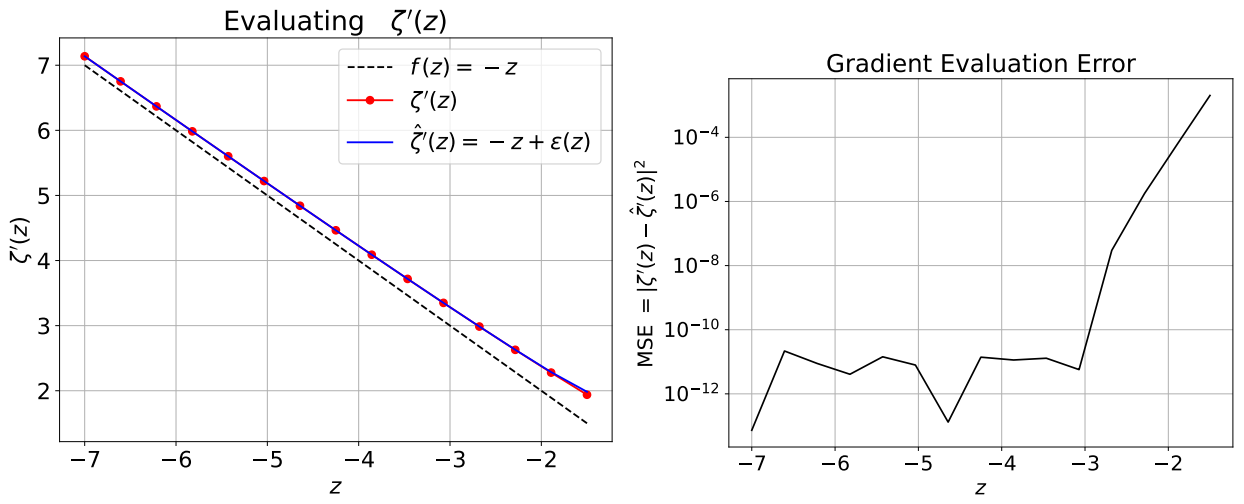
Lemma 1 derives an upper and lower bound for this ratio $\zeta'(z)$ for $z < 0$. From this, it is evident that the value of $\zeta'(z)$ asymptotically approaches the linear function $f(z) = -z$ as $z \to -\infty$. For negative values below a threshold z_{thresh}, a surrogate gradient value, using a residual $\epsilon(z)$, is evaluated as

$$\hat{\zeta}'(z) = -z + \epsilon(z), \text{ for } z < z_{\text{thresh}}. \tag{3.18}$$

As seen in Lemma 1, the value of $\zeta'(z)$ is sandwiched between $-z$ and $-z - 1/z$ for $z < 0$. Thus $\epsilon(z) \to 0$ as $z \to -\infty$. This residual is empirically evaluated, utilizing the series expansion

$$\epsilon(z) = -\frac{1}{z} + \frac{c_2}{z^2} + \frac{c_3}{z^3} + \frac{c_4}{z^4} + \dots \tag{3.19}$$

Using the least squares fit, the coefficient values are evaluated as $c_2 = -0.09, c_3 = 1.80, c_4 = 1.95$. Further, we observe that the computation of the residual up to 4 orders, i.e., $\frac{c_4}{z^4}$ is sufficient for the desired accuracy in gradient evaluation. The plots in Fig. 3.1 illustrate the fit of the gradient using (3.18)-(3.19). Based on



(a) (b)

Figure 3.1: Accuracy of the numerically stable gradient of the CDF-based likelihood (a) Comparing the curve fit of (3.9b) and (3.20), (b) Mean square error of using the approximation (3.20)

Fig. 3.1, the evaluation of the gradient using (3.18) approximates the actual gradient to a high degree of accuracy for large negative values. Thus, the surrogate gradient $\hat{\zeta}'(z)$ is

$$\zeta'(z) = \begin{cases} -z - \frac{1}{z} + \frac{c_2}{z^2} + \frac{c_3}{z^3} + \frac{c_4}{z^4}, & \text{for } z < z_{\text{thresh}} \\ -\frac{\phi(z)}{\Phi(z)}, & \text{for } z \geq z_{\text{thresh}} \end{cases} \tag{3.20}$$

As opposed to (3.9b), the surrogate $\hat{\zeta}'(\cdot)$ in (3.20) avoids the CDF computation and any ensuing numerical instabilities. An improved GD algorithm for the log-CDF based likelihood is presented in Algorithm 13, with the following salient features:

- The vector $\mathbf{G}\mathbf{x}^{(t)}$ for the t^{th} iteration is evaluated.
- Based on a pre-determined threshold [1] $z_{\text{thresh}} = -5$, each index of the vector $\mathbf{G}\mathbf{x}^{(t)}$ is classified as $\mathcal{I}^+$ or $\mathcal{I}^-$ (see line 2-3 in Algorithm 13).
- Depending on the classification of each index, $\zeta'(\left[\mathbf{G}\,\mathbf{x}^{(t)}\right]_i)$ is evaluated using (3.20).
- The final output $\mathbf{x}^{(T)}$ is normalized to the M-QAM magnitudes, as required

[1] empirically chosen threshold based on numerical results (see Fig. 3.1(b))

Algorithm 3: Improved GD for log-CDF likelihood

Input: T, $\mathbf{G}$, $\mathbf{x}^{(0)} = 0$, $\{\alpha^{(t)}\}_{t=0}^{T-1}$, z_{thresh}
Output: $\mathbf{x}^{(T)}$

for $t = 0$ **to** $T-1$ **do**
  $\mathcal{I}^+ = \{i \,|\, \left[\mathbf{G}\,\mathbf{x}^{(t)}\right]_i \geq z_{\text{thresh}}\}$;
  $\mathcal{I}^- = \{i \,|\, \left[\mathbf{G}\,\mathbf{x}^{(t)}\right]_i < z_{\text{thresh}}\}$;
  for i *in* $\mathcal{I}^-$ **do**
    Evaluate $\zeta'(\left[\mathbf{G}\,\hat{\mathbf{x}}^{(t)}\right]_i)$ as (3.20), case 1;
  end
  for i *in* $\mathcal{I}^+$ **do**
    Evaluate $\zeta'(\left[\mathbf{G}\,\hat{\mathbf{x}}^{(t)}\right]_i)$ as (3.20), case 2;
  end
  Evaluate $\nabla_{\mathbf{x}}^{(t)}$ using (3.7a);
  Update $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha^{(t)}\nabla_{\mathbf{x}}^{(t)}$;
end
$\mathbf{x}^{(T)} \leftarrow \eta \frac{\mathbf{x}^{(T)}}{||\mathbf{x}^{(T)}||}$;

3.3.3 Accelerated Gradient Descent for faster convergence

The general accelerated gradient descent (AGD) method for a convex β-smooth function was first introduced in [141] as an algorithm to attain the optimum oracle complexity for smooth convex functions and has been widely applied to various applications in signal processing [142].

Applying AGD to the CDF-based one-bit likelihood optimization (3.8), gives the update

$$\mathbf{d}^{(t)} = \gamma^{(t)}\,(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}) \tag{3.21a}$$

$$\hat{\mathbf{x}}^{(t)} = \mathbf{x}^{(t)} + \mathbf{d}^{(t)} \tag{3.21b}$$

$$\mathbf{x}^{(t+1)} = \hat{\mathbf{x}}^{(t)} - \alpha^{(t)}\nabla_{\mathbf{x}}\mathcal{L}(\hat{\mathbf{x}}^{(t)}). \tag{3.21c}$$

Here, $\mathbf{d}^{(t)}$ is the momentum update at the t^{th} iteration, which is a step taken in addition to the gradient step. The scalar $\gamma^{(t)}$ is the weighting coefficient for the momentum. The gradient $\nabla_{\mathbf{x}}\mathcal{L}(\hat{\mathbf{x}}^{(t)})$ is evaluated using the improved gradient method described in Sec. 3.3.2.

The AGD algorithm, utilizing the improved GD update for the CDF likelihood, is presented in Algorithm 15. Different from the GD, i.e., Algorithm 13, AGD is able to modify the update step with an additional correction from the gradient direction, determined by the previous estimates. The momentum $\mathbf{d}^{(t)}$ in (3.21a) accumulates the gradients from the previous iterations, preventing the algorithm slowdown due to vanishing gradient[142]. The momentum endows a "speed" to the GD algorithm, preventing saturation in such regions of very low gradient values. This is particularly effective for speeding up the likelihood decrease

Algorithm 4: Accelerated GD for log-CDF likelihood

Input: T, $\mathbf{G}$, $\mathbf{x}^{(0)} = 0$, $\mathbf{d}^{(0)} = 0$, $\{\alpha^{(t)}\}_{t=0}^{T-1}$, z_{thresh}, γ
Output: $\mathbf{x}^{(T)}$

for $t = 0$ **to** $T - 1$ **do**
  Evaluate $\hat{\mathbf{x}}^{(t)} = \mathbf{x}^{(t)} + \mathbf{d}^{(t)}$;
  $\mathcal{I}^{+} = \{i \,|\, \left[\mathbf{G}\,\hat{\mathbf{x}}^{(t)}\right]_i \geq z_{\text{thresh}}\}$;
  $\mathcal{I}^{-} = \{i \,|\, \left[\mathbf{G}\,\hat{\mathbf{x}}^{(t)}\right]_i < z_{\text{thresh}}\}$;
  for i *in* $\mathcal{I}^{-}$ **do**
    Evaluate $\zeta'(\left[\mathbf{G}\,\hat{\mathbf{x}}^{(t)}\right]_i)$ as (3.20), case 1;
  end
  for i *in* $\mathcal{I}^{+}$ **do**
    Evaluate $\zeta'(\left[\mathbf{G}\,\hat{\mathbf{x}}^{(t)}\right]_i)$ as (3.20), case 2;
  end
  Evaluate $\nabla^{(t)}_{\hat{\mathbf{x}}^{(t)}}$ using (3.7a);
  Update $\mathbf{x}^{(t+1)} = \hat{\mathbf{x}}^{(t)} - \alpha^{(t)}\nabla^{(t)}_{\hat{\mathbf{x}}}$;
  Update $\mathbf{d}^{(t+1)} = \gamma\,(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)})$;
end
$\mathbf{x}^{(T)} \leftarrow \eta\,\frac{\mathbf{x}^{(T)}}{||\mathbf{x}^{(T)}||}$;

for the CDF-based likelihood optimization without any finite minimizer, as detailed next.

3.3.4 Likelihood decay for the GD-based algorithms

Through this section we analyze the likelihood decay performance for the unconstrained GD Algorithms 13 and 15.

Likelihood decay for GD

The likelihood decay for smooth functions with a finite minimizer and minima has been extensively analyzed [140]. In particular, the GD iterations decrease the likelihood function at the rate $1/t$. However, as stated in Sec 3.3.1, there is no finite $\mathbf{x} \in \mathbb{R}^{2K}$ that achieves the infimum of the likelihood, i.e., $\mathcal{L}(\mathbf{x}) > 0\,\forall\,\mathbf{x}$ (see Remark 3.3.1). The convergence is thus analyzed to a surrogate minimum ϵ of the likelihood. The following theorem provides the likelihood decay after T iterations of the improved GD algorithm.

Theorem 2. *For a given surrogate minimum ϵ, the likelihood decay after T steps of the GD algorithm, i.e., Algorithm 13, with step size $\alpha^{(t)} = \frac{1}{\beta}$, is given by*

$$\mathcal{L}(\mathbf{x}^{(T)}) - \epsilon \leq \frac{\lambda_0(\epsilon)}{T + \lambda_1(\epsilon)} \tag{3.22}$$

where the scalars $\lambda_0(\epsilon)$ and $\lambda_1(\epsilon)$ are dependent on ϵ.

The proof follows the same steps as the general decay rate analysis using a finite minimizer [140], with the surrogate minimum ϵ for the likelihood appropriately chosen. The exact steps for this proof is provided in Appendix 3.A of this chapter.

Theorem 2 provides the best possible convergence rate for GD, utilizing the optimally chose step size, i.e., $\alpha^{(t)} = 1/\beta$, for a finite horizon GD algorithm.

Comparing likelihood decay for GD and AGD

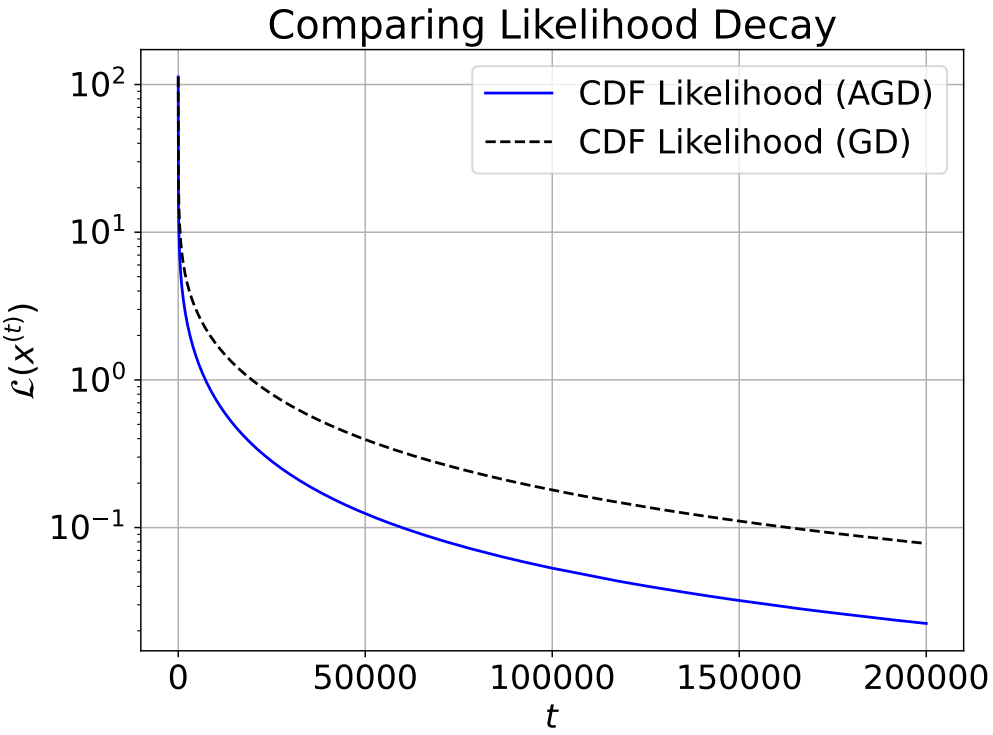


Figure 3.2: Comparing decrease in CDF-based likelihood for AGD vs GD.

For a general β-smooth function $f(\mathbf{x})$, the AGD has been proven to converge to the minimum as [140, 141]

$$f(\mathbf{x}^{(T)}) - f(\mathbf{x}^*) < \frac{c_1}{T^2}. \tag{3.23}$$

Applying the AGD to the recovery of symbols by minimizing the one-bit likelihood (3.8) shows a similar gain in convergence rate. This is empirically illustrated in Fig. 3.2, comparing the likelihood convergence rate to the infimum for GD vs AGD. As seen from the plots, the likelihood decays to a much lower value for AGD, with the gap to the GD-based likelihood decay increasing with T. This empirically illustrates the strength of using the AGD for the unconstrained optimization (3.8).

Through theoretical bounds on likelihood decay, as well as the empirical results for AGD, we illustrate that unconstrained GD-based techniques will be able to converge arbitrarily close to the infimum, provided that there are no constraints on the number of iterations. This greatly scales the GD horizon T, proving infeasible for practical receivers, operating to minimize computational complexity. Resilient and simplified GD for practical receivers is added through *(i)* Improved surrogate likelihoods to allow for larger step sizes and thereby speed up convergence, and *(ii)* Projected GD to efficiently converge to the solution within the constrained set $\mathcal{M}^{2K}$. Each of these is elaborated in Sections 3.4 and 3.5, respectively.

3.4 Improved CDF Surrogates for Modeling One-bit Likelihood

This section explores surrogate functions of the CDF, focusing primarily on the logistic regression (LR), to model an approximate one-bit likelihood for signal recovery. Insights into the improved likelihood decay for the LR are provided, followed by the GD algorithms for this likelihood.

3.4.1 Modeling one-bit likelihood through logistic regression

The approximation for the Gaussian CDF using the sigmoid function was first proposed in [143]. This was initially applied to the one-bit MIMO receiver in [1], where, motivated by the utilization of the sigmoid function as a prevalent nonlinear activation in DNNs, the GD-based receiver was implemented as an unfolded DNN, i.e., the OBMNet.

The LR-based likelihood expression involves substituting the value of the sigmoid function $\sigma(z)$ for the general likelihood $\zeta(z)$ in (3.6), giving the expressions

$$\zeta(z) = -\log \sigma(z) \tag{3.24a}$$

$$\zeta'(z) = -\sigma(-z) \tag{3.24b}$$

$$\zeta''(z) = \sigma(z)(1-\sigma(z)). \tag{3.24c}$$

The unconstrained ML optimization is given by

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \underset{\mathbf{x}\in\mathbb{R}^{2K}}{\operatorname{argmin}} \sum_{i=1}^{2N} -\log \sigma\big(y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}\big). \tag{3.25}$$

The following is the analysis the LR-based likelihood.

Convexity

On substituting the Hessian for the LR-based likelihood (3.26) in (3.7b), it is evident that each element of the matrix $\text{diag}(\zeta''(\mathbf{Gx}))$ is always positive. Thus the Hessian is positive semi-definite (PSD). Therefore, the LR-based likelihood is a convex function of $\mathbf{x}$.

Smoothness

Analogous to (3.15), $\mathcal{H}_\beta$ is evaluated as

$$\mathcal{H}_\beta = \beta\,\mathbf{I} - \mathbf{H}^{\mathrm{T}}\text{diag}(\sigma(\mathbf{Gx})(1-\sigma(\mathbf{Gx})))\mathbf{H}. \tag{3.26}$$

For any vector $\mathbf{z} \in \mathbb{R}^{2K}$. We have

$$\begin{aligned}
\mathbf{z}^{\mathrm{T}}\mathcal{H}_{\mathbf{x}}^{\beta}\mathbf{z}^{\mathrm{T}} &= \beta\,||\mathbf{z}||^2 - \mathbf{z}^{\mathrm{T}}\mathbf{H}^{\mathrm{T}}\text{diag}(\sigma(\mathbf{Gx})(1-\sigma(\mathbf{Gx}))\mathbf{Hz} \\
&\geq \beta\,||\mathbf{z}||^2 - ||\text{diag}\big(\sqrt{\sigma(\mathbf{Gx})(1-\sigma(\mathbf{Gx}))}\,\big)\mathbf{Hz}||^2 \\
&> \beta\,||\mathbf{z}||^2 - \frac{1}{4}||\mathbf{Hz}||_2^2 \\
&\geq ||\mathbf{z}||^2(\beta - \frac{1}{4}||\mathbf{H}||_2^2),
\end{aligned} \tag{3.27}$$

where we utilize $\sigma(z)(1-\sigma(z)) \leq 1/4\ \forall\, z$, and the Cauchy-Schwartz inequality. The Hessian is PSD if

$$\beta \geq \frac{||\mathbf{H}||_2^2}{4}. \tag{3.28}$$

Comparing this to (3.17) gives $\beta_{\text{LR}} = \frac{1}{4}\beta_{\text{CDF}}$. Following the model-based selection of the step size $\alpha^{(t)} = 1/\beta$, the LR enables an increase by a factor of 4.

3.4.2 Step size robustness of LR for GD

In addition to better smoothness characterization over the CDF, the LR offers additional robustness to larger step sizes $\alpha^{(t)} >> 1/\beta_{\text{LR}}$, resulting in faster likelihood decay without diverging to the incorrect solution. This is attributed to the properties of the Hessian matrix; the plots in Fig. 3.3 pictorially show different behavior, which translates to increased robustness for LR. This is further elaborated below.

Consider the general likelihood expression $\mathcal{L}(\mathbf{x})$, given by (3.6). As described earlier, smoothness parameter β determines the step size for the GD algorithm. Following the analysis in (3.16) and (3.27), we

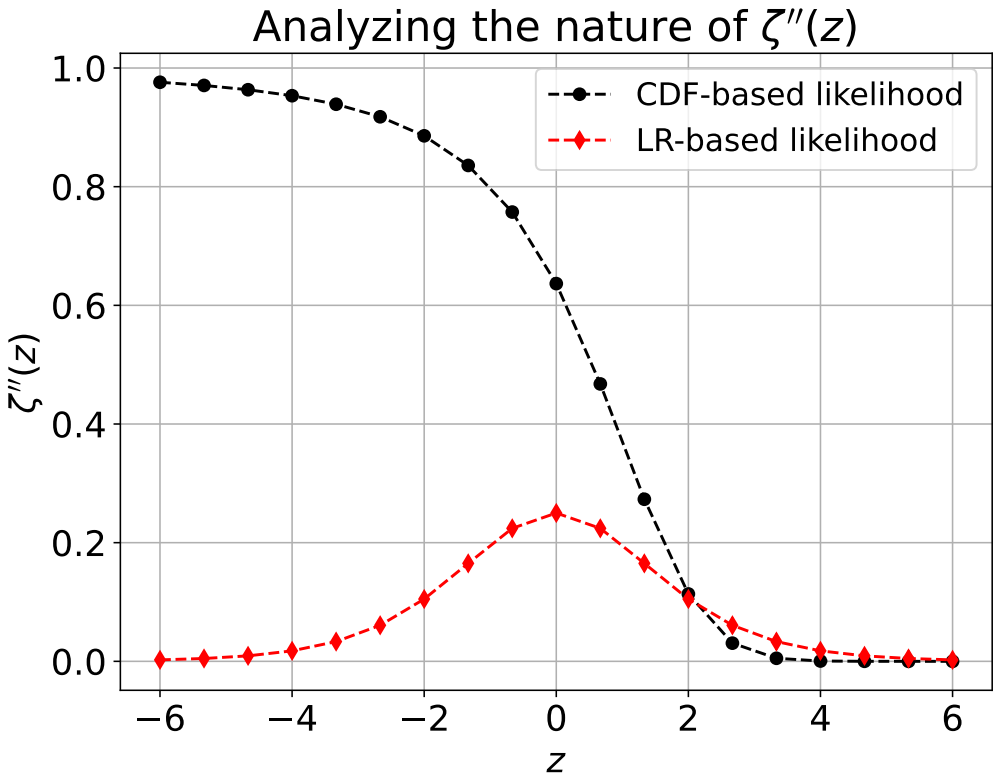


Figure 3.3: Comparing the values of $\zeta''(z)$ for the LR and CDF-based likelihoods.

have that the following bound on β

$$\beta \geq ||\text{diag}\big(\sqrt{\zeta''(\mathbf{Gx})}\,\big)\,\mathbf{H}||_2^2. \tag{3.29}$$

Further, the maximum value for the RHS of (3.29) is given by

$$\beta_{\max} = \max_{\mathbf{x}\in\mathbb{R}^{2K}} ||\text{diag}\big(\sqrt{\zeta''(\mathbf{Gx})}\big)\,\mathbf{H}||_2^2 = \Big[\max_{z\in\mathbb{R}} \zeta''(z)\Big]||\mathbf{H}||_2^2. \tag{3.30}$$

Choosing the step size $\alpha^{(t)}$ utilizing the value $\beta = \beta_{\max}$ is sufficient to guarantee convergence of GD over all $\mathbf{x} \in \mathbb{R}^{2K}$. This is the model-based limit chosen for the CDF and LR-based likelihoods, as given in Sec 3.3.1 and 3.4.1, respectively. However, a step size larger than this limit can be utilized by analyzing the value of the $\mathbf{x}^{(t)}$ for the attainment of $\beta = \beta_{\max}$. This, in turn, depends on the value of $\zeta''(z^{(t)})$, which is compared for both the LR and CDF-based likelihoods in Fig. 3.3. Based on the plots, we analyze this further.

LR-based likelihood

As seen by the curve for the LR-based likelihood in Fig. 3.3, the value for $\beta_{\max}$ is attained at $z = 0$, corresponding to the case $\mathbf{x} = 0$. However, the practical GD trajectory is considered via the attainment of the values of $\zeta''(z^{(t)})$ for two zones, as seen in Fig. 3.3: *(i)* The convergence zone corresponding to the positive z-axis, and *(ii)* The divergence zone corresponding to the negative z-axis

- Convergence zone, i.e., $[\mathbf{Gx}]_i > 0\ \forall\, i$: It has been shown that $||\mathbf{x}^{(t)}||$ increases unbounded with each GD iteration[2]. Thus the gap of the expression $\zeta''(z^{(t)})$ to the maximum value of 0.25 increases monotonically. This, in turn allows a much larger step size, i.e., $\alpha^{(t)} >> 1/\beta_{\max}$.

- Divergence zone, i.e., $[\mathbf{Gx}]_i < 0\ \forall\, i$: The symmetry of the plot for $\zeta''(z^{(t)})$ plays an important role for the divergence zone as well. With increasing divergent behavior, i.e., increasing negative values of $z^{(t)}$, the gap of $\zeta''(z^{(t)})$ to the maximum value also increases. This further increases the maximum value of the step size that can be taken to move in the convergence direction. Since the value of $\zeta''(z^{(t)})$ can decrease to zero, there will always exist a point after which the GD algorithm (with a fixed step size) will move in the direction of convergence.

The same logic will also hold for intermediate behavior between convergence and divergence zones of the GD algorithm, wherein the GD algorithm will never indefinitely diverge.

[2]Proof of Theorem 2 shows the conditions for monotonic decrease of $||\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}||$.

CDF-based likelihood

For the CDF-based likelihood plots in Fig. 3.3, the same robustness as the LR will hold for the convergence zone. However, for the GD algorithm dynamics in the divergence zone, the dependence on the step size is inverted compared to the LR-based likelihood. With increasing negative values of $\zeta''(z^{(t)})$, the gap to $\beta_{\max}$ decreases. This implies the need to take smaller step sizes for convergence; using a larger step size will result in indefinite divergence of the GD algorithm. This increased sensitivity to step sizes larger than $1/\beta_{\max}$ results in use of step sizes smaller than LR, resulting in greater GD iterations for convergence.

3.4.3 GD for LR-based likelihood and algorithm convergence

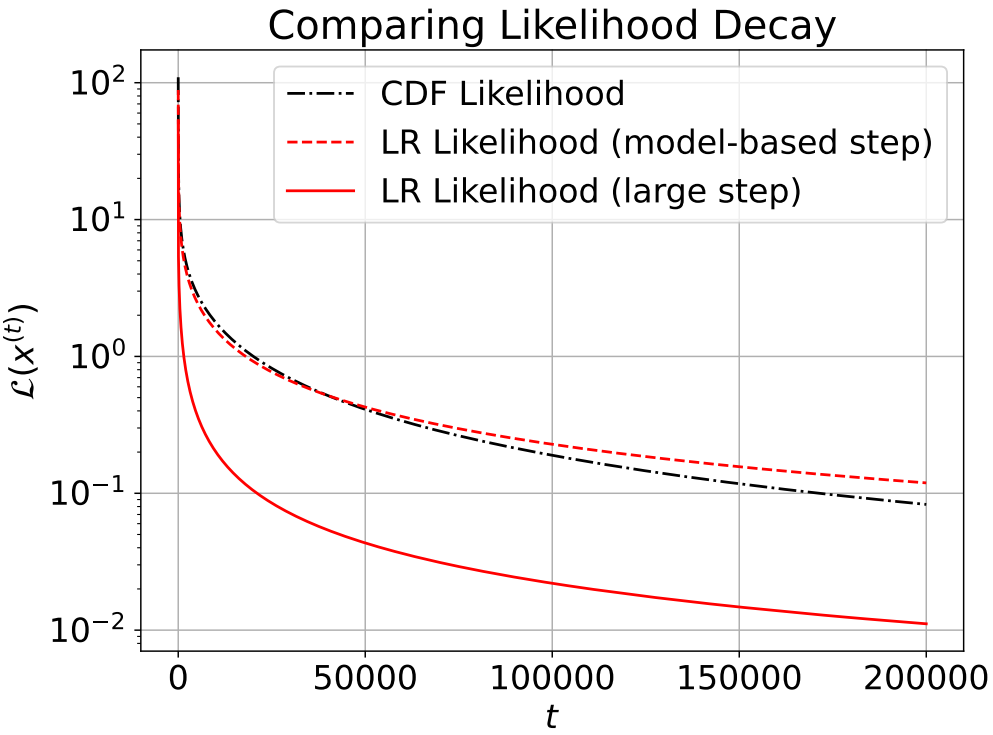


Figure 3.4: Comparing decrease in CDF-based likelihood vs LR-based likelihood, with model-based step size and large step size, due to GD.

Applying GD to the likelihood (3.25) gives the GD update

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \alpha^{(t)}\mathbf{G}^{\mathrm{T}}\sigma(-\mathbf{G}\mathbf{x}^{(t)}). \tag{3.31}$$

Similar to the application of the GD for the CDF-based likelihood, the choice of the step size parameter $\alpha^{(t)}$ is dependent on β_{LR}. In order to guarantee convergence of GD, the step size is chosen such that,

$\alpha^{(t)} = 1/\beta_{\max}$, as explained in 3.4.1. However, the LR is more resilient to larger step sizes, allowing for faster convergence.

The convergence analysis for GD algorithm of the LR follows the same analysis as the CDF-based likelihood, i.e., Theorem 2. However the specific constants will differ for the LR, owing to the different likelihood function. This convergence of the GD algorithm for the LR-based likelihood is illustrated in Fig. 3.4. The plots compare the GD convergence of the LR-based likelihood, using both the model-based step size $\alpha^{(t)} = 1/\beta_{\max}$ and the large step size $\alpha^{(t)} >> 1/\beta_{\max}$, to the CDF-based likelihood. All the GD-based algorithms decay as $1/t$, validating Theorem 2. The similar decay performance for the GD algorithms with the model-model based step size is attributed to the fact that $\alpha^{(t)}_{\text{LR}} = \alpha^{(t)}_{\text{CDF}}$. However, the larger step size resilience for the LR-based likelihood is clearly seen by the significantly improved convergence.

Remark. *Although the OBMNet [1] learns the step sizes $\alpha^{(t)}$ at each GD iteration, these do not need to be explicitly learnt. The evaluation of the Lipschitz constant β, theorizes the the required optimal step size. Additionally, empirical evaluation of the GD algorithm (OBMNet) with learnt step sizes and static step sizes shows no difference in performance. However, the latter enables the analysis of the rate of likelihood decrease.*

3.4.4 AGD for LR-based likelihood and algorithm convergence

The AGD algorithm, introduced in 3.3.3, is applied to the LR-based likelihood. The resulting GD update step is

$$\hat{\mathbf{x}}^{(t)} = (1 + \gamma^{(t)})\mathbf{x}^{(t)} - \gamma^{(t)}\mathbf{x}^{(t-1)} \tag{3.32a}$$

$$\mathbf{x}^{(t+1)} = \hat{\mathbf{x}}^{(t)} + \alpha^{(t)}\mathbf{G}^{\mathrm{T}}\sigma(-\mathbf{G}\hat{\mathbf{x}}^{(t)}). \tag{3.32b}$$

The entire T-step AGD (3.32) can also be equivalently implemented as a T-layer unfolded DNN, i.e., the accelerated-OBMNet (A-OBMNet). Different from the original OBMNet with individual disjoint subnetworks to implement (4.5), the A-OBMNet also additionally links the each subsequent OBMNet sub-network stage and adds increased robustness to the signal recovery. At each iteration, we can learn the scalar coefficients $\alpha^{(t)}$ and $\gamma^{(t)}$ using the loss function (4.7). As stated in Remark 3.4.3, for the A-OBMNet too, we empirically observe no difference in performance for learning the parameters $\{\alpha^{(t)}, \gamma^{(t)}\}_{t=1}^{T}$ or statically choosing these through the smoothness properties of the LR-based likelihood. The performance comparison of the A-OBMNet to the original OBMNet is given in Sec. 4.6.

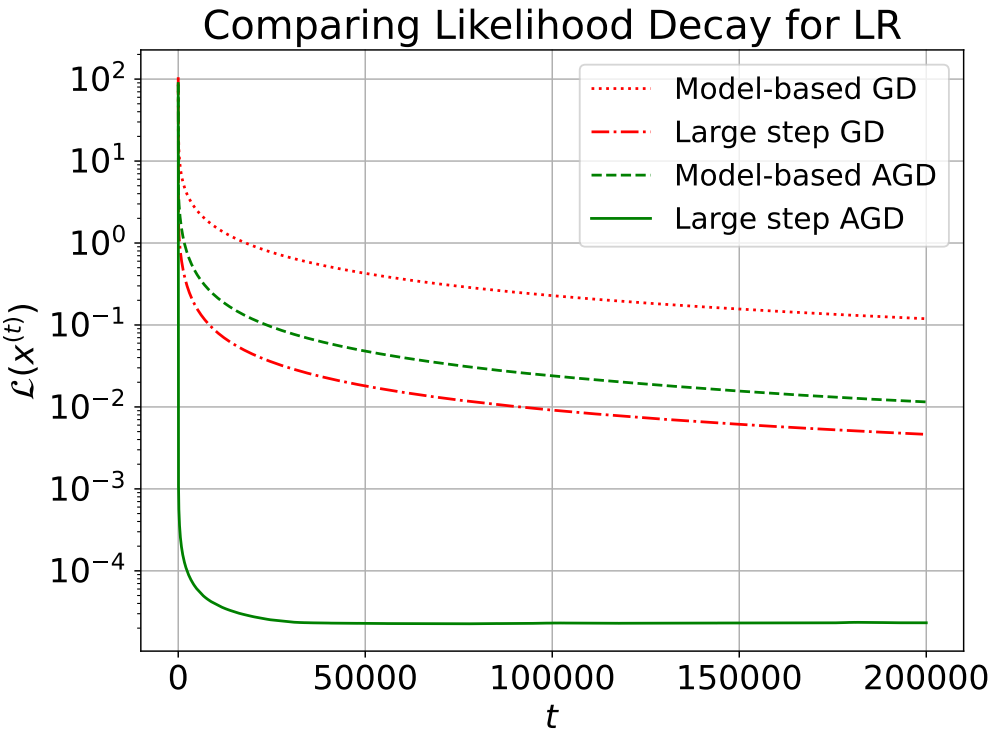


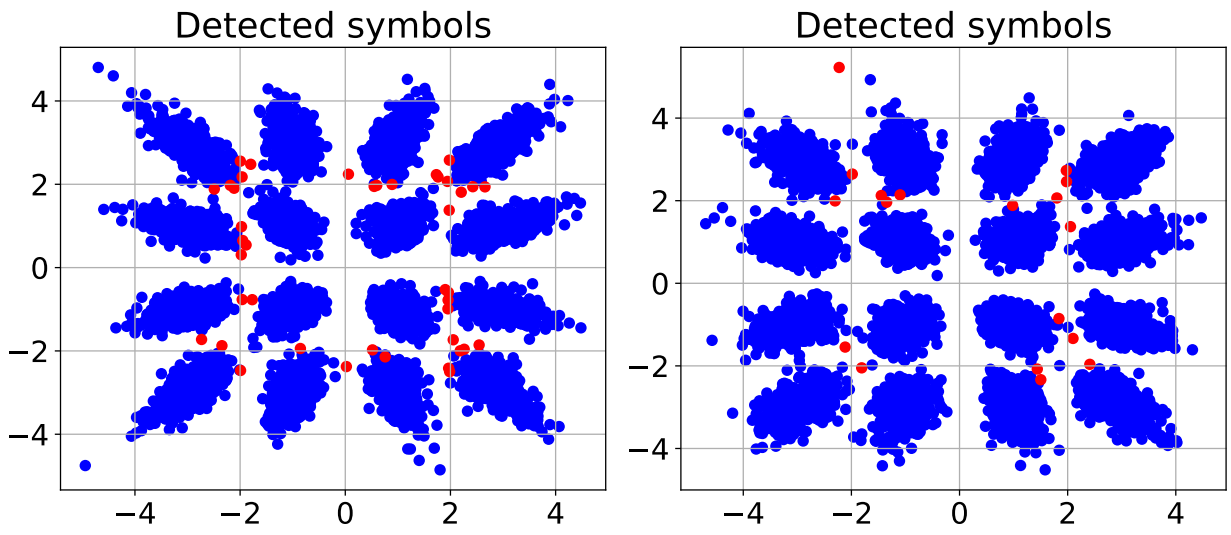
Figure 3.5: Comparing decrease in LR-based likelihood for different variants of the GD algorithms

3.5 Projected Gradient Descent - DNN-Aided Optimization for M-QAM Symbols

This section begins by elucidating the significance of the projection step. This is followed by the general two-tier projection strategy employed for the M-QAM constellation symbols. Finally, the entire projected AGD algorithm is implemented as an unfolded DNN, the A-PrOBNet.

3.5.1 Significance of M-QAM projection for GD

One of the main limitations of of applying the unconstrained GD algorithm, optimizing over $\mathbb{R}^{2K}$, for the recovery of symbols generated from the M-QAM constellation is symbol recovery with large cluster spread. The recovered symbols are illustrated in Fig. 3.6. The consequences of this large cluster spread on the unconstrained GD-based symbol recovery, specifically Algorithms 13 and 15, are explained below.



(a) CDF-based likelihood (3.8)

(b) LR-based likelihood (3.25)

Figure 3.6: Recovered 16-QAM constellation plots using unconstrained GD for M-QAM constellations with $K = 8$ users and $N = 128$ BS antennas (blue - correctly detected and red are incorrectly detected symbols).

Slow rate of gradient decay

We begin by first understanding the road to convergence, specifically through the gradient decay. Consider the expression for the gradient at the t^{th} iteration for a general likelihood, i.e., (3.7a),

$$\nabla_{\mathbf{x}}^{(t)} = \mathbf{G}^{\mathrm{T}}\, \zeta'(\mathbf{G}\mathbf{x}) = \sum_{i=1}^{2N} \mathbf{g}_i\, \zeta'(y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}^{(t)}), \tag{3.33}$$

where $\mathbf{g}_i$ and $\mathbf{h}_i$ are the i^{th} rows of the matrices $\mathbf{G}$ and $\mathbf{H}$, respectively. Firstly, the function $\zeta'(\cdot)$ is strictly positive-valued and the rows are drawn from a normal distribution, the gradient decays to zero if $\zeta'(y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}^{(t)}) \to 0, \forall i$. Secondly, the input $\mathbf{x}$ is drawn from the M-QAM constellation points. Both these factors imply that for all i, $y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}^{(t)}$ should be large positively-scaled constellation symbols, with very low cluster spread, in order for the gradient to decay to zero.

The presence of large symbol cluster spread affects the positivity of the expression $y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}^{(t)}$ for some indices, even though the recovered symbols are within the right symbol boundaries. This is an induced negative bias, due to large cluster spreads. Due to this negative bias, the GD is significantly slowed down, correcting for both incorrectly detected symbols as well as reducing the cluster spread of correctly detected symbols. This makes the GD process very slow and inefficient, if applied by itself, as seen from the different convergence results of Sec 3.3 and 3.4. The slow convergence is corrected through the use of projected GD, as explained below.

GD step – projection step positive feedback

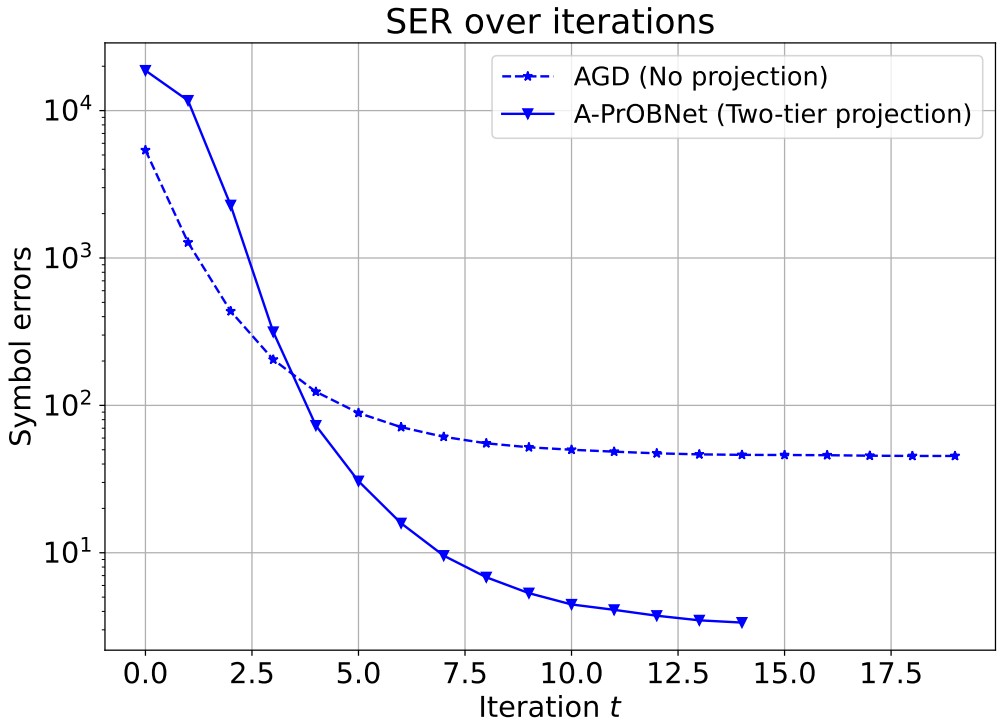


Figure 3.7: Iteration dynamics of SER: Comparing CDF-based AGD with and without projection. Recovery of 16-QAM symbols received from $K = 8$ users at a BS antenna with $N = 128$ antennas for SNR $= 25$ dB

The improvement in convergence via projected GD is pictorially illustrated in Fig. 3.7, portraying the symbol error rate (SER) reduction over the GD iterations. In the absence of any projection step at each GD iteration, the SER quickly saturates and further reduction is very slow, i.e., the rate of symbol error correction doesn't follow the rate of likelihood decay. In the absence of projection, the GD iteration itself works towards reducing the cluster spread, which does not have any bearing on the SER. On the other hand, the the two-tier projection (explained in Sec. 3.5.2) improves performance and speeds up convergence. The projection step, by itself, does not help correct symbol errors; it is only responsible for improved regularization of the recovered symbols into smaller clusters. This reduces the negative bias and the GD iteration is able to efficiently correct the M-QAM symbol errors in the subsequent step, which is further helpful to better regularize the recovered M-QAM symbols, and so on. This creates positive feedback with the projection step helping the GD step, and the GD step helping the projection step, to greatly speed up convergence.

Remark. *Although the above analyzes symbol recovery with GD Algorithms 13 and 15 for the CDF-based likelihood, these observations are general to the unconstrained optimization and also apply to the surrogate likelihood based on the LR.*

3.5.2 Two-tier projected GD framework

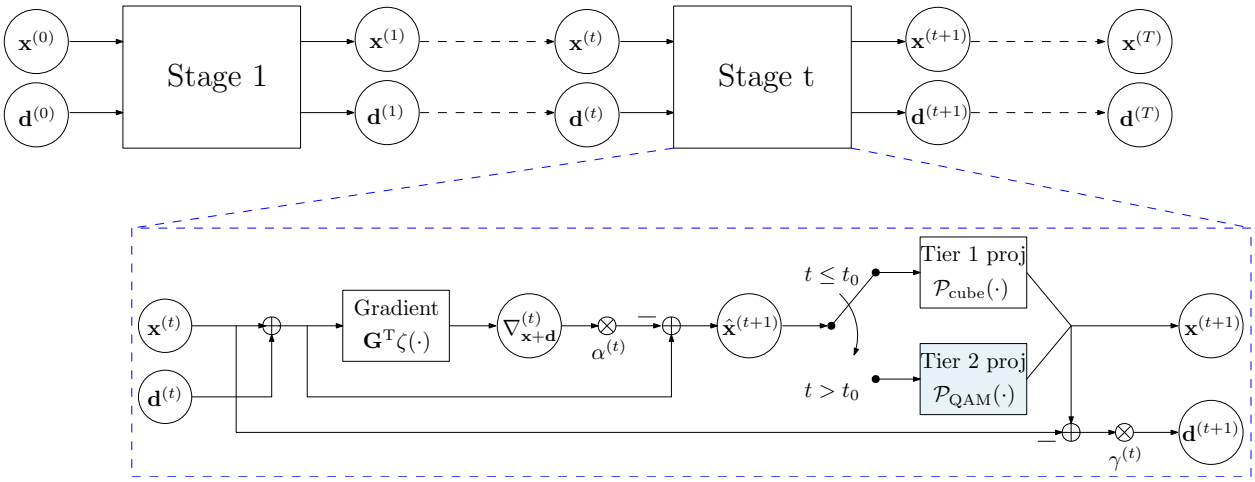


Figure 3.8: Block diagram for the A-PrOBNet - Unfolded DNN to implement the projected AGD update (3.37). The blue shaded blocks in each stage represent the learnable parameters in the unfolded DNN.

The use of a learnt M-QAM projection has been applied for one-bit MIMO in [20], which utilizes quantizers based on the rectified linear unit (ReLU) function. However, one of the major limitations of this approach is the absence of a structure for the projection, causing the detection to undergo unstable initial GD iterations, before being stabilized in the later stages. Differently, this work introduces a two-tier structured projection applied to the GD and AGD algorithms. This is explained below.

Tier 1 - Hypercube projection

The tier 1 projection maps each GD iterand to the M-QAM $2K$-dimensional hypercube, defined as $\mathcal{S}_{\mathrm{cube}} \in \mathbb{R}^{2K}$ such that

$$\mathcal{S}_{\mathrm{cube}} = \{\mathbf{x} \,|\, |\mathbf{x}[i]| \leq s_{\max}, \forall i = 1, 2, \ldots, 2K\}, \tag{3.34}$$

where $s_{\max}$ is the maximum value of the M-QAM quadrature component. We define the projection operation $\mathcal{P}_{\mathrm{cube}} : \mathbb{R}^{2K} \to \mathcal{S}_{\mathrm{cube}}$ through the element-wise transformation

$$\Big[\mathcal{P}_{\mathrm{cube}}(\mathbf{x})\Big]_i = \begin{cases} \mathbf{x}[i], \text{ if } |\mathbf{x}[i]| \leq s_{\max} \\ s_{\max}, \text{ otherwise.} \end{cases} , \forall i = 1, 2, \ldots 2K. \tag{3.35}$$

Applying this, we have the following projected GD update

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} - \alpha^{(t)} \nabla_{\mathbf{x}}^{(t)} \tag{3.36a}$$

$$\mathbf{x}^{(t+1)} = \mathcal{P}_{\text{cube}}(\hat{\mathbf{x}}^{(t+1)}). \tag{3.36b}$$

Similarly, applying this projection to the AGD method gives

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{d}^{(t)} - \alpha^{(t)} \nabla_{\mathbf{x}+\mathbf{d}}^{(t)} \tag{3.37a}$$

$$\mathbf{x}^{(t+1)} = \mathcal{P}_{\text{cube}}(\hat{\mathbf{x}}^{(t+1)}) \tag{3.37b}$$

$$\mathbf{d}^{(t+1)} = \gamma^{(t)}(\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}). \tag{3.37c}$$

The tier 1 hypercube projection improves GD as follows

- Bounding each $\mathbf{x}^{(t)}[i]$ as $-s_{\max} \leq \mathbf{x}^{(t)}[i] \leq s_{\max}$, the GD update (3.36) converges faster due to the larger value of the smoothness parameter β over the set $\mathcal{S}_{\text{cube}}$.
- The tier 1 projection is linear inside the M-QAM hypercube, which is a soft projection and hence not too restrictive. This allows for more flexible symbol recovery and error correction in the initial stages of the GD algorithm. This flexibility in projection enables the formation of the initial M-QAM constellation clusters for the recovered symbols, which are efficiently fined-tuned using the subsequent projection method.

Tier 2 - Gaussian denoiser

The tier 2 projection $\mathcal{P}_{\text{QAM}}$ maps from the set $\mathcal{S}_{\text{cube}} \to \mathcal{S}_{\text{cube}}$ through an exhaustive weighted sum of all the symbols in $\mathcal{M}^{2K}$. This requires modeling the posterior distribution of the transmitted symbols.

The vector of M-QAM transmitted symbols from the K different users is given by $\mathbf{s}$. Each GD iterand $\mathbf{x}^{(t)}$ after the tier 1 projection (3.35) is modeled as

$$\mathbf{x}^{(t)} = \mathbf{s} + \triangle\mathbf{s}^{(t)}, \tag{3.38}$$

where the residual $\triangle\mathbf{s}^{(t)}$ is the deviation from the transmitted symbols, drawn from the Gaussian distribution $\mathcal{N}(\mathbf{0}, (\sigma^{(t)})^2\mathbf{I})$. We assume that this residual component at the t^{th} iteration $\triangle\mathbf{s}^{(t)}$ is independent of the previous residuals $\{\triangle\mathbf{s}^{(t)}\}_{t=0}^{t-1}$. Further, we consider the uniform non-informative prior $\Pr(\mathbf{s})$ over all the symbols in $\mathcal{M}^{2K}$. The tier 2 projection $\mathcal{P}_{\text{QAM}}$ is the MMSE estimate of the transmitted symbols using this

estimation model for $\mathbf{x}^{(t)}$. Hence, using the modeled Gaussian distribution with the independent increment assumption, the tier 2 projection at each iteration is given as

$$\hat{\mathbf{s}}^{(t)} = \mathcal{P}_{\text{QAM}}(\mathbf{x}^{(t)}) = \mathbb{E}_{\mathbf{s}|\mathbf{x}^{(t)}}(\mathbf{s}), \tag{3.39}$$

which is the posterior mean of the distribution $\Pr(\mathbf{s}|\mathbf{x}^{(t)})$. Using the Gaussian likelihood $f(\mathbf{x}^{(t)}|\mathbf{s})$ and the uniform prior $\Pr(\mathbf{s})$, the MMSE estimate is given by

$$\hat{\mathbf{s}}^{(t)} = c^{(t)} \sum_{i=1}^{M^K} \mathbf{s}_i \exp\Big(- \frac{||\mathbf{x}^{(t)} - \mathbf{s}_i||^2}{2(\sigma^{(t)})^2} \Big), \tag{3.40}$$

where $c^{(t)} = \Big(\sum_{i=1}^{M^K} \exp\Big(- \frac{||\mathbf{x}^{(t)} - \mathbf{s}_i||^2}{2(\sigma^{(t)})^2} \Big) \Big)^{-1}$ is the normalization constant and $\mathbf{s}_i$ is the i^{th} element of $\mathcal{M}^{2K}$. The parameter $\sigma^{(t)}$ is a learnt over each iteration (see Sec. 3.5.3) Since $\mathbf{s}^{(t)}$ consist of $2K$ independent components, corresponding to the real and imaginary parts of K users, the element-wise evaluation of the tier 2 projection is given by

$$\hat{\mathbf{s}}[i] = c^{(t)} \sum_{k=1}^{\sqrt{M}} s_k \exp\Big(- \frac{(\mathbf{x}^{(t)}[i] - s_k)^2}{2(\sigma^{(t)})^2} \Big), \tag{3.41}$$

where s_k is the k^{th} quadrature component of the M-QAM constellation. The equation (3.41) is the Gaussian denoiser, formed by a convex summation of all the elements in $\mathcal{M}^{2K}$. This convex projection clearly also maps to a point in the hypercube $\mathcal{S}_{\text{cube}}$. Based on this projection, we have the following.

- Different from the tier 1 projection (3.35), the tier 2 projection is weighted by the ℓ_2 distance of the iterand $\mathbf{x}^{(t)}$ to each constellation point, via a Gaussian kernel. Thus, the values $\mathbf{x}^{(t)}[i]$ close to the constellation points $\{s_k\}$ are compactly clustered around these points. This enables reducing the cluster spread of the recovered constellation.

- The iteration-dependent parameter $(\sigma^{(t)})^2$ quantifies the cluster spread of the recovered symbols. The initial iterations begin with a large value of $(\sigma^{(t)})^2$, allowing for flexible symbol error correction. The value of this parameter reduces with iterations, due to increasing confidence in detected symbol values, resulting in more compact clusters. This trend over the GD iterations is learnt from training data, as explained in the subsequent sub-section.

A threshold iteration value t_0 denotes the switch from the tier 1 to the tier 2 projection. Thus, the overall

two-tier projected GD update is given by

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} - \alpha^{(t)} \nabla_{\mathbf{x}}^{(t)} \tag{3.42a}$$

$$\mathbf{x}^{(t+1)} = \mathcal{P}(\hat{\mathbf{x}}^{(t+1)}) = \begin{cases} \mathcal{P}_{\text{cube}}(\hat{\mathbf{x}}^{(t+1)}), \text{ if } t \leq t_0 \\ \mathcal{P}_{\text{QAM}}(\hat{\mathbf{x}}^{(t+1)}), \text{ if } t > t_0. \end{cases} \tag{3.42b}$$

The AGD update with the two-tier projection is given by

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{d}^{(t)} - \alpha^{(t)} \nabla_{\mathbf{x}+\mathbf{d}}^{(t)} \tag{3.43a}$$

$$\mathbf{x}^{(t+1)} = \mathcal{P}(\hat{\mathbf{x}}^{(t+1)}) = \begin{cases} \mathcal{P}_{\text{cube}}(\hat{\mathbf{x}}^{(t+1)}), \text{ if } t \leq t_0 \\ \mathcal{P}_{\text{QAM}}(\hat{\mathbf{x}}^{(t+1)}), \text{ if } t > t_0 \end{cases} \tag{3.43b}$$

$$\mathbf{d}^{(t+1)} = \gamma^{(t)} (\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}). \tag{3.43c}$$

The unfolded DNN implementing the AGD algorithms with the two-tier projection is explained next.

3.5.3 Unfolded DNN implementation of projected AGD

The proposed accelerated projected one-bit network (A-PrOBNet) is illustrated in Fig. 3.8. The following present salient features for this framework.

- The T-step AGD algorithm is unfolded as a T-stage DNN, with each Stage t denoting a distinct sub-network.
- The initial inputs are provided as $\mathbf{x}^{(0)} = \mathbf{d}^{(0)} = \mathbf{0}$, empirically shown to have a well-conditioned initial gradient value to start the GD.
- Within each Stage t, the gradient is evaluated using a shallow neural network, with the two static weight matrices $\mathbf{G}$ and $\mathbf{G}^{\mathrm{T}}$ and the hidden layer nonlinearity $\zeta'(z)$. For the A-ProbNet, we implement the CDF-based likelihood and hence the element wise nonlinearity $\zeta'(z)$ is evaluated using the improved gradient method (3.20).
- The learnable parameters (denoted by the blue shaded box in Fig. 3.8) for the network are the scalers $\{\sigma^{(t)}\}_{t=t_0+1}^{T}$ for the tier 2 projection (3.41). The values of the different static parameters $\{\alpha^{(t)}, \gamma^{(t)}\}$ are chosen differently for different M-QAM constellations. This is elaborated in Sec. 4.6.

- Learning the Gaussian denoiser parameters $\{\sigma^{(t)}\}_t$ specializes each stage of the A-ProbNet to gradually reduce the cluster spread of the recovered symbols. As explained through Fig. 3.7 this has a significant effect on improving the rate of convergence.

- The A-ProbNet parameters are trained in an end-to-end manner, using the MSE loss for the ideal constellation symbols, given by

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} ||\tilde{\mathbf{x}}_n - \tilde{\mathbf{x}}_{\text{train},n}||^2. \tag{3.44}$$

We now present some finer points through means of a brief discussion on the overall projected AGD framework.

3.5.4 Discussion

Generalization of the learnt quantization-based projection

As stated earlier, the work in [20] also introduced a learnt quantization-based denoiser for M-QAM projection, utilized the two nearest neighbors. The general Gaussian denoiser for the proposed two-tier projection weights the symbol against all the constellation symbol values, adding more robustness and flexibility, especially in the initial iterations.

Loss function for end-to-end learning

The work in [36] introduced a novel regularized DNN loss function that captured both the MSE and symbol errors. This loss implicitly captured the effect of projection during DNN training. However, differently, this work does not utilize this regularized loss due to the explicit use of the projection operation. Further, the application of the loss on the final symbols with sharp Gaussian denoisers results in the MSE capturing the symbol errors exclusively. However, the use of a regularized loss is still relevant for iteration-dependent loss functions, utilizing and fine-tuning all the intermediate estimates $\{\mathbf{x}\}_{t=1}^{T}$[3]..

Generalized Gaussian denoising

Through the Gaussian denoiser introduced in this work, a single scalar parameter $\sigma^{(t)}$ per iteration t. This has the potential to be generalized further. To this end, our future work will explore this generalization, when applied to more advanced channel models.

[3] outside the scope of this work

3.6 Experimental Results

3.6.1 Simulation setup

All the different prior works for one-bit MIMO receivers (see Sec. 5.1) benchmark the algorithm for lower and higher order M-QAM constellations, i.e., QPSK and 16-QAM. However, all these approaches perform comparably for QPSK symbols. Hence, in order to show true robustness to higher order M-QAM, we perform detailed testing and benchmarking of this work for the 16-QAM constellation symbols.

The 16-QAM constellation symbols are transmitted from $K = 8$ users, $N = 128$ BS antennas with $\text{SNR} = \frac{\mathbb{E}(||\mathbf{Hx}||^2)}{\mathbb{E}(||\mathbf{n}||^2)}$ in the range 10 to 45 dB. This setup follows the standard multi-user 16-QAM simulations conducted in [1, 35, 36, 132]. The Rayleigh fading channel $\mathbf{H}$ is considered with each entry chosen from the $\mathcal{CN}(0, 1)$ distribution.

Performance benchmarks

We compare the proposed algorithm against the different model-based and learning based frameworks. *(i)* The N-ML algorithm from [35] is used to establish the original benchmark using the CDF-based likelihood. *(ii)* The OBMNet in [1] forms the original LR-based likelihood benchmark. *(iii)* The FBM-DetNet from [20] is the existing state-of-the-art benchmark, utilizing the learnt quantization-based projection to the M-QAM set.

Benchmark algorithm and network parameters

The n-ML [35] is executed for a maximum of $T = 500$ iterations, with a step size of 0.001, (to ensure convergence). Consistent with the benchmarks established in [1], the OBMNet is run for $T = 15$ iterations. The same parameters are also taken for the FBM-DetNet [20].

Improved GD, AGD and A-PrOBNet

The following are the parameters chosen for the different algorithms and networks introduced in this work in Sec. 3.3.2, 3.3.3 and 3.5.3.

- The improved GD, i.e., Algorithm 13, is run for $T = 100$ iterations, to ensure convergence of the likelihood. The step size $\alpha = 0.03$.
- The AGD, i.e., Algorithm 15, is run for $T = 20$ iterations. The momentum parameter γ is taken as 0.63 and step size $\alpha = 0.03$, based on empirical testing.

- The A-PrOBNet is run for $T = 15$ iterations. The momentum parameter $\gamma = 0.63$ and step size $\alpha = 0.03$. The denoiser parameters $\{\sigma^{(t)}\}_{t=0}^{T-1}$ are the only learnable parameters. The training for the DNN is similar to the training strategy in [36]. The network training is carried out via minibatch gradient descent, with the chosen batch size $N_{\text{train}} = 32$. In order to train the A-PrOBNet on the set of randomly generated Rayleigh channel matrices, each minibatch is generated from a different channel matrix $\mathbf{H}$, denoted by $\mathcal{B}_{\mathbf{H}}$. Based on the described system model (5.1)-(3.2), the minibatch set is generated as $\mathcal{B}_{\mathbf{H}} = \{\bar{\mathbf{x}}_n, \bar{\mathbf{n}}_n, \bar{\mathbf{y}}_n\}_{n=1}^{N_{\text{train}}}$. We utilize the MSE loss function (4.7). We practically implement minibatch gradient descent with the Adam update [144] for each training minibatch to keep a check on the learning rate. For regularization of DNN weights, we utilize weight decay to further increase resilience by preventing exploding network weights.

3.6.2 Intrinsic testing

In this sub-section the algorithms and DNNs proposed in this work are tested by varying the different parameters.

CDF-based likelihood performance

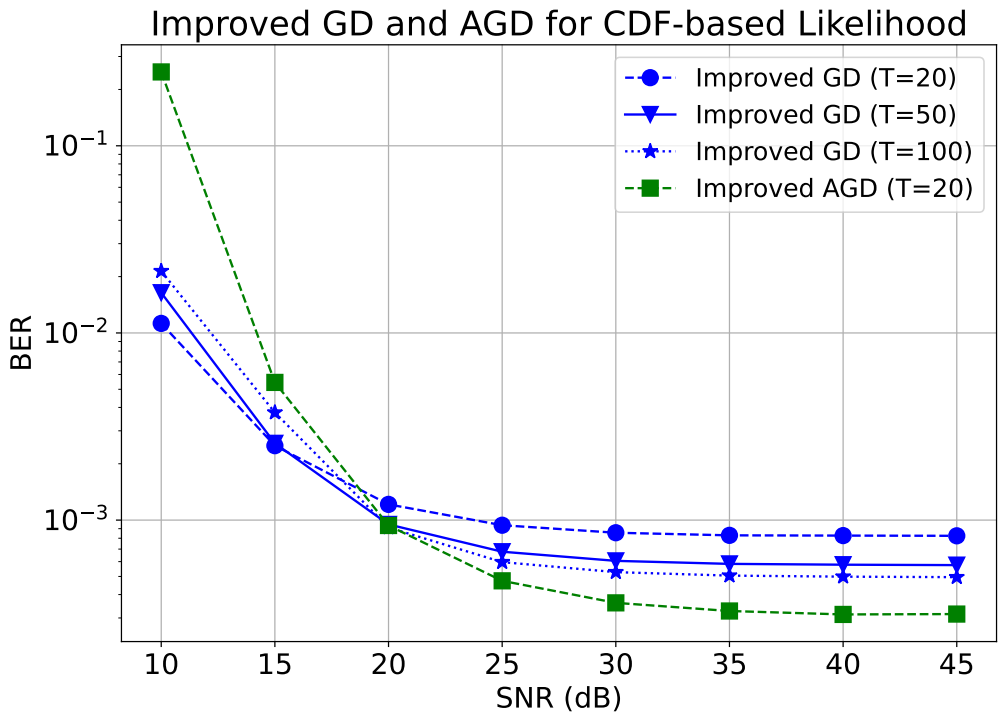


Figure 3.9: Intrinsic comparison of improved GD and AGD performance for CDF-based likelihood for given simulation setup

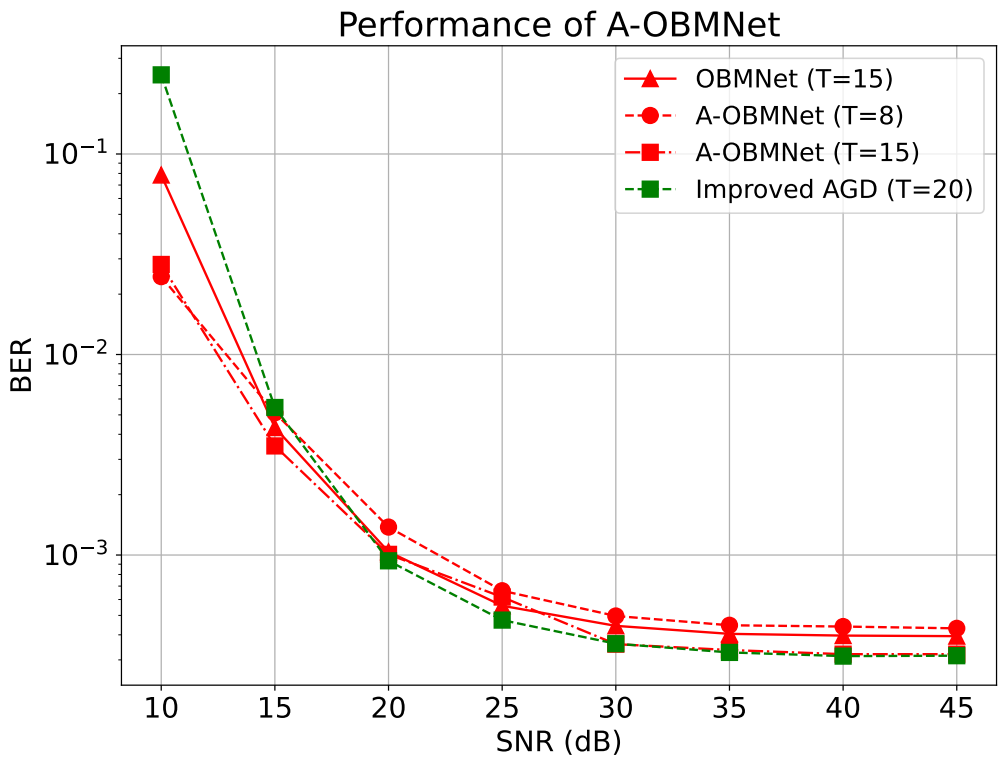


Figure 3.10: Testing the performance of AGD on surrogate likelihood using LR, i.e., (3.32) for the given simulation setup.

The performance for the improved GD and AGD, Algorithms 13 and 15 respectively, is evaluated as a function of the number of iterations in Fig 3.9. As seen from these plots, the improved GD performance saturates beyond $T = 50$ iterations. In addition, the the momentum-based GD clearly outperforms the GD, with significantly fewer iterations. The performance of both Algorithms 13 and 15 are limited due to the unit sphere normalization. Further improvement is only possible by modifying the projection step as seen in the subsequent tests.

Evaluating surrogate likelihoods

The performance of the surrogate likelihood based AGD-update steps (3.32) are given in Fig. 3.10. As can be seen by the results, the LR-based likelihood converges in a fewer number of steps using AGD (see Fig. 3.5). The BER performance for the AGD update is comparable to the GD update using half the number of iterations. This is attributed to the step size robustness for the LR-based likelihood. However, as seen by the plots, increasing the number of iterations for AGD doesn't improve BER significantly. This shows that in addition to the robustness in step size as well as the advantages of accelerated GD, projection plays a vital role in improving BER performance.

Performance of projected AGD framework

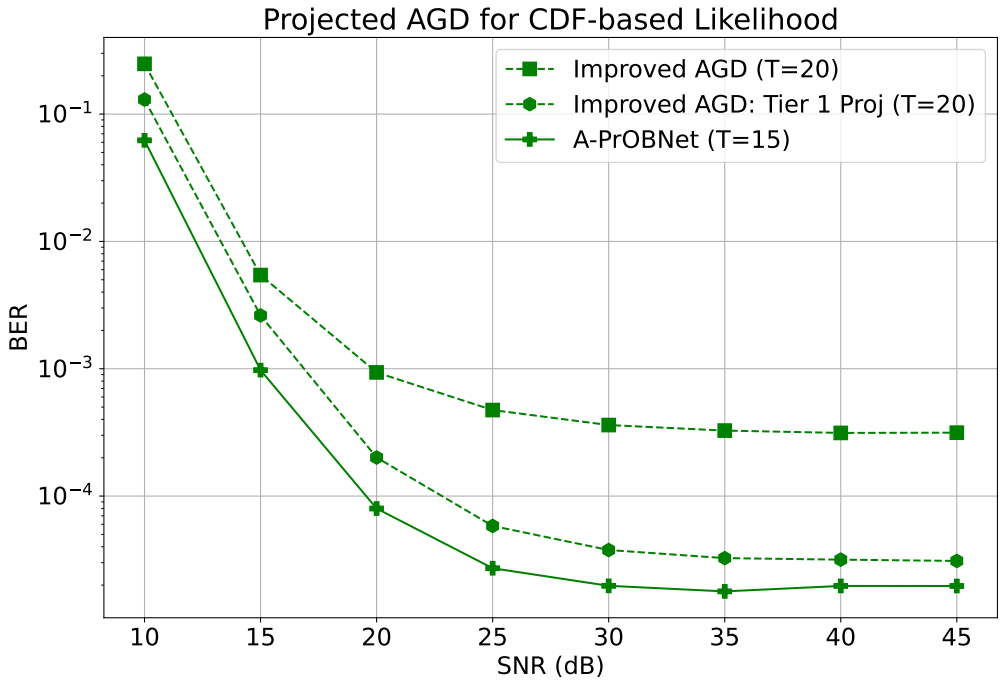


Figure 3.11: Testing the role of different projection strategies on the CDF-based AGD for given simulation setup

We evaluate the role of the different projection strategies on the better performing AGD algorithm. The role of the different projection strategies is highlighted through the results in Fig. 3.11. As seen from these plots, Tier 1 projection is a significantly better strategy compared to projection on the unit sphere. The two-tier learnt strategy of the A-PrOBNet further improves on the BER by directly reducing the cluster spread.

3.6.3 Detection for general channel

We now compare the performance of the A-PrOBNet to the state-of-the-art recovery algorithms for a general channel matrix drawn from the distribution of Rayleigh distributed channels. The recovery performance is given in Fig. 3.12. As can be seen from these plots, the performance of the proposed A-PrOBNet matches the current state-of-the-art performance of the FBM-DetNet with the same number of iterations (outperforming the OBMNet and n-ML using unit sphere normalization). However, differently, this algorithm does not make any additional approximations on the likelihood like utilization of a surrogate function. The A-PrOBNet thus establishes the limit of optimum performance for the original CDF-based likelihood without any additional approximations. Further, the two-tier projection is developed as a gener-

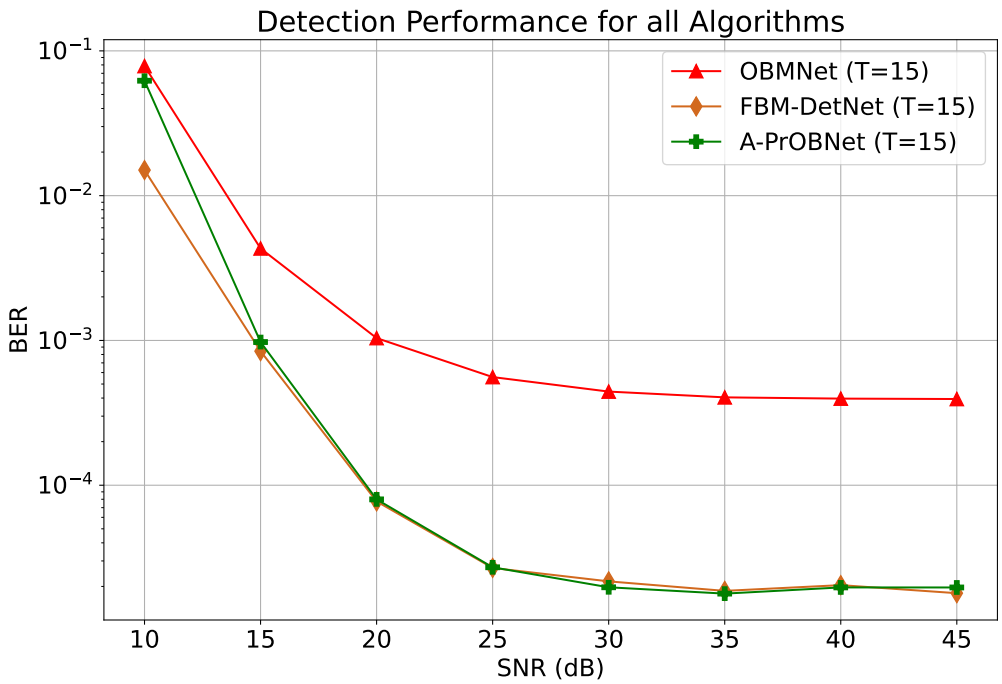


Figure 3.12: Testing state of the art detection performance of all algorithms for given simulation setup

alization of the quantization-based projection. The latter is clearly a better strategy at lower SNR values owing to weighting by a fewer M-QAM neighbors.

3.7 Conclusions

This work provides insights into the ML optimization for one-bit MIMO receivers, enabling a better understanding of the GD-based signal recovery algorithm. The accelerated GD, with faster convergence, is introduced into the class of different algorithms. These insights are extended to the surrogate likelihood function, the logistic regression, explaining the improved robustness and speed of convergence. Finally, the significance of an effective per-iteration projection step is highlighted in the GD-based recovery. The accelerated GD, with a novel two-tier projection is unfolded into a T-stage DNN, the A-PrOBNet, to achieve state of the art performance. Future work in this area involves the extension of this work to mmWave channels. The challenge of non-uniform power distribution among the different users makes joint-detection especially challenging for one-bit MIMO systems.

Chapter 3, in part, is a reprint of the material as it appears in Aditya Sant, and Bhaskar D. Rao. "Insights into Maximum Likelihood Detection for One-bit Massive MIMO Communications", IEEE Transactions on Wireless Communications, (under review). The dissertation author was the primary investigator

and author of this paper.

Appendices

3.A Proof of Theorem 2

This section provides a supplementary material to the original manuscript. In particular, this provides a proof of Theorem 1. We begin by defining a surrogate minimum and minimizer for the general likelihood and the subsequent convergence to this surrogate minimizer.

Defining the surrogate minimum and minimizer

As stated in Sec. III in the original manuscript, there is no finite $\mathbf{x} \in \mathbb{R}^{2K}$ that achieves the infimum of the likelihood, i.e., $\mathcal{L}(\mathbf{x}) > 0$. The convergence is thus analyzed to an ϵ-closeness to the infimum. A surrogate minimum ϵ is chosen to analyze the decay of the likelihood such that

$$\epsilon \leq \mathcal{L}(x^{(t)}),\ \forall\, t = 1, 2, \ldots, T. \tag{3.45}$$

We now define a structured surrogate minimizer $\mathbf{x}^*_\epsilon$ that uniquely maps to this surrogate minimum. Consider the set $\mathcal{X}_1$, defined in Sec. III-A of the original manuscript. This contains the set of all the separating hyperplanes $\mathcal{X}_{\text{plane}} \in \mathcal{X}_1$. An element $\mathbf{x}_{\text{plane}} \in \mathcal{X}_{\text{plane}}$ if

$$y_i \mathbf{h}_i^{\mathrm{T}} \mathbf{x}_{\text{plane}} \geq 0,\ \forall i = 1, 2, \ldots, 2N, \quad \text{and} \tag{3.46a}$$

$$||\mathbf{x}_{\text{plane}}|| = 1. \tag{3.46b}$$

Among the set of all the separating hyperplanes, the margin-maximizing hyperplane is uniquely defined, depending on the observed data $\{\mathbf{H}, \mathbf{y}\}$ only, as

$$\mathbf{x}_{\text{mm}} = \max_{\mathbf{x}} \min_{i} y_i \mathbf{h}_i^{\text{T}} \mathbf{x}, \quad \text{and} \tag{3.47a}$$

$$||\mathbf{x}|| = 1. \tag{3.47b}$$

Using the above margin-maximizing hyperplane, the surrogate minimizer $\mathbf{x}_\epsilon^*$ is now defined as

$$\mathbf{x}_\epsilon^* = \kappa_\epsilon \mathbf{x}_{\text{mm}}, \text{ with } \kappa_\epsilon \text{ such that} \tag{3.48a}$$

$$\mathcal{L}(\mathbf{x}_\epsilon^*) = \epsilon. \tag{3.48b}$$

We state the following lemma for uniqueness of this surrogate minimizer.

Lemma 2. *If we define two surrogate minimizers* $\mathbf{x}_1^* = \kappa_1 \mathbf{x}_{\text{mm}}$ *and* $\mathbf{x}_2^* = \kappa_2 \mathbf{x}_{\text{mm}}$*, we then have*

$$\mathcal{L}(\mathbf{x}_1^*) = \mathcal{L}(\mathbf{x}_2^*) \implies \kappa_1 = \kappa_2. \tag{3.49}$$

Proof. Let us assume there exist 2 distinct values κ_1 and κ_2, such that $\mathcal{L}(\mathbf{x}_1^*) = \mathcal{L}(\mathbf{x}_2^*)$. Further, wlog, consider $\kappa_1 > \kappa_2$. We than have

$$\begin{aligned} \mathcal{L}(\mathbf{x}_1^*) &= -\sum_{i=1}^{2N} \log \Phi(\kappa_1 y_i \mathbf{h}_i^{\text{T}} \mathbf{x}_{\text{mm}}^*) \\ &< -\sum_{i=1}^{2N} \log \Phi(\kappa_2 y_i \mathbf{h}_i^{\text{T}} \mathbf{x}_{\text{mm}}^*) \\ &< \mathcal{L}(\mathbf{x}_2^*). \end{aligned} \tag{3.50}$$

This is a contradiction and hence $\mathcal{L}(\mathbf{x}_1^*) \neq \mathcal{L}(\mathbf{x}_2^*)$ □

Using this framework, the surrogate minimizer, defined as the scaled margin-maximizing hyperplane (3.48), uniquely maps each surrogate minimum to a surrogate minimizer. Specifically, given a particular value of a ϵ, we can evaluate a unique surrogate going along the direction of the scaled margin-maximizing classifier.

Proof of Theorem 2

Based on the above definitions of the surrogate minimizers, we now analyze the convergence of the GD algorithm to the surrogate minimum.

The following quantities are defined for the given finite-horizon GD:

$$\nabla_{\max} = \max_t ||\nabla_{\mathbf{x}}^{(t)}|| \tag{3.51a}$$

$$\nabla_{\min} = \min_t ||\nabla_{\mathbf{x}}^{(t)}|| \tag{3.51b}$$

$$X_{\max} = \max_t ||\mathbf{x}^{(t)}||. \tag{3.51c}$$

We begin by analyzing the convergence of $\mathbf{x}^{(t)}$ to $\mathbf{x}_\epsilon^*$, defined in (3.48). We define $\nabla_\epsilon^* = \nabla_{\mathbf{x}}(\mathcal{L}(\mathbf{x}_\epsilon^*))$. We now that $r^{(t)} = ||\mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*||$ is monotonically decreasing over the GD iterations with step size $\alpha^{(t)} = 1/\beta$. Consider the expression for $r^{(t+1)\,2}$

$$\begin{aligned}
r^{(t+1)\,2} &= ||\mathbf{x}^{(t+1)} - \alpha^{(t)}\nabla_{\mathbf{x}}^{(t)} - \mathbf{x}_\epsilon^*||^2 \\
&= r^{(t)\,2} - 2\alpha^{(t)}\langle \mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*, \nabla_{\mathbf{x}}^{(t)}\rangle + \alpha^{(t)\,2}||\nabla_{\mathbf{x}}^{(t)}||^2 \\
&= r^{(t)\,2} - 2\alpha^{(t)}\langle \mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*, \nabla_{\mathbf{x}}^{(t)} - \nabla_\epsilon^*\rangle \\
&\quad - 2\alpha^{(t)}\langle \mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*, \nabla_\epsilon^*\rangle + \alpha^{(t)\,2}||\nabla_{\mathbf{x}}^{(t)}||^2 \\
&\leq r^{(t)\,2} - \frac{2\alpha^{(t)}}{\beta}||\nabla_{\mathbf{x}}^{(t)} - \nabla_\epsilon^*||^2 \\
&\quad - 2\alpha^{(t)}\langle \mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*, \nabla_\epsilon^*\rangle + \alpha^{(t)\,2}||\nabla_{\mathbf{x}}^{(t)}||^2 \\
&= r^{(t)\,2} - \frac{1}{\beta}\Big[\frac{1}{\beta}\Big(||\nabla_{\mathbf{x}}^{(t)}||^2 + 2||\nabla_\epsilon^*||^2 - \\
&\qquad\qquad 4\langle \nabla_{\mathbf{x}}^{(t)}, \nabla_\epsilon^*\rangle\Big) + 2\langle \mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*, \nabla_\epsilon^*\rangle\Big] \\
&\leq r^{(t)\,2} - \frac{1}{\beta}\Big[\frac{1}{\beta}\Big(||\nabla_{\min}||^2 + 2||\nabla_\epsilon^*||^2 - \\
&\qquad\qquad 4||\nabla_{\max}||\,||\nabla_\epsilon^*||\Big) + 2\langle \mathbf{x}^{(t)} - \mathbf{x}_\epsilon^*, \nabla_\epsilon^*\rangle\Big] \\
&\leq r^{(t)\,2} - \frac{1}{\beta}\beta\Big[\frac{1}{\beta}\Big(||\nabla_{\min}||^2 + 2||\nabla_\epsilon^*||^2 - \\
&\qquad\qquad 4||\nabla_{\max}||\,||\nabla_\epsilon^*||\Big) - 2(x_{\max} + \kappa_\epsilon)||\nabla_\epsilon^*||\Big] \\
&= r^{(t)\,2} - \frac{1}{\beta}\Big[\frac{2}{\beta}||\nabla_\epsilon^*||^2 - \\
&\quad \Big(\frac{4||\nabla_{\max}||}{\beta} + 2(x_{\max} + \kappa_\epsilon)\Big)||\nabla_\epsilon^*|| + \frac{1}{\beta}||\nabla_{\min}||^2\Big] \\
&= r^{(t)\,2} - \frac{1}{\beta}\,R(||\nabla_\epsilon^*||)
\end{aligned} \tag{3.52}$$

Here, the different inequalities are derived using the properties of β-smooth functions[4] and the Cauchy-

[4]Nesterov, Y., 2003. Introductory lectures on convex optimization: A basic course (Vol. 87). Springer Science & Business Media.

Schwarz inequality. Since $R(||\nabla^*_\epsilon||) \to \frac{1}{\beta}||\nabla_{\min}||^2$ as $||\nabla^*_\epsilon|| \to 0$, for a small enough value of ϵ, we can consider $0 < R(||\nabla^*_\epsilon||) < \frac{1}{\beta}||\nabla_{\min}||^2$. Thus the sequence $r^{(t)}$ is monotonically decreasing. Using evaluate the bound on $\triangle^{(t)} = \mathcal{L}(\mathbf{x}^{(t)}) - \mathcal{L}(\mathbf{x}^*_\epsilon)$.

Using convexity of $\mathcal{L}(\mathbf{x}^{(t)})$, we have $\mathcal{L}(\mathbf{x}^*_\epsilon) \geq \mathcal{L}(\mathbf{x}^{(t)}) - \langle \nabla^{(t)}_{\mathbf{x}}, \mathbf{x}^{(t)} - \mathbf{x}^*_\epsilon \rangle$. Thus we can bound $\triangle^{(t)}$ as

$$\begin{aligned} \triangle^{(t)} &\leq \langle \nabla^{(t)}_{\mathbf{x}}, \mathbf{x}^{(t)} - \mathbf{x}^*_\epsilon \rangle \\ &\leq ||\nabla^{(t)}_{\mathbf{x}}||\, ||\mathbf{x}^{(t)} - \mathbf{x}^*_\epsilon|| \\ &\leq ||\nabla^{(t)}_{\mathbf{x}}||\, r_0 \end{aligned} \tag{3.53}$$

For the β-smooth function

$$\mathcal{L}(\mathbf{x}^{(t+1)}) \leq \mathcal{L}(\mathbf{x}^{(t)}) - \frac{1}{2\beta}||\nabla^{(t)}_{\mathbf{x}}||^2. \tag{3.54}$$

Subtracting $\mathcal{L}(\mathbf{x}^*_\epsilon)$ and using (3.53)

$$\begin{aligned} \triangle^{(t+1)} &\leq \triangle^{(t)} - \frac{1}{2\beta}||\nabla^{(t)}_{\mathbf{x}}||^2 \\ \triangle^{(t+1)} &\leq \triangle^{(t)} - \frac{1}{2\beta}\frac{\triangle^{(t)\,2}}{r_0^2} \\ \frac{1}{\triangle^{(t)}} &\leq \frac{1}{\triangle^{(t+1)}} - \frac{1}{2\beta\, r_0^2}\frac{\triangle^{(t)}}{\triangle^{(t+1)}} \\ \frac{1}{\triangle^{(t)}} &\leq \frac{1}{\triangle^{(t+1)}} - \frac{1}{2\beta\, r_0^2} \end{aligned} \tag{3.55}$$

Adding the T terms of the telescoping summation gives us

$$\mathcal{L}(\mathbf{x}^{(T)}) - \mathcal{L}(\mathbf{x}^*_\epsilon) \leq \frac{2\beta r_0^2 \triangle^{(0)}}{\triangle^{(0)} T + 2\beta r_0^2}, \tag{3.56}$$

which is the required form of the decay rate.

The above theorem provides the best possible convergence rate for GD, utilizing the optimally chose step size, i.e., $\alpha^{(t)} = 1/\beta$. Further, for $\epsilon \approx 0$, the theorem provides the framework to choose of the number of iterations T required to get arbitrarily close to the infimum of the likelihood.

Chapter 4

DNN-aided One-bit MIMO Detection

4.1 Introduction

Next generation massive MIMO communication system design promises high-speed wireless communication and an entire network of interconnected devices [3, 4]. However, widescale deployment brings in challenges for system cost, power consumption and complexity. Several advances in model-based algorithm design as well as high performance DNNs are being made to combat these challenges for both channel estimation as well as end-to-end communication. The general parametric structure of DNNs, coupled with their advantage as universal functional approximators [121, 122], makes these an integral part of the future of robust wireless communication, exploited for a variety of applications from beamformer design [52–54], channel estimation [123–125] as well as end-to-end detection [55–59].

One of the major challenges for widescale deployment is the design of high-resolution analog to digital converters (ADCs). Prior analysis of system design has shown that high-resolution ADCs account for significant system cost and power consumption [23, 120]. Moving in the direction of low cost and complexity, low-resolution ADCs have been gaining significant interest, due to advances in both signal processing and DNN-based algorithms [20–25]. A special case of low-resolution ADCs is the one-bit ADC. One-bit signal recovery has seen various innovations in general signal processing research [145–148]. In our work, we focus on the application of DNN-based methods to symbol recovery for one-bit massive MIMO communication systems. DNN based detectors appear to be naturally suited for this problem because of the inherent nonlinearity in the measurement process.

One-bit MIMO data detection benefited significantly with the application of Bussgang's theroem to

linearize the input-output relation [126]. Based on this linearization, a large class of linear receivers as well as MMSE receivers has been proposed for both single carrier and multi-carrier systems [29–31]. In addition several works utilize this linearization to characterize the one-bit system and evaluate the overall system performance and capacity [127–129]. Additional robust model-based detectors improving on the Bussgang linear detectors have also been proposed in several key works [32, 33]. In addition to one-bit data detection, one bit channel estimation for mmWave communication systems has also been studied [26, 27]. Our previous work [28] characterizes the subspace of the one-bit transformed signal and even generalizes this behavior to a broader class of odd-symmetric nonlinearities. In addition to the different model-based approaches, different works applying DNNs to one-bit detection have also been proposed [133–137, 149, 150].

One of the most resilient class of one-bit detectors is based on the one-bit likelihood maximization of the received signal using the Gaussian cumulative distribution function (cdf) [34]. The work in [35] introduced a near maximum likelihood (n-ML) detector based on a two step iterative algorithm - gradient descent (GD) followed by projection onto the unit sphere. Other works applying the Gaussian cdf likelihood formulation have also been used extending this idea [130,131]. However, one of the limitations of applying the GD iteration on the Gaussian cdf is its instability at high signal-to-noise ratio (SNR) values [132]. The work in [1] applied the sigmoid approximation of the Gaussian cdf [143] to the one-bit likelihood. The ensuing detector, that the authors named the OBMNet, formulated this detection as an unfolded DNN, learning the GD step sizes at each iteration. The sigmoid approximation was shown empirically to stabilize the gradient and addressed more explicitly in [132]. This work is, at present, the current state-of-the-art for M-QAM data detection.

Our work builds on the state-of-the-art OBMNet formulation, with the following contributions.

- We introduce a novel, regularized GD approach for one bit detection. We augment each GD iteration with a learnable DNN-based step. This DNN-based step performs an explicit regularization of each GD iteration of the OBMNet algorithm, enhancing recovery for data symbols transmitted from an M-QAM constellation. We capitalize on not only the model-based OBMNet structure, but also increase the network expressivity through this DNN-aided regularization block per iteration.

- We improve on the generalization capability of existing end-to-end detection networks (mentioned earlier), which are trained and tested on a single channel response. By designing the architecture, input data as well as training on on multiple randomly sampled Rayleigh-fading channels, we avoid the need to re-train the detector network for each different channel state information matrix.

- We implement two unique networks, for the above mentioned regularized GD approach:

1. ROBNet: A deep unfolded network with an identical, *different*, sub-network block per GD iteration
2. OBiRIM: A deep recurrent neural network utilizing estimation memory for one-bit estimation

To the best of our knowledge, the latter, i.e. the OBiRIM, presents the first approach using a recurrent neural network for one-bit detection.

- Contrary to the mean square error (MSE) loss used for network training, we introduce a novel loss function, tailored to MIMO communication symbol recovery. In particular, we incorporate a constellation-aware regularized MSE loss function to penalize the symbol errors as well as the bit errors. We envision this as a general communication system loss function, not just limited to one-bit symbol recovery.

Our experimental results, implemented on the i.i.d. Rayleigh fading channel, show the utility of considering a robust regularized GD algorithm through sharper and more compact recovered constellation clusters with significantly reduced cluster spread. This improved recovery is especially significant for improved detection performance of higher order M-QAM constellations. Although the analysis of multi-bit MIMO receivers falls outside the scope of this work, the presented regularized GD framework and robust constellation aware DNN loss function can potentially be applied to deal with the nonlinearities of these systems as well.

The purpose of this document is provide background and details necessary for the mmWave extension presented at IEEE ICASSP 2023.

Organization: This manuscript is organized as follows - Sec. 4.2 introduces the system model, one-bit detection problem and the gradient-descent based approaches used. Sec. 4.3 introduces our proposed framework for general regularized one-bit detection, while Sec. 4.4 explains the specific DNN implementation used. Sec. 4.6 provides experimental validation of our proposed framework and Sec. 5.6 concludes the manuscript.

Notation: We use lower-case boldface letters $\mathbf{a}$ and upper case boldface letters $\mathbf{A}$ to denote complex valued vectors and matrices respectively. The notation $\mathfrak{Re}(\cdot)$ and $\mathfrak{Im}(\cdot)$ denote the real and imaginary parts, respectively. The operation $(\cdot)^{\mathrm{T}}$ denotes the transpose of the array or matrix. Unless otherwise specified, all scalar functions like $\tanh(\cdot)$ or $\mathrm{sign}(\cdot)$, when applied to arrays or matrices, imply element-wise operation. The notation $\mathbf{x}^{(t)}$ is used to denote the value of the variable $\mathbf{x}$ at iteration t of the algorithm. For the DNN training, the size of the training set is given by N_{train} and the notation $\hat{\mathbf{x}}_{n,\text{train}}$ denotes the n^{th} sample from this set. Unless otherwise specified, the norm $||\cdot||$ represents the ℓ_2-norm for a vector and Frobenius norm for a matrix.

4.2 System model and background

The one-bit MIMO model follows from the same model introduced in the equivalent section in Chapter 3. However, to recap, the system model is repeated here in brief. It is assumed that all the signals are converted to the equivalent real-valued forms using the framework in (3.3).

The uplink unquantized signal is received as $\mathbf{r}$ and the one-bit quantized signal is given by $\mathbf{y}$. These quantities are given as

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{z} \tag{4.1a}$$

$$\mathbf{y} = \text{sign}(r), \tag{4.1b}$$

where $\mathbf{H}$ is the MIMO channel matrix, $\mathbf{x}$ is the vector of transmitted M-QAM symbols and $\mathbf{z}$ is the AWCGN. The following two sub-sections briefly recap the approaches for the GD-based detection for one-bit MIMO.

4.2.1 One-bit maximum likelihood and GD-based detection

The one-bit maximum likelihood (ML) problem has been derived in [34] as

$$\hat{\mathbf{x}}_{\text{ML}} = \underset{\mathbf{x}\in\mathcal{M}^{2K}}{\text{argmax}} \sum_{i=1}^{2N} \log \Phi\big(\sqrt{2\rho}\, y_i \mathbf{h}_i^{\text{T}} \mathbf{x}\big), \tag{4.2}$$

where $\Phi(\cdot)$ is the cumulative distribution function (cdf) for $\mathcal{N}(0,1)$ and $\mathcal{M}^{2K}$ represents the set of the $2K$-dimensional vectors, consisting of the real-valued representation (see eq. (3.3)) of the K-dimensional vectors of M-QAM constellation symbols. The search over this constrained, finite, non-convex set $\mathcal{M}^{2K}$ scales this problem exponentially in the number of users. Different approaches based on relaxations of the optimization (4.2) have been proposed [1, 35, 133].

One of the proposed relaxations for the constrained optimization (4.2) involves unconstrained GD over the entire subspace $\mathbb{R}^{2K}$, followed by a projection onto the subspace of interest [35]. The unconstrained GD update step has been derived in [35] as

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \alpha^{(t)} \sqrt{2\rho}\, \mathbf{G}^{\text{T}} \frac{\phi(\sqrt{2\rho}\, \mathbf{G}\mathbf{x})}{\Phi(\sqrt{2\rho}\, \mathbf{G}\mathbf{x})} \tag{4.3}$$

where $\alpha^{(t)}$ is the step size at iteration t, $\mathbf{G} = \text{diag}(y_1, y_2, \ldots, y_{2N})\, \mathbf{H}$ and $\phi(\cdot)$ is the Gaussian probability density function. The subsequent step projects this estimate $\mathbf{x}^{(t+1)}$ onto the unit hyper-sphere.

This optimization approach is limited by the behavior of the Gaussian cdf $\Phi(\cdot)$ at high SNR values.

It is empirically observed that this function drops rapidly to zero at high SNR values, making the likelihood gradient explode to large values. Further the Hessian matrix for the same is empirically observed to contain a high condition number [132]. All this makes the optimization (4.2) unstable at high SNR values.

4.2.2 Current state-of-the-art one-bit detector: OBMNet

An approximate ML estimation framework was proposed in [1] using the logistic cdf approximation of the Gaussian cdf [143]. This approximation involves sigmoids, a popular activation function in neural networks, and naturally leads to a DNN based detector. The authors in [1] empirically observe a robustness in detection to incorrect symbol estimation as well as imperfect CSI at the detector as a result of this approximation. This can be explained by examining the gradient of the approximate ML and noting that it is much better behaved at high SNR [132]. The approximate ML problem using the sigmoid log-likelihood is given by

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \underset{\mathbf{x}\in\mathcal{M}^{2K}}{\operatorname{argmin}} \sum_{i=1}^{2N} \log\left(1 + e^{-c\sqrt{2\rho}y_i \mathbf{h}_i^{\mathrm{T}}\mathbf{x}}\right), \tag{4.4}$$

with the value of $c = 1.702$. Applying GD to the likelihood (4.4), we have the update equation

$$\begin{aligned} \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} - \alpha^{(t)}\nabla_{\mathbf{x}}^{(t)} \\ &= \mathbf{x}^{(t)} + \alpha^{(t)}\mathbf{G}^{\mathrm{T}}\sigma(-\mathbf{G}\mathbf{x}^{(t)}), \quad t = 0, \ldots, T-1, \end{aligned} \tag{4.5}$$

where $\sigma(\cdot)$ is the logistic sigmoid function. The constants have been absorbed into the step size $\alpha^{(t)}$. After executing T iterations of GD, the final estimate $\hat{\mathbf{x}}^{(T)}$ is normalized as

$$\tilde{\mathbf{x}} = \frac{\sqrt{K}}{||\mathbf{x}^{(T)}||}\hat{\mathbf{x}}^{(T)}. \tag{4.6}$$

The T-step unconstrained update (4.5) is implemented as a T-layer unfolded DNN with sigmoid nonlinearity and network weights depending on the CSI matrix and one-bit measurements, i.e., the OBMNet [1]. The step sizes at each iteration $\alpha^{(t)}$ are the only learnable parameters. The network parameters are trained on the MSE loss function

$$\mathcal{L} = \frac{1}{N_{\mathrm{train}}} \sum_{n=1}^{N_{\mathrm{train}}} ||\tilde{\mathbf{x}}_n - \tilde{\mathbf{x}}_{\mathrm{train},n}||^2. \tag{4.7}$$

The results in [1] show the OBMNet as an efficient low-complexity detector for QPSK as well as 16-QAM symbols. However this detector has a few limitations, described below.

(i) *Limited network expressivity:* The OBMNet, as a general DNN, is highly underparameterized. Any

changes in the network architecture, loss function and training procedure do not show up in improved performance for the network.

(ii) *Constellation cluster spread:* Analyzing scatter plots for the recovered symbols, high cluster spread is evident (see Fig. 4.8). More on this in the next section. Although this does not compromise bit error rate (BER) for lower order constellations like QPSK, it degrades performance at higher order constellations like 16-QAM.

(iii) *Gap to ML:* The original two-step OBMNet detection (4.5)-(4.6) falls short of the theoretical exponential search based ML solution to (4.4). The authors in [1] fine-tune their estimates through a constrained lower order ML search step to bridge this gap.

4.3 Regularized GD for one-bit MIMO detection

In order to address some of the observed limitations of the OBMNet, we introduce the framework of regularized neural one-bit detection, building on the OBMNet framework (Sec. 4.2.2). The specific network structure and implementation details for our approach are provided in the next section. Here, we begin with the general regularized GD framework, with a learnable DNN-aided regularization. Next, for robust DNN training, we have developed a novel constellation-aware quantization based loss function. Finally we comment on the ability to generalize to any arbitrary Rayleigh fading channel.

4.3.1 DNN-aided regularized GD for one-bit MIMO detection

In order to improve the detection robustness, we modify the unconstrained OBMNet update step (4.5) to a regularized GD update, per iteration t, given by

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} - \alpha^{(t)} \nabla_{\mathbf{x}}^{(t)} \tag{4.8a}$$

$$\mathbf{x}^{(t+1)} = \hat{\mathbf{x}}^{(t+1)} + h_{\phi}^{(t)}(\mathbf{x}^{(t)}, \nabla_{\mathbf{x}}^{(t)}, \hat{\mathbf{x}}^{(t+1)}). \tag{4.8b}$$

Here, the first step (4.8a), the intermediate update, is the same as the unconstrained OBMNet update (4.5). The second step (4.8b) represents the introduced correction to this unconstrained update. Based on regularizing the estimate $\hat{\mathbf{x}}^{(t+1)}$ to account for optimization within the M-QAM constellation space, the overall update (4.8) is called the regularized GD detection for one-bit MIMO. We introduce a parametric regularization function $h_{\phi}^{(t)}(\cdot)$, per iteration, implemented via a DNN (exact implementation in Sec. 4.4). By

means of an additional learnable regularization we increase the network expressivity of the original OBMNet, by increasing the number of learnable network parameters. We also enable per-iteration projection of the iterand $\mathbf{x}^{(t)}$ onto the set of the real-valued representation of M-QAM constellation points $\mathcal{M}^{2K}$.

The detector FBM-DetNet, introduced in [20], also implements a per-iteration projection of the OBMNet estimate on the $\mathcal{M}^{2K}$ subspace at each iteration. This is implemented using a learnable hard quantization of each iterand $\hat{\mathbf{x}}^{(t+1)}$ to the M-QAM constellation. Differently, the regularized GD (4.8) learns a general projection function, implemented as a residual correction at each step.

4.3.2 Improved DNN loss function

In order to capitalize on the general parametric regularization structure, we design a constellation-aware loss function. The MSE loss function (4.7), utilized by the OBMNet, penalizes the magnitude of the symbol error for the received signal. We attempt to add in an additional robustness to network training by also penalizing symbol flips in the estimated symbols, thus implicitly penalizing bit flips in the recovered data.

Incorporating this robustness, we improve on the MSE loss by using the following modification

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} \big[||\mathbf{x}_n^{(T)} - \tilde{\mathbf{x}}_{\text{train},n}||^2 + \lambda \, \mathcal{R}(\mathbf{x}_n^{(T)}, \tilde{\mathbf{x}}_{\text{train},n}) \big], \tag{4.9}$$

where $\mathcal{R}(\cdot)$ is a constellation-aware regularization for DNN training. This regularization is based on a smooth quantization of the network output, and implemented as

$$\mathcal{R}(\mathbf{x}_n^{(T)}, \tilde{\mathbf{x}}_{\text{train},n}) = ||\mathcal{Q}_\beta(\mathbf{x}_n^{(T)}) - \tilde{\mathbf{x}}_{\text{train},n}||^2. \tag{4.10}$$

Here, the function $\mathcal{Q}_\beta(\cdot)$ is a smooth constellation-aware quantization function, utilizing the nonlinearity $f_\beta(z) = \tanh(\beta z)$ with a hyperparameter β. The choice of the scaled $\tanh(\cdot)$ nonlinearity is inspired by *(i)* The saturating behavior for quantization, *(ii)* Differentiability for backpropagation of the loss, and *(iii)* Ease of tuning to regulate the quantization degree. For the two considered constellations in this work, we implement the quantization function $\mathcal{Q}_\beta(\cdot)$ as follows.

1. $\mathcal{Q}_\beta(x)$ for QPSK constellation:

$$\mathcal{Q}_\beta(x) = \tanh(\beta \, x). \tag{4.11}$$

2. $\mathcal{Q}_\beta(x)$ for 16-QAM constellation:

$$\mathcal{Q}_\beta(x) = \tanh(\beta\,(x+2)) + \tanh(\beta\,x) + \tanh(\beta\,(x-2)). \tag{4.12}$$

The quantization function (4.12) for the 16-QAM constellation is plotted in Fig. 4.1. The plots illustrate that the quantization (4.12) implements a smooth version of the symbol-mapper, that can be backpropogated through the regularization network, to the 16-QAM constellation symbols. By specifically modifying the loss function as (4.19) to communication system symbol recovery, we are able to incorporate a symbol error rate (SER) metric into the training phase of our networks. The role of the quantizer (4.20) is to cluster the estimated symbols within a very small neighborhood of the nearest M-QAM symbols. Thus the symbol loss for the staying within the "right" symbol boundaries is attenuated and the symbol loss for crossing over the symbol boundaries is amplified. Thus the regularization loss will be dominated by symbol errors (implicitly bit errors). By incorporating this into the training phase, we also account for an improved BER performance, a metric that is imperative to communication system design.

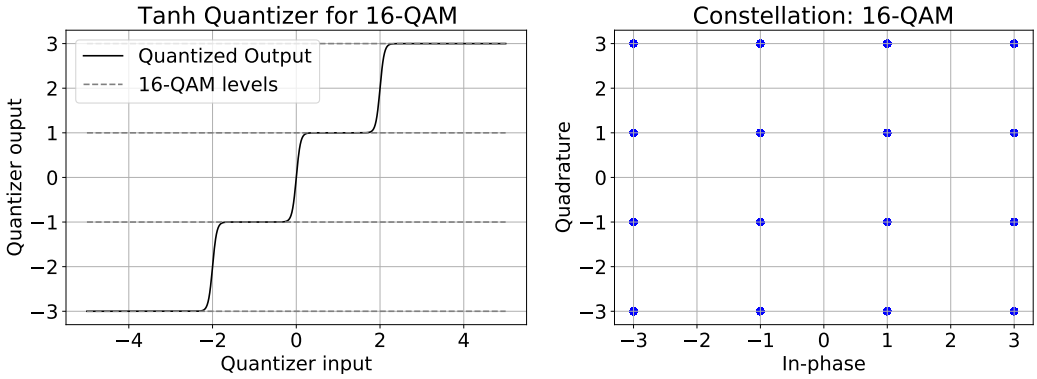


Figure 4.1: Illustration for 16-QAM quantizer (4.12). The value of $\beta = 10$.

Remark. *The general quantization function $\mathcal{Q}_\beta(\cdot)$ can be implemented using different nonlinearites, like the ReLU. Analysis of such alternate quantization functions will extract larger patterns in the behavior of the constellation-aware regularization. This work conceptually introduces improved loss functions through the use of M-QAM constellation mapping, in order to incorporate symbol error rate in the network training. The specific detailed analysis of alternate loss functions falls outside the scope of this work.*

Before illustrating our specific implementations of this neural detector, we present another advantage of this framework, i.e., the generalizability to arbitrary Rayleigh fading channels.

4.3.3 Generalization of one-bit neural detection

Contrary to the conventional end-to-end learning approaches for one-bit detection [133–135], we develop our regularized model beyond a channel-specific detector. The regularization network $h_{\phi}^{(t)}(\cdot)$ in (4.8) implicitly takes in the channel information via the gradient of the estimate at each iteration $\nabla_{\mathbf{x}}^{(t)}$, which is used for the signal recovery. Each new subsequent channel matrix $\mathbf{H}$ results in a new sequence of gradient expressions for the unfolded network, i.e., $\nabla_{\mathbf{x}}^{(t)}$. This, in turn, enables the network to uniquely identify the inputs with the channel response. As opposed to directly feeding in the input channel matrix $\mathbf{H}$ to the regularization network, our approach exploits the main advantage of unfolded deep learning [50] by using the channel information in an appropriate, model-based form. By feeding the gradient of the signal, for any generated channel matrix $\mathbf{H}$, to the regularization network, we are able to efficiently fine tune the original GD algorithm for that particular channel matrix $\mathbf{H}$. This approach of learning a parametric regularization from the gradient of the linear model was also used for recurrent inference machines (RIMs) [151]. We thus overcome the need to re-train or fine-tune the network for each unique channel matrix $\mathbf{H}$. This enables completely eliminating the need to transmit any other additional pilot symbols (for any online training) following the initial access and channel state information (CSI) estimation phase.

4.4 DNN-Aided Regularized GD: Implementation

Based on the proposed regularized GD and loss function framework introduced earlier, the next two subsections present the specific implementation via two distinct approaches, namely, the unfolded ROBNet and the recurrent OBiRIM.

4.4.1 Unfolded one-bit DNN: ROBNet

Model-based algorithm unrolling and the use of unfolded DNNs have been explored in different applications of signal processing and wireless communication [50, 152]. These networks are able to account for any model mismatch and can significantly save on the number of iterations, compared to the original model-based algorithms. The ability to use such network structures to complement model-based analysis motivates us to incorporate such an unfolded DNN to implement our regularized one-bit GD approach (4.8).

Our proposed unfolded network implementation, the regularized one-bit network (ROBNet) is illustrated in Fig. 4.6. Based on this, we present the following salient features of the unfolded learning approach.

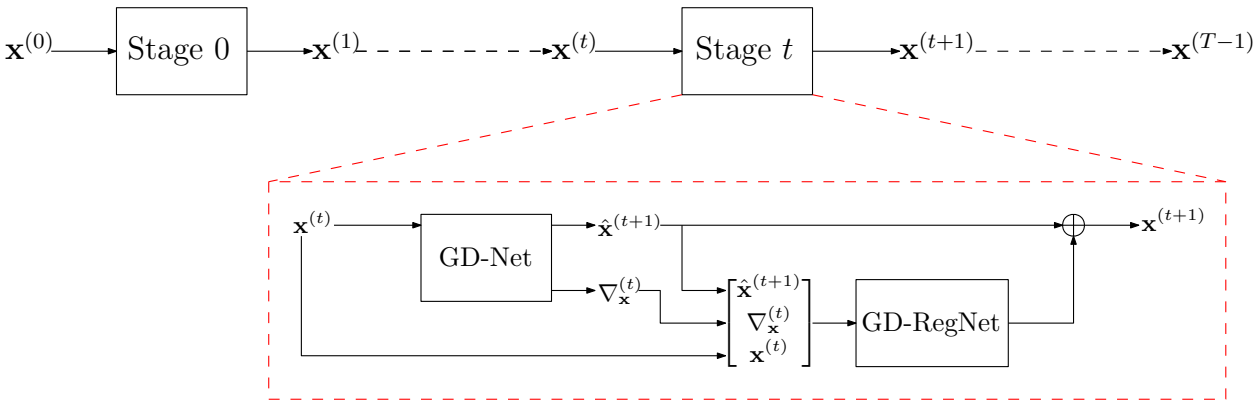


Figure 4.2: Block diagram for the Regularized One-bit Detector (ROBNet)

- The ROBNet, implementing a T-stage regularized GD algorithm, is unfolded into T distinct sub-networks (each represented as Stage t in Fig. 4.6). Each sub-network at Stage t consists of two sequential phases.

  1. GD-Net - Identical to each OBMNet [1] iteration, this implements (4.8a), with the t^{th} gradient and unconstrained iterate given by $\nabla_{\mathbf{x}}^{(t)}$ and $\hat{\mathbf{x}}^{(t+1)}$, respectively. The GD step size $\alpha^{(t)}$ is the only learnable parameter.

  2. GD-RegNet - Denoted by $h_{\phi}^{(t)}(\hat{\mathbf{x}}^{(t+1)}, \nabla_{\mathbf{x}}^{(t)}, \mathbf{x}^{(t)})$, this is a larger parametric network that regularizes each GD iteration, i.e., (4.8b). This increases network exressivity through a larger number of learnable parameters.

- Additionally, for each Stage t, a residual link from the GD-Net output $\hat{\mathbf{x}}^{(t)}$ is fed to the output of the GD-RegNet. Thus the role of each GD-RegNet at Stage t is to impart an appropriate stage-dependent correction, learnt from the data, to the unconstrained gradient step.

We now provide the specific technical details of this GD regularization, along with the general channel training.

GD-RegNet structure and training: We begin by describing the input to the GD-RegNet at each Stage t, consisting of the GD-Net output - the unconstrained update $\hat{\mathbf{x}}^{(t+1)}$, gradient $\nabla_{\mathbf{x}}^{(t)}$ and previous iterand $\mathbf{x}^{(t)}$. These three components are converted into 6 channels, with two channels, per component, for the real and imaginary parts, respectively. This is propagated through the GD-RegNet as follows:

1. First a 1-D convolution extracts the input features into a set of output channels[1]

[1]The 1-D convolution empirically shown to provide improved results, compared to only using fully connected layers. Feature extraction from the OBMNet estimate and gradient enables a more robust GD regularization (4.8b).

2. The output of the 1-D convolution is flattened and passed through a fully connected network (FCN), consisting of three hidden layers. The output of the FCN is a vector in $\mathbb{R}^{2K}$, same as the $\hat{\mathbf{x}}^{(t+1)}$

3. A residual link from the OBMNet output $\hat{\mathbf{x}}^{(t+1)}$ is added at the output of the GD-RegNet, generating the final iterand $\mathbf{x}^{(t+1)}$.

4. We normalize the final output $\mathbf{x}^{(T)}$, analogous to (4.6), as

$$\mathbf{x}^{(T)} \leftarrow \eta_M \frac{\mathbf{x}^{(T)}}{||\mathbf{x}^{(T)}||}, \tag{4.13}$$

where η_M depends on the constellation order M[2].

The specific details of the parameters in each layer, for a general number of users K, are given in Table 4.2.

The network training is carried out via minibatch gradient descent, with the chosen batch size $N_{\text{train}} = 32$. In order to train the ROBNet on the set of randomly generated Rayleigh channel matrices, each minibatch is generated from a different channel matrix $\mathbf{H}$, denoted by $\mathcal{B}_{\mathbf{H}}$. Based on the described system model (5.1)-(3.2), the minibatch set is generated as $\mathcal{B}_{\mathbf{H}} = \{\bar{\mathbf{x}}_n, \bar{\mathbf{z}}_n, \bar{\mathbf{y}}_n\}_{n=1}^{N_{\text{train}}}$. We employ the modified loss function (4.19), discussed in Sec. 4.3.2, to train the ROBNet. We practically implement minibatch gradient descent with the Adam update [144] for each training minibatch to keep a check on the learning rate. For regularization of DNN weights, we utilize weight decay to further increase resilience by preventing exploding network weights.

4.4.2 Recurrent one-bit DNN: OBiRIM

We now investigate an alternate network strategy that can model the sequence of iterands $\{\mathbf{x}^{(t)}\}_{t=0}^{T}$ as a time-series generated via the regularized GD algorithm. We thus turn towards recurrent neural networks to learn this time series pattern, resulting in a parametrically efficient network design.

Recurrent neural networks (RNNs) have been one of the earliest DNNs to incorporate time-series information in pattern extraction for applications like speech and NLP [153, 154]. A specific class of these networks, the recurrent inference machines (RIMs), proposed in [151], have shown much success in medical imaging. The ability of the RIM architecture to parametrically model a prior distribution and as well as the optimization procedure is responsible for its superior performance over conventional approaches [46, 151].

[2]For lower order constellations, i.e., QPSK, we incorporate η_M into the learning process during training, making it data-dependent. However, we have empirically observed that for higher order QAM, i.e., 16-QAM, this value should be fixed. On the whole, the difference between statically choosing η_M and learning it from the data does not have any change in overall performance.

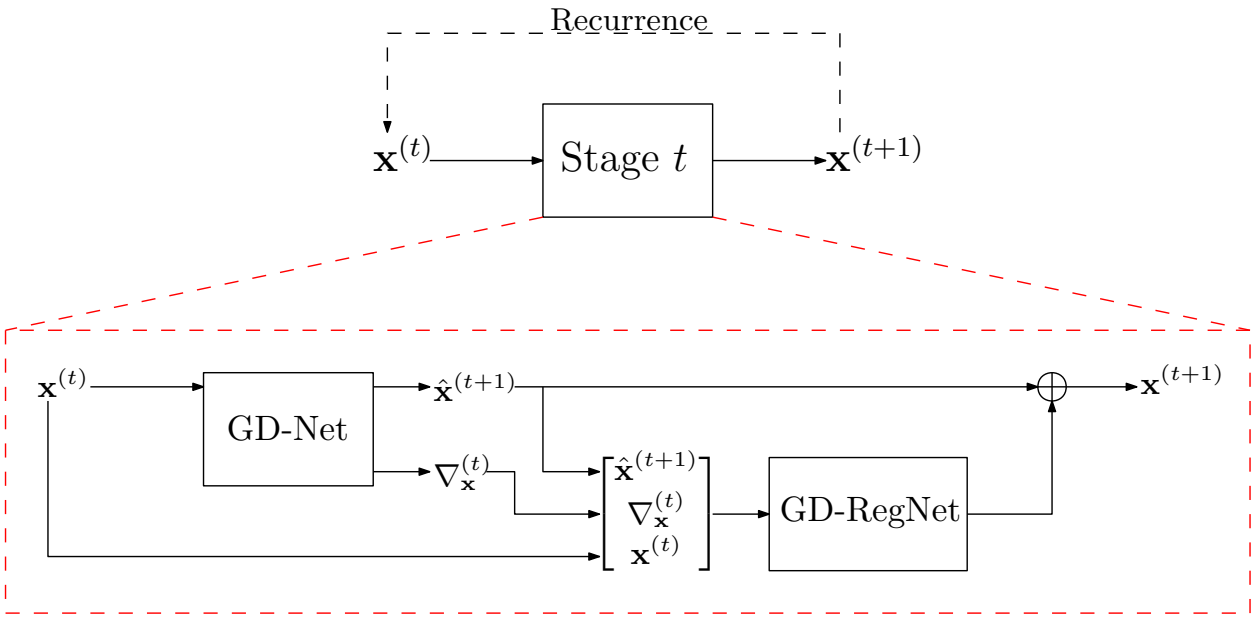


Figure 4.3: Block diagram for the Projected-Regularized One-bit Recurrent Inference Machine (OBiRIM) Detector

The use of a parametric prior distribution as regularization, along with an iterative estimation algorithm of one-bit detection fits in perfectly with the strengths of the RIM framework.

To this end, we implement our own version, the one-bit RIM (OBiRIM) for the regularized GD algorithm (4.8). The overall network structure for the OBiRIM is shown in Fig. 4.3. Different from the ROBNet, the OBiRIM utilizes parameter sharing for the GD-RegNet, such that the same set of parameters h_ϕ are used for each GD iteration in (4.8). The presence of recurrent cells in the OBiRIM, stores the relevant estimation memory for the iterative GD algorithm and fine-tunes each OBMNet estimate $\hat{\mathbf{x}}^{(t)}$ based on the system history. By sharing parameters among the different iterations and exploiting the system memory, this network is highly parameter efficient. To the best of our knowledge, the OBiRIM is the first recurrent DNN for detection of one-bit MIMO, which can be generalized for detection to any arbitrary Rayleigh fading channel.

The overall regularized GD framework for the OBiRIM, as seen in Fig. 4.3, is similar to the ROBNet, i.e., implementation of the regularized GD algorithm (4.8). We highlight some of the salient features of this network here below.

- The OBiRIM, implementing a T-step regularized GD algorithm, consists of T temporal iterations. At each Stage t (see Fig. 4.3), the data is sequentially processed through two phases.

  1. GD-Net - This is similar to GD-Net block in the ROBNet (see Fig. 4.6).
  2. GD-RegNet - Different from the equivalent network of the ROBNet, this GD-RegNet incorporates

DNN recurrence to temporally fine tune each estimate $\mathbf{x}^{(t)}$. DNN memory enables temporal processing, while sharing parameters across different OBiRIM stages.

- At each Stage t, a residual link from the GD-Net output $\hat{\mathbf{x}}^{(t)}$ is fed to the output of the GD-RegNet, thus imparting a stage-dependent correction to the unconstrained gradient step.

We now provide the technical parameters of this GD regularization, along with the general channel training.

Table 4.1: DNN Parameters of GD-RegNet in ROBNet & OBiRIM (K Users)

Network	**Layer**	**Parameters**
ROBNet (each stage)	*Convolution (1-D)*	conv + ReLU + bn Input dim - K Input chan - 6 Output chan - 64 Kernel size - 3
	Fully-connected	Input dim - 64K Output dim - 2K Hidden layers - 3 Hidden dim - {128, 64, 32} bn dim - {128, 64, 32} Nonlinearity - ReLU
OBiRIM (each stage)	*Convolution*	Same as ROBNet
	GRU	Num of GRUs - 2 GRU1 input dim - 64K GRU1 hidden dim - 1024 GRU2 input dim - 1024 GRU2 hidden dim - 1024
	Fully-connected	Input dim - 1024 Output dim - 2K Hidden layers - 4 Hidden dim - {512, 128, 64, 32} bn dim - {512, 128, 64, 32} Nonlinearity - ReLU

GD-RegNet structure and training: The GD-Net output at each Stage t is fed as 6 input channels to the GD-RegNet, similar to the ROBNet. Different from the series of GD-RegNets $\{h_\phi^{(t)}\}_{t=1}^T$ of the ROBNet, the GD-RegNet h_ϕ of the OBiRIM is a single recurrent network using gated recurrent units (GRU) to store the estimation memory. We choose the GRU as the recurrent block due to its ability to capture long and short term memory by resetting and updating the hidden state using the input sequence [153, 155]. The overall propagation of the input through the GD-RegNet, for each temporal Stage t, is given as:

1. First, a 1-D convolution extracts the input features into a set of output channels.
2. The output of the convolution stage is flattened and passed to the recurrent step of the GD-RegNet. This consists of two sequential GRU blocks, with the output hidden state of the first GRU cell passed

as the input to the second GRU cell. The hidden states of both these recurrent cells are initialized to zero.

3. Post propagation through the two GRU cells, the output hidden state of the second GRU cell is flattened and passed to a FCN with four hidden layers, with the output of the FCN, having the same dimension as the OBMNet output $\hat{\mathbf{x}}^{(t+1)}$.

4. A residual link, from the OBMNet output $\hat{\mathbf{x}}^{(t+1)}$, is added to the output of the FCN, similar to the ROBNet.

5. The normalization of the final estimate $\mathbf{x}^{(T)}$ is carried out as in (4.13).

The specific details of the GD-RegNet parameters of the OBiRIM, for a general number of users K, are given in Table 4.2. The training data as well as the training parameters are the same as the that of the ROBNet (see Sec. 4.4.1). Further, the same improved loss function (4.19) is used to also train the OBiRIM network parameters.

Remark. *Both the ROBNet and OBiRIM are trained based on the loss function* (4.19)*, incorporating the final one-bit estimate* $\mathbf{x}^{(T)}$*. Differently, the original RIM framework, introduced in [151], incorporates all the intermediate iterands* $\mathbf{x}^{(t)}$*, with* $t < T$*, in the evaluation of the MSE loss. Although the analysis stemming from the explicit incorporation of these intermediate iterands in the final loss function falls outside the scope of this work, we have utilized this strategy for a different context of one-bit detection. This analysis for one-bit MIMO is left for our future work.*

4.5 DNN-aided GD for MmWave One-bit Receivers

This section presents the GD-based detection tailored specifically to the mmWave channel. Beginning with the challenges for joint detection, we describe our proposed approach and implementation.

4.5.1 System model - One-bit receiver for mmWave channel

We begin by describing the K-user sectored LOS mmWave channel model [12]. The complex-valued channel model is expressed as

$$\bar{\mathbf{H}} = \Big[\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \ldots, \mathbf{a}(\theta_K)\Big] \ . \ \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_K), \tag{4.14}$$

where each $\mathbf{a}(\theta_i)$ is the uniform linear array (ULA) manifold with N antennas for the i^{th} user. Each user path gain α_i and path angle θ_i is independently drawn from the distributions $\mathcal{N}(0,1)$ and $\mathcal{U}(-\pi/3, \pi/3)$, respectively. We order the four users 1 to 4 in decreasing order of received channel powers[3], i.e., $\{|\alpha_i|^2\}_{i=1}^{K}$.

The received uplink unquantized signal and the corresponding one-bit quantized output of the BS is the same as eq. (4.1). For the simplification of subsequent algorithm notations, the transformations in eq. (3.3) are defined using the operators $\mathcal{T}_{\text{m}}(\cdot) : \mathbb{C}^{M\times N} \to \mathbb{R}^{2M\times 2N}$ and $\mathcal{T}_{\text{v}}(\cdot) : \mathbb{C}^{M} \to \mathbb{R}^{2M}$, for matrices and arrays, respectively.

$$\mathbf{H} = \mathcal{T}_{\text{m}}(\bar{\mathbf{H}}) = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{H}}) & -\mathfrak{Im}(\bar{\mathbf{H}}) \\ \mathfrak{Im}(\bar{\mathbf{H}}) & \mathfrak{Re}(\bar{\mathbf{H}}) \end{bmatrix}, \mathbf{r} = \mathcal{T}_{\text{v}}(\bar{\mathbf{r}}) = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{r}}) \\ \mathfrak{Im}(\bar{\mathbf{r}}) \end{bmatrix},$$
$$\mathbf{x} = \mathcal{T}_{\text{v}}(\bar{\mathbf{x}}) = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{x}}) \\ \mathfrak{Im}(\bar{\mathbf{x}}) \end{bmatrix}, \quad \mathbf{n} = \mathcal{T}_{\text{v}}(\bar{\mathbf{n}}) = \begin{bmatrix} \mathfrak{Re}(\bar{\mathbf{n}}) \\ \mathfrak{Im}(\bar{\mathbf{n}}) \end{bmatrix}.$$

An overview of the OBMNet [1] framework is provided in Sec. 4.2.2. The following section presents the challenges of directly applying this joint detection framework to the mmWave channel.

4.5.2 Challenges to joint detection using GD for the mmWave channel

This sub-section begins by analyzing the structure of the mmWave gradient in terms of the channel characteristics. Following this, we elucidate the difference in these channel characteristics between the Rayleigh and mmWave channel, with their particular bearing on the detector design. This motivates the need for a detector framework with equitable user performance.

Analyzing the mmWave one-bit likelihood gradient

The general gradient expression at each time step t, $\nabla_{\mathbf{x}}^{(t)}$, is evaluated in (4.5), following the same analysis as in [1]. The positive constant $c\sqrt{2\rho}$ is assumed to be absorbed into the matrix $\mathbf{G}$ for the remainder of this analysis. We represent the complex gradient by $\bar{\nabla}_{\mathbf{x}}^{(t)}$, written out in terms of the OBMNet gradient expression $\nabla_{\mathbf{x}}^{(t)}$ as

$$\begin{aligned} \bar{\nabla}_{\mathbf{x}}^{(t)} &= \mathcal{T}_{\text{v}}^{-1}(\nabla_{\mathbf{x}}^{(t)}) \\ &= \left[\nabla_{\mathbf{x}}^{(t)}\right]_{1:K} + j \left[\nabla_{\mathbf{x}}^{(t)}\right]_{K+1:2K}, \end{aligned} \tag{4.15}$$

[3]User ordering done during CSI acquisition and initial access

where the notation $[\cdot]_{1:K}$ is explained in Sec. 5.1. In addition, we factorize the general CSI matrix $\mathbf{H}$ as

$$\bar{\mathbf{H}} = \bar{\mathbf{A}}\,\mathbf{D}_\alpha, \tag{4.16}$$

where each column of $\bar{\mathbf{A}}$, i.e., $\mathbf{a}_i$, has unit norm and $\mathbf{D}_\alpha = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_k)$. Each α_i is a complex scalar with $|\alpha_i| = ||\bar{\mathbf{h}}_i|| \,\forall i$. Each scalar α_i is thus the channel path gain coefficient (up to a phase factor). This is not a unique factorization, but we can always factorize every complex matrix this way. One such factorization for the mmWave channel is given by (4.14). Using the factorization (4.16) for the gradient expression, the role of the varying path gains becomes evident in the detection process. We formulate this explicit factorization of the gradient as the following lemma.

Lemma 3. *The complex gradient given in* (4.15) *is directly dependent on the path gains for any general channel, and can be factorized as*

$$\bar{\nabla}_{\mathbf{x}}^{(t)} = \mathbf{D}_\alpha^{\mathrm{H}} \cdot \big(\tilde{\sigma}_{\mathbf{x},\mathrm{R}}^{(t)} + j\,\tilde{\sigma}_{\mathbf{x},\mathrm{I}}^{(t)}\big), \tag{4.17}$$

where $\mathbf{D}_\alpha$ *is as per* (4.16)*, and* $\tilde{\sigma}_{\mathbf{x},\mathrm{R}}^{(t)}$ *and* $\tilde{\sigma}_{\mathbf{x},\mathrm{I}}^{(t)}$ *are vectors in* $\mathbb{R}^K$ *consisting of homogeneous intermixing terms.*

Proof. The proof is provided in the Appendix 4.A. □

Lemma 3 illustrates the role of channel quality in the multi-user GD-based detection process. Since the i^{th} component of the complex gradient $\bar{\nabla}_{\mathbf{x}}^{(t)}[i]$ consists of the t^{th} step for the i^{th} user, each user converges to the local optima point at a different rate since it is scaled by the path loss for that particular channel, i.e., α_i. The spread in the channel powers ($\propto ||\alpha_i||^2$) will play a role in the multi-user detection performance.

The intermixing terms $\tilde{\sigma}_{\mathbf{x},\mathrm{R}}^{(t)}$ and $\tilde{\sigma}_{\mathbf{x},\mathrm{I}}^{(t)}$ in (4.17) also implicitly carry path-gain information. However, the gradient behavior is predominantly dictated by the $\mathbf{D}_\alpha$. This is illustrated in Sec. **??**. We now describe the gradient behavior, and result on joint detection, by analyzing the statistics of the mmWave and Rayleigh-fading channel.

MmWave Vs Rayleigh-fading - Ordered channel statistics

Different from the rich scattering of the Rayleigh-fading channel, there is lower diversity in the mmWave channel due to antenna correlation [12, 13]. This is pictorially represented in Fig. 4.4 by the histograms of the square root of channel power (see eq. (4.14)), i.e., $||\bar{\mathbf{h}}_i||$, in decreasing order from User 1 ($||\bar{\mathbf{h}}_1||$) to User 4 ($||\bar{\mathbf{h}}_4||$). These plots portray a larger spread in mmWave channel powers per user, vs the more equitable power distribution of the Rayleigh channel. The impact of this channel power spread among the users affects detection performance, i.e., bit error rate (BER), of each user. This is quantitatively

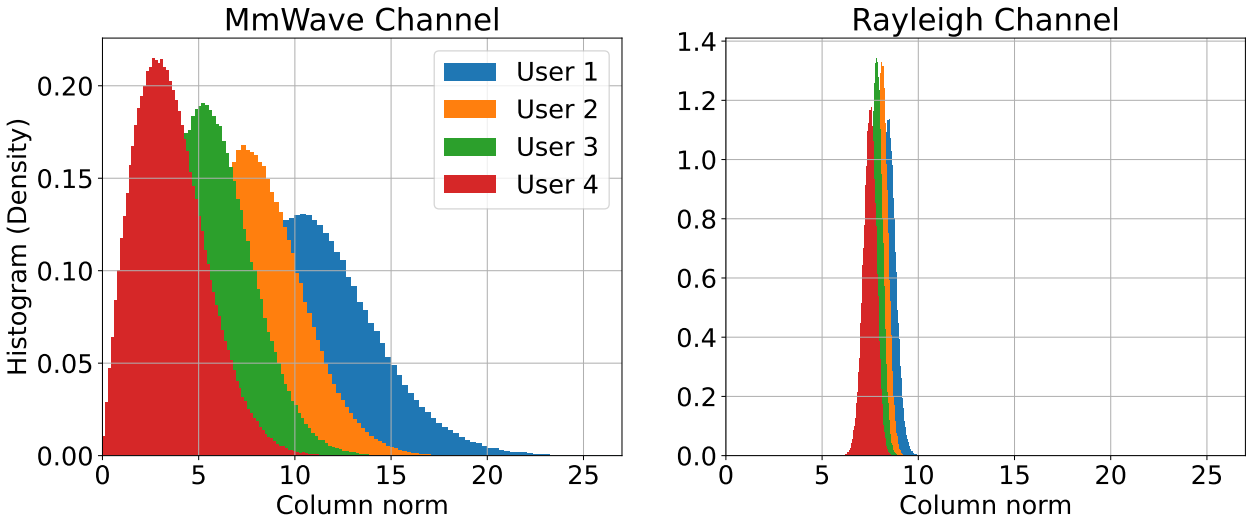


Figure 4.4: Distribution of the square root power for mmWave (left) and Rayleigh-fading channel (right) with $N = 64$ antennas, $K = 4$ users. Here User 1 has the strongest channel and User 4 the weakest.

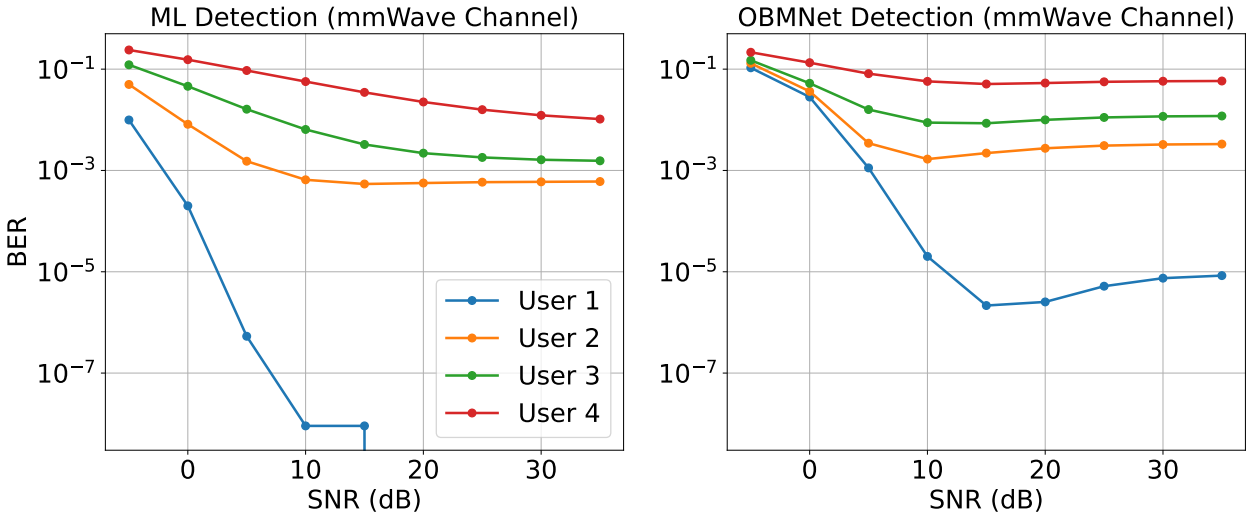


Figure 4.5: Comparing performance of ML (left) & OBMNet [1] (right) detection for mmWave channel (4.14) with $K = 4$ users, $N = 64$ antennas, each user transmitting QPSK symbols. User 1 has the strongest channel and User 4 the weakest.

illustrated in Fig. 4.5, comparing maximum likelihood (ML) detection, using exhaustive search in $\mathcal{M}^{2K}$, to the OBMNet [1], for QPSK symbols. The left plot in Fig. 4.5 shows the overt disparity in the ML detection performance for the different users; User 1, with the strongest channel, experiences more than six orders of magnitude lower BER, at saturation, compared to User 4, with the weakest channel. The right plot in Fig. 4.5 highlights that the OBMNet [1], optimizing for the joint detection performance, is limited by the weakest users, thus unable to equitably handle users with stronger channels.

Towards equitable detection for mmWave channels

Based on the BER performance plots for the mmWave channel in Fig. 4.5, it is evident that *(i)* The reduced diversity of the mmWave channel results in user performance being capped by the channel quality, and further *(ii)* Joint multi-user GD-based detection will be biased by the weakest users, hence unable to capitalize on the stronger user channels.

A possible solution to improve joint detection would be to regulate mmWave channel access to users with similar channel powers, thereby clustering different users based on channel quality. However, this being a highly dynamic approach, especially in the number of user clusters, forebodes a difficult access protocol problem. The neural detection framework developed in this work is motivated by improving the joint multi-user detection to equitably match the performance of each user to the channel quality. In particular, we aim to address the following two main challenges for the general mmWave channel: *(i)* Overcoming the path-loss based scaling of the likelihood gradient to stabilize the trajectory of each user, and *(ii)* Ensure detection performance is not limited by the weakest users in the channel. To this end, the following section addresses GD-based joint multi-user detection for mmWave channels, to overcome these challenges.

Signal detection for the receiver (4.1) is approached via regularized DNN-aided GD, with a novel hierarchical detection training strategy. Our approach consists of three major components, elaborated next.

4.5.3 User-matched regularized GD

The regularized GD detection for one-bit receivers [36] augments the GD step (4.5) via a DNN-aided projection step, thereby fine-tuning each iteration. For mmWave channels, we devise a modification - a user-matched regularized GD. Herein, we incorporate the mmWave channel path gains per-user $\{\alpha_i\}_{i=1}^{K}$,

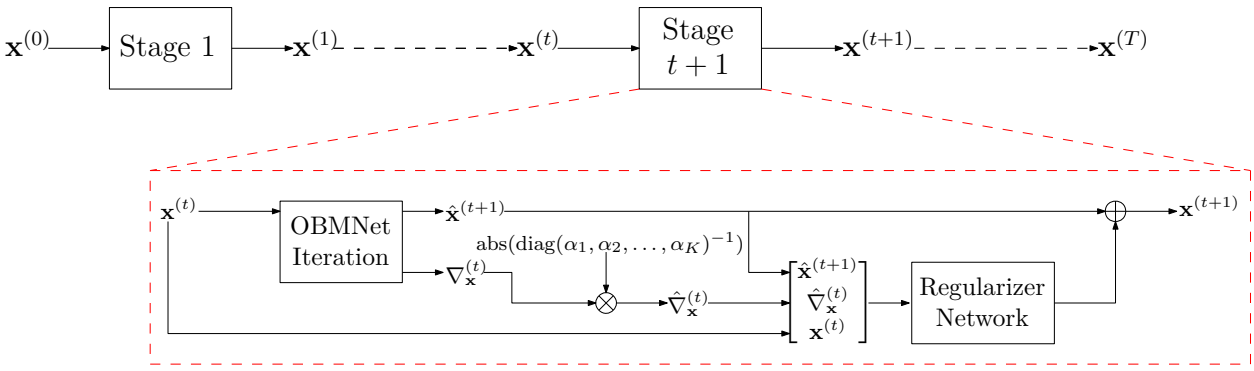


Figure 4.6: Block diagram for the mmW-ROBNet

from (4.14), as

$$\hat{\nabla}_{\mathbf{x}}^{(t)} = \mathcal{T}_{\text{m}}\big(\text{abs}(\text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_K)^{-1})\big)\, \nabla_{\mathbf{x}}^{(t)} \tag{4.18a}$$

$$\hat{\mathbf{x}}^{(t+1)} = \mathbf{x}^{(t)} - \alpha^{(t)} \hat{\nabla}_{\mathbf{x}}^{(t)} \tag{4.18b}$$

$$\mathbf{x}^{(t+1)} = \hat{\mathbf{x}}^{(t+1)} + h_{\phi}^{(t)}(\mathbf{x}^{(t)}, \hat{\nabla}_{\mathbf{x}}^{(t)}, \hat{\mathbf{x}}^{(t+1)}). \tag{4.18c}$$

The user-matching step (4.18a) homogenizes the gradient from (4.5), stabilizing the unequal channel power scaling among the users, as seen in Lemma 3. Update steps (4.18b)-(4.18c) execute regularized GD with the user-matched gradient. The function $h_{\phi}^{(t)}$ is the DNN-based parametric regularization at iteration t, elaborated further in Sec. 4.5.6. The regularized GD algorithm (4.18), implemented as an unfolded DNN, i.e., the mmWave regularized one-bit net (mmW-ROBNet), is illustrated in Fig. 4.6.

4.5.4 Constellation-aware DNN loss function

In order to tailor the DNN loss function to the M-QAM symbol recovery, we incorporate the regularized loss function from Sec. 4.3.2.

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} \big[||\mathbf{x}_n^{(T)} - \tilde{\mathbf{x}}_{\text{train},n}||^2 + \lambda\, \mathcal{R}(\mathbf{x}_n^{(T)}, \tilde{\mathbf{x}}_{\text{train},n})\big], \tag{4.19}$$

where $\mathcal{R}(\cdot)$ is a constellation-aware regularization, based on a smooth quantization of the network output, $\mathcal{Q}_{\beta}(\cdot)$, implemented as

$$\mathcal{R}(\mathbf{x}_n^{(T)}, \tilde{\mathbf{x}}_{\text{train},n}) = ||\mathcal{Q}_{\beta}(\mathbf{x}_n^{(T)}) - \tilde{\mathbf{x}}_{\text{train},n}||^2. \tag{4.20}$$

For QPSK symbol recovery, this is given by $\mathcal{Q}_\beta(x) = \tanh(\beta\, x)$. This saturating nonlinearity $\mathcal{Q}_\beta(\cdot)$ attenuates symbol errors within the true constellation boundaries and amplifies errors for crossing over symbol boundaries. The constellation-aware regularization (4.20) thus incorporates symbol error rate (SER), in addition to MSE, into the total loss (4.19), resulting in a robust training.

4.5.5 Hierarchical detection training

The final block for robust DNN-aided GD is the enhanced DNN training procedure. In particular, we control the trajectory of intermediate GD iterates $\{\mathbf{x}^{(t)}\}_{t=1}^{T-1}$ in (4.18), essential to the final estimate $\mathbf{x}^{(T)}$. To this end, we propose a sub-loss $\mathcal{L}^{(t)}$ at each iteration t as

$$\begin{aligned}\mathcal{L}^{(t)} = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} \big[||\mathbf{w}^{(t)} \odot \mathbf{x}_n^{(t)} - \mathbf{w}^{(t)} \odot \tilde{\mathbf{x}}_{\text{train},n}||^2 \\ + \lambda\, \mathcal{R}(\mathbf{w}^{(t)} \odot \mathbf{x}_n^{(t)}, \mathbf{w}^{(t)} \odot \tilde{\mathbf{x}}_{\text{train},n})\big]\end{aligned} \tag{4.21}$$

where,

$$\mathbf{w}^{(t)} = \tilde{c}^{(t)} \left[1, \exp(-\kappa_t), \ldots, \exp(-(K-1)\,\kappa_t)\right]^{\mathrm{T}}.$$

Here $\mathbf{w}^{(t)}$ is the masking vector, κ_t is the user masking coefficient and $\tilde{c}^{(t)}$ is the normalization constant such that $||\mathbf{w}^{(t)}|| = 1$, at the t^{th} iteration. The operator $\odot$ denotes element-wise product. The regularization $\mathcal{R}(\cdot)$ is the same as (4.20). The total DNN training loss function is given by

$$\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}^{(t)}. \tag{4.22}$$

Owing to the exponential decay over the user index, the masking vector $\mathbf{w}^{(t)}$ attenuates the users that experience a weaker channel, reducing their contribution to the loss function (4.21). Further, we begin with a large value for the masking coefficient κ_t and decrease this over the GD iterations to $\kappa_T = 0$. This results in the first few regularization sub-networks $h_\phi^{(t)}$ being trained to detect the stronger users only. The subsequent sub-networks gradually add users in decreasing order of their channel quality, by jointly detecting these with the stronger users. Finally, the last sub-network $h_\phi^{(T)}$ is trained to jointly detecting all the users. Since the training strategy (4.21)-(4.22) detects multiple users from the strongest to the weakest, it is called hierarchical detection or *HieDet training*.

Table 4.2: DNN Parameters of the Regularizer Network (For K Users)

Network	Layer	Parameters
mmW-ROBNet (each stage)	*Convolution (1-D)*	conv + ReLU + bn
		Input dim - K
		Input chan - 6
		Output chan - 64
		Kernel size - 3
	Fully-connected	Input dim - 64K
		Output dim - 2K
		Hidden layers - 3
		Hidden dim - {128,64,32}
		bn dim - {128,64,32}
		Nonlinearity - ReLU

4.5.6 DNN implementation for user-matched regularized GD

Inspired by the potential of model-based algorithm unrolling for general signal processing [1, 50, 54, 133], the unfolded DNN, ROBNet, is developed as an approach to solve the constrained optimization (4.4) for any general channel matrix $\mathbf{H}$, via a DNN-augmented GD algorithm [36]. The mmW-ROBNet framework (4.18), introduced in Sec. 4.5.3, incorporates the specific properties of the mmWave channel. The implementation details are elaborated as follows.

(i) The T-stage regularized GD algorithm (4.18), is unfolded into T distinct sub-networks (each represented as Stage t in Fig. 4.6).

(ii) At the beginning of each Stage $t+1$, the OBMNet iteration (4.5) generates the t^{th} gradient and output $\nabla_{\mathbf{x}}^{(t)}$ and $\hat{\mathbf{x}}^{(t+1)}$, respectively.

(iii) The mmWave-channel powers per user, $\{|\alpha_i|^2\}_{i=1}^K$, user-matches the OBMNet-generated gradient $\nabla_{\mathbf{x}}^{(t)}$ as (4.18a), to get $\hat{\nabla}_{\mathbf{x}}^{(t)}$

(iv) The previous estimate $\mathbf{x}^{(t)}$, user-matched gradient $\hat{\nabla}_{\mathbf{x}}^{(t)}$ and OBMNet output $\hat{\mathbf{x}}^{(t+1)}$ is passed to the Regularization Network $h_\phi^{(t)}$ for fine-tuning (see Table 4.2 for DNN parameters).

(v) Additionally, a residual link from the OBMNet output is fed to the output of the Regularization Network, thereby imparting a stage-dependent correction, to the unconstrained OBMNet step.

(vi) The final output $\mathbf{x}^{(T)}$ is normalized as $\mathbf{x}^{(T)} \leftarrow \frac{\sqrt{2K}}{||\mathbf{x}^{(T)}||}\mathbf{x}^{(T)}$.

4.6 Experimental Results: ROBNet and OBiRIM

We now evaluate our regularized networks ROBNet and OBiRIM. First we describe the simulation setup, followed by the results of the various tests along with comments.

Simulation setup We evaluate the detector on two different M-QAM constellations with different channels, user, BS antennas and input SNR $= \frac{\mathbb{E}(||\mathbf{Hx}||^2)}{\mathbb{E}(||\mathbf{n}||^2)}$ parameters:

(i) The QPSK constellation with $K = 4$ users, $N = 32$ BS antennas and SNR in the range -5 to 35 dB.

(ii) The 16-QAM constellation with $K = 8$ users, $N = 128$ BS antennas with SNR in the range 10 to 45 dB.

Both the simulations setups (i) and (ii) follow the standard simulations conducted in [1, 35, 132]. For both the constellation cases, a Rayleigh fading channel $\mathbf{H}$ is considered with each entry chosen from the $\mathcal{CN}(0, 1)$ distribution. Unless otherwise stated, we assume perfect channel state information (CSI) available at the BS.

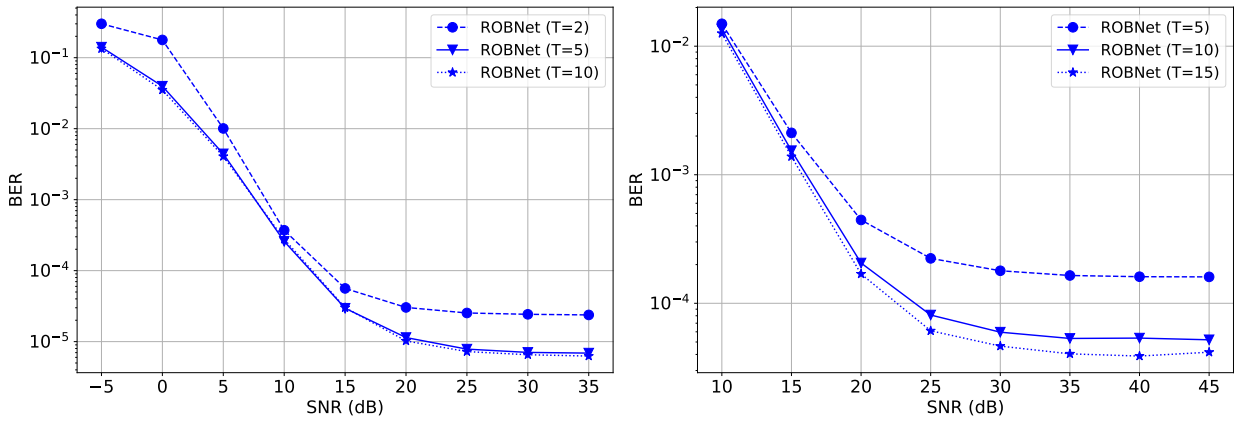
Performance benchmarks We benchmark our algorithms against the existing model-based and DNN-aided one-bit detectors for state-of-the-art detection. For the simulation setup *(i)*, as described in the paragraph above, we lower bound the BER by the maximum-likelihood detector (ML Detector). Using the exhaustive constellation search, this method grows exponentially with each added user as well as increase in modulation order. However, this presents the best recovery possible, directly solving the constrained optimization problem (4.4). ML detection for 16-QAM (simulation setup *(ii)*) presents as a much larger computational complexity for our scale of the simulation setup considered, and is hence not evaluated. The OBMNet [1] is used as the main benchmark, on which we propose improving, by means of the regularized GD (4.8). We also provide the performance of the n-ML algorithm, from [35], to benchmark against the GD-based detector using cdf-based likelihood (4.2). For testing the general channel detection performance we also benchmark our algorithm against the FBM-DetNet [20], implemented for the same number of iterations as the OBMNet.

Remark. *The work in [132] extensively tests end-to-end learning via different DNNs like Resnets, Densenets and Hypernetworks for one-bit detection. However, the presented results in this work show the robust model-based OBMNet to exceed the performance of these networks. Thus, we have omitted the inclusion of these end-to-end learning approaches for benchmarking our regularized one-bit detection approach.*

Network and model parameters Consistent with the benchmarks established in [1], the OBMNet is run for ten and fifteen iterations ($T \in \{10, 15\}$) for simulation setups *(i)* and *(ii)*, respectively. The n-ML method is executed for a maximum of $T = 500$ iterations, with a step size of 0.001, to ensure convergence. The network parameters and training details for our proposed networks - ROBNet and OBiRIM, have been provided in Sec. 4.4.1 and 4.4.2. In contrast to the higher number of iterations for the given benchmarks

above, we execute both the ROBNet and the OBiRIM for only five and ten GD iterations ($T \in \{5, 10\}$) for simulation setups *(i)* and *(ii)*, respectively. Thus the added utility of the regularized GD algorithm also presents as an advantage in reducing the number of GD iterations. To avoid overloading the networks for large SNR ranges during training, the proposed networks are trained for a single intermediate SNR (15 dB for simulation *(i)* and 25 dB for simulation *(ii)*) and tested on the entire range mentioned above. A similar strategy for training unfolded and recurrent neural networks was used in [54].

4.6.1 Intrinsic testing of DNN-aided regularized GD



(a) (b)

Figure 4.7: Testing the ROBNet for different number of stages T (a) QPSK transmitted symbols, with $N = 32$, $K = 4$ (b) 16-QAM transmitted symbols, with $N = 128$, $K = 8$

We begin by testing the performance of the networks implementing the regularized GD update (4.8). In particular, we test the performance by varying the network parameters, i.e., the number of network stages T. For the considered test, we evaluate the performance of the unfolded network, the ROBNet.

We train and test the ROBNet for different stages T, corresponding to number of regularized GD iterations. The plots given in Fig. 4.7(a) portray the BER performance for the QPSK transmitted symbols. Based on the performance plots, There is a marked gain in performance from $T = 2$ to 5 stages, with saturation in performance beyond this. By increasing the number of layers beyond a certain limit, we increase network complexity and runtime with extremely marginal gains in performance. This is also supported by the results for the 16-QAM constellation, as shown by the plots in Fig. 4.7(b). Here too, a significant performance boost is observed as we increase from $T = 5$ to 10, with subsequent increase in the number of stages only marginally increasing performance. Based on the observed recovery results, we utlize $T = 5$ and $T = 10$ layers for QPSK and 16-QAM symbols, respectively.

4.6.2 Recovered constellation

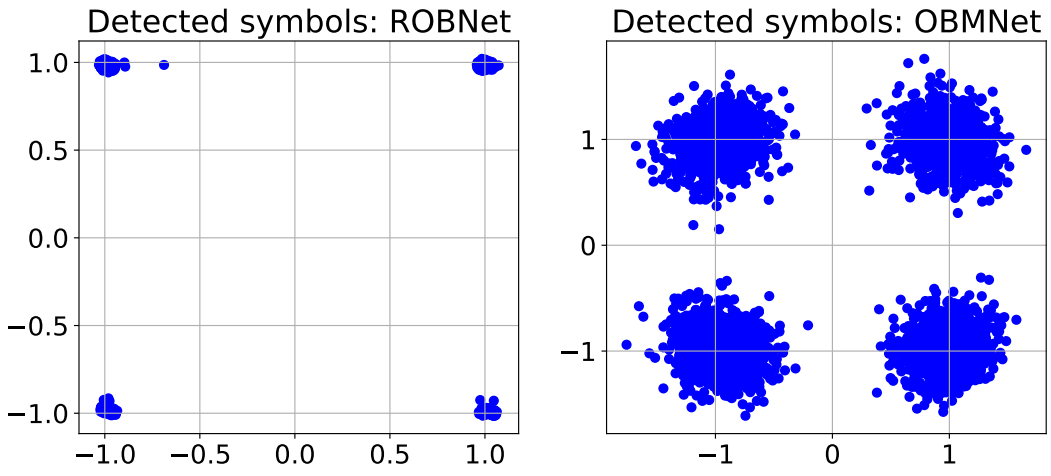


Figure 4.8: Recovered QPSK constellation for ROBNet compared to OBMNet [1], with $N = 32$, $K = 4$ (red dots represent incorrectly detected symbols)

We qualitatively analyze the recovered M-QAM constellation symbols, comparing these to the recovered symbols from the OBMNet, which doesn't utilize any additional regularization. The symbol recovery is demonstrated for the training SNR of the networks, i.e., 15 dB for QPSK and 25 dB for 16-QAM. The recovered QPSK symbols are given in Fig. 4.8. As can be seen from these plots, the OBMNet results in recovered symbols with a larger cluster spread. The combination of the increased network expressivity for the ROBNet, along with the constellation-aware network loss function (4.19), results in much sharper recovered symbol clusters.

The symbol recovery for 16-QAM constellation presents the more stark contrast on the effect of the regularized GD method (4.8). Although the OBMNet is able to effectively recover the 16-QAM symbols from the one bit data, the cluster shapes are non-homogeneuous in the symbol power. As can be seen from the density of incorrectly detected symbols (red scatter points), this non-homogeneity results in more incorrectly detected symbols. The regularization introduced by the ROBNet, in contrast, presents a more homogeneous recovered constellation, irrespective of the 16-QAM symbol powers. As visually evident, this is responsible for fewer incorrectly detected symbols.

Following the qualitative visual analysis of the recovered symbols, we now move on to the quantitative analysis.

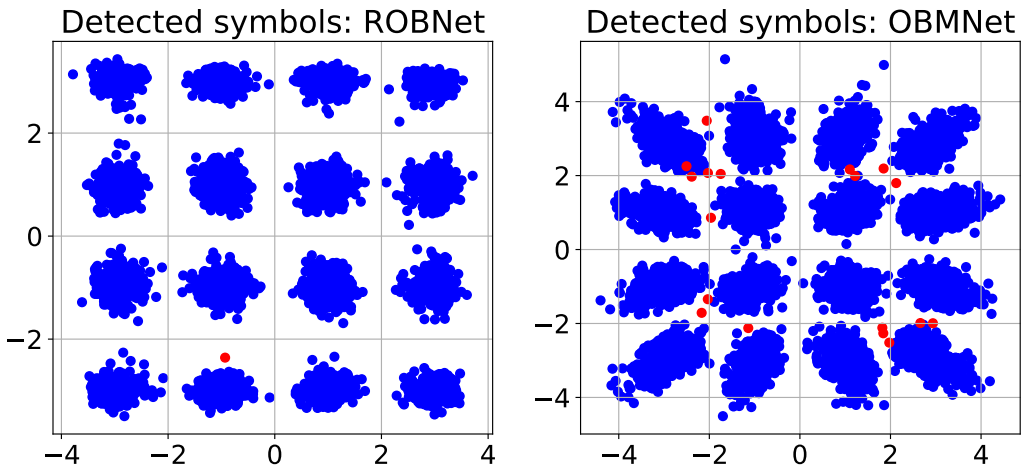


Figure 4.9: Recovered 16-QAM constellation for ROBNet compared to OBMNet [1], with $N = 128$, $K = 8$ (red dots represent incorrectly detected symbols)

4.6.3 Detection for single Rayleigh-fading channel

Through this test we demonstrate the strength of our proposed approach, when implemented for channel-specific detection. As stated in Sec. 5.1, most conventional end-to-end DNN-based detectors, both for unquantized as well as one-bit received data [57–59, 133, 134, 136], are trained and tested for a single channel. Such detectors are applicable to highly static and directional channels, with minimum CSI variation. Real-world channels, like Rayleigh-fading channels, are more dynamic; robust detector design should thus be channel state-invariant, trained on the entire set of random Rayleigh-fading channels and avoiding the need to be retrained for each new CSI matrix. Prior to testing the proposed networks in this work on the entire distribution of Rayleigh-fading channels, we perform the channel-specific detection to ascertain the performance for this widely utilized model by different works. In the context of DNN design, this test is akin to the overfitting test. The different networks and approaches are trained and tested on a single channel $\mathbf{H}$, sampled from the distribution of Rayleigh-fading channels. Further, we normalize the columns of the channel matrix $\mathbf{H}$ and scale it by the number of antennas N, ensuring each user receives the same channel power.

The channel-specific BER performance for QPSK symbols is shown in Fig. 4.10. As seen from this plot, all the networks and the algorithms approach very low BER values when trained and tested with a very well conditioned channel with equal power distribution among the corresponding users [4]. However, such ideal performance requires overfitting the networks for a given channel model, which presents extensive

[4]Equal per-user channel power is especially important for joint GD-based detection. We analytically validate this, in detail, through our future work.

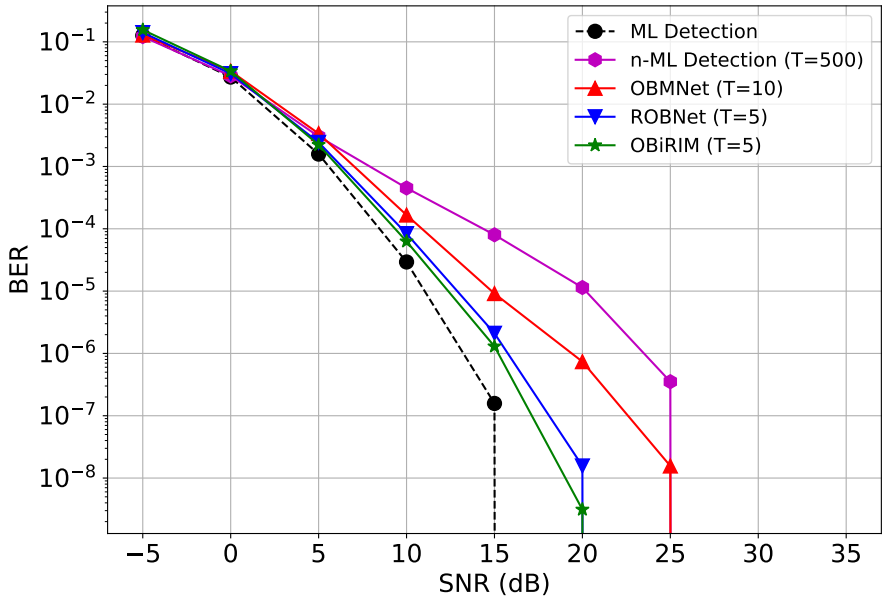


Figure 4.10: Performance comparison of improved networks for channel-specific detection for QPSK constellation with number of antennas $N = 32$ and the number of users $K = 4$.

practical challenges. As seen from the plots, the OBMNet, with its improved sigmoid likelihood formulation exceeds the n-ML approach, further highlighting the utility of this likelihood formulation. As can be seen from the results, both our proposed networks exceed the OBMNet performance for the channel-specific detection, approaching the ideal ML-detection. This further enforces the utility of the proposed regularized GD algorithm and the constellation-aware loss function.

The channel-specific BER performance for the 16-QAM symbols is shown in Fig. 4.11. Here too, all the networks and algorithms are trained and tested on a single channel matrix, with equal per-user channel power. Based on the results in this plot, the contrast in performance between the regularized one-bit GD and the competing algorithms is more starkly visible, highlighting the strength of this strategy for higher order M-QAM constellations. The presence of a non-zero BER floor for all algorithms (as compared to Fig. 4.10), stems from the more challenging case of recovering higher order constellation symbols from one-bit data.

4.6.4 Detection for general channel

We now present the results for the networks trained and tested on the set of all Rayleigh-fading channel matrices $\mathbf{H}$, by randomly sampling from this distribution for each training minibatch. Once the

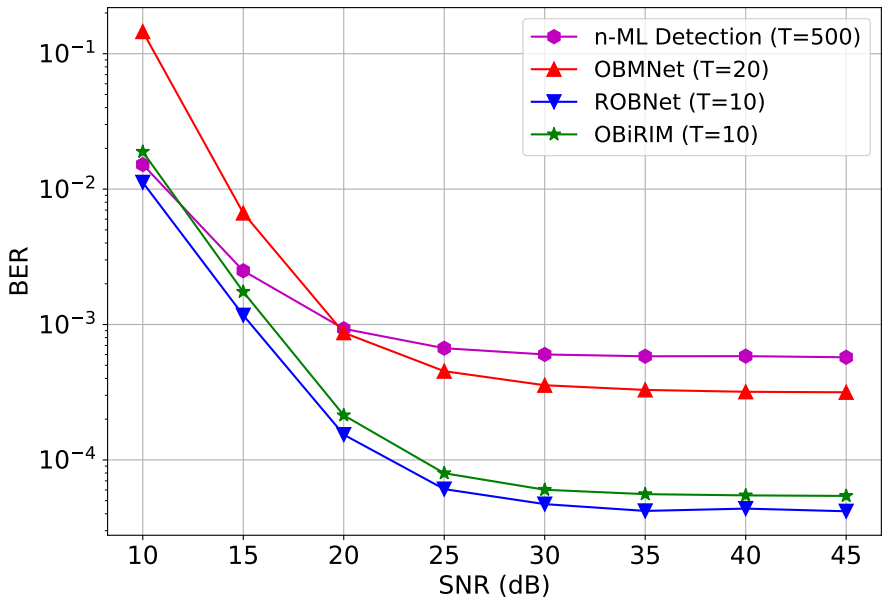


Figure 4.11: Performance comparison of improved networks for channel-specific detection for 16-QAM constellation with number of antennas $N = 128$ and the number of users $K = 8$.

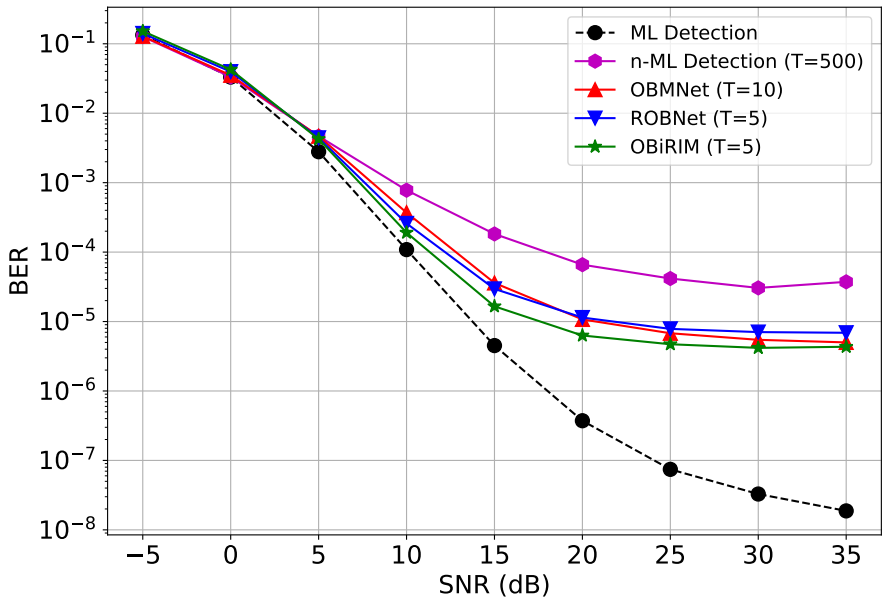


Figure 4.12: Performance comparison of improved networks for general channel detection for QPSK constellation with number of antennas $N = 32$ and the number of users $K = 4$.

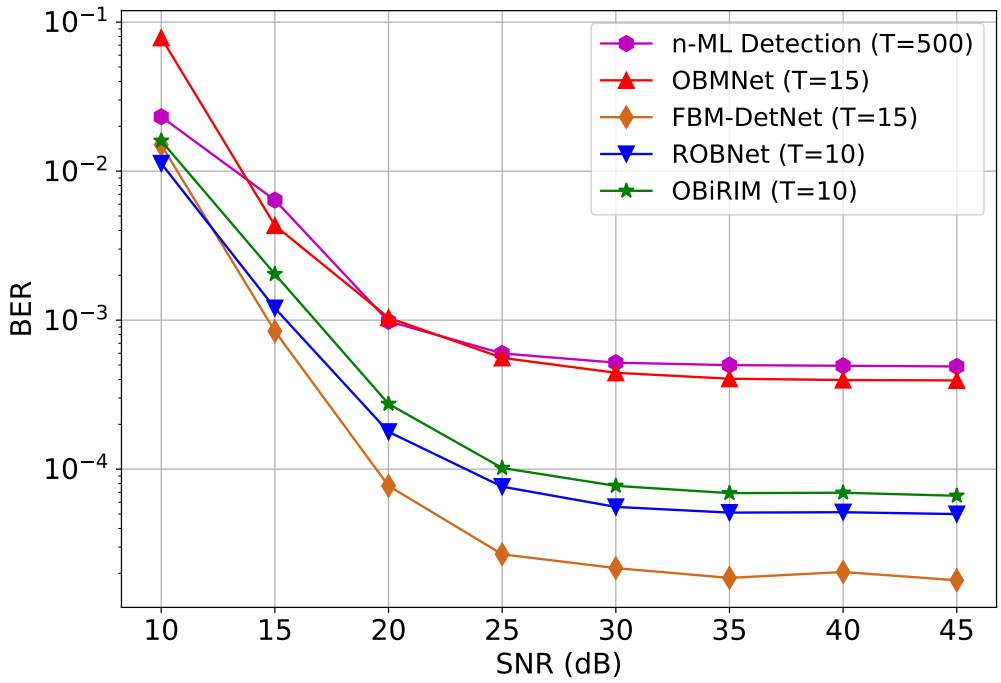


Figure 4.13: Performance comparison of improved networks for general channel detection for 16-QAM constellation with number of antennas $N = 128$ and the number of users $K = 8$.

networks have been trained in this manner, they do not need to be fine-tuned or re-trained for each new channel matrix, thus acting as general Rayleigh channel detectors.

The general channel BER performance for the QPSK symbols is given in Fig. 4.12. As seen from the performance plots, our proposed DNN-based detectors, ROBNet and OBiRIM, are able to generate the same performance as the OBMNet and the FBM-DetNet for the QPSK symbols with much fewer GD iterations. In addition, as seen from the recovered constellations plots in Sec. 4.6.2, these networks generate sharper constellation clusters with much smaller cluster spread. Although this does not directly translate to improved BER performance for lower order constellations like QPSK, it does bode advantageous for higher order constellations.

The BER performance for the 16-QAM constellation symbols is shown in Fig. 4.13. As can be seen from these plots, the improved regularization framework directly translates to an improved relative BER performance, as compared to the OBMNet, for the higher order 16-QAM constellation symbols. Further, the BER performance, especially the high-SNR BER floor, for general channel detection is similar to that of the channel-specific detection, seen in Fig. 4.11. This can be attributed to the channel hardening effect seen by increasing the number of receiver antennas, improving the overall channel conditioning for any general 16-QAM channel. However, we observe that the quantization-specific learnable projection of the

FBM-DetNet outperforms both the ROBNet and the OBiRIM. The sharper learnable quantization to the M-QAM symbols is responsible for lower cluster spreads. The ROBNet and OBiRIM are unable to sharpen constellation clusters beyond a certain limit and the hence under-perform in BER to the FBM-DetNet.

4.6.5 Detection for general channel - Noisy channel estimate

The different model-based and DNN-based approaches described above rely on perfect channel estimates. However, practical systems introduce an estimation error in acquiring the channel state information. Although different channel estimation algorithms have been studied for one-bit systems [26–28], we model the channel estimation via a general estimation error. The estimated Rayleigh fading channel is modeled as $\hat{\mathbf{H}} = \mathbf{H} + \Delta\mathbf{h}$. Here the introduced estimation error is modeled as an additive complex Gaussian with each term $\left[\Delta\mathbf{h}\right]_{i,j}$ drawn from the distribution $\mathcal{CN}(0, \sigma^2_{\mathbf{h}})$. We analyze the BER performance on the pre-trained networks with perfect CSI as a function of the introduced channel noise $\sigma^2_{\mathbf{h}}$. All the trainable networks in the subsequent performance comparison are trained on perfect CSI, setting a uniform reference point for all networks, but tested on noisy CSI. Through this test, we assess the inherent network resiliency for both the ROBNet as well as OBiRIM, compared to other benchmarks.

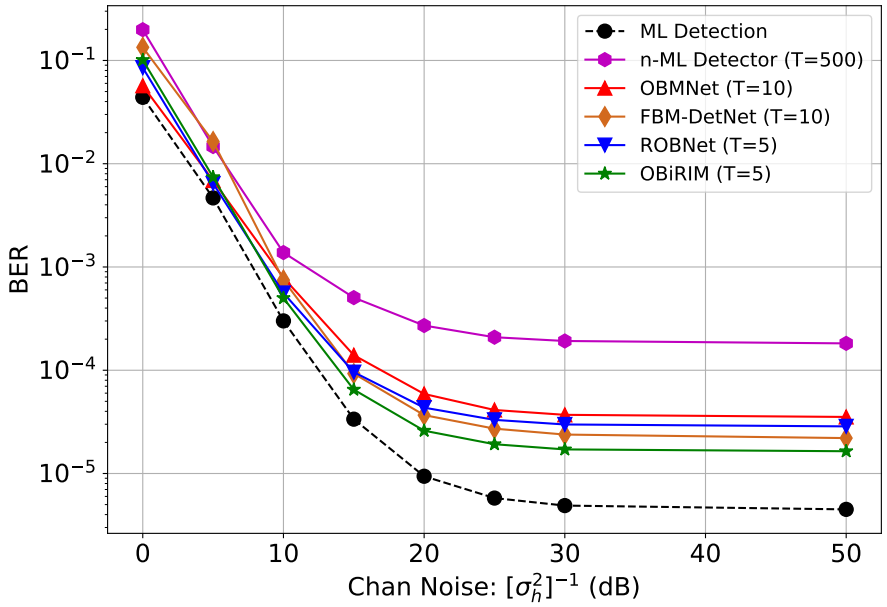


Figure 4.14: Performance comparison of improved networks for general channel detection with imperfect CSI for QPSK constellation with number of antennas $N = 32$ and the number of users $K = 4$.

The BER performance for the QPSK constellation symbols, as a function of this estimation noise

$\sigma_{\mathbf{h}}^2$ is given in Fig. 4.14. The detection performance is analyzed at the training SNR for QPSK symbols, i.e., 15 dB. As can be seen from these plots, both the regularized GD networks, i.e., the ROBNet and the OBiRIM, are more resilient to channel estimation noise, as compared to the unregularized OBMNet. Further, the FBM-DetNet also performs comparably to the ROBNet and OBiRIM. The small performance gap among all these algorithms goes on to further highlight the strength of the original OBMNet framework for lower order M-QAM constellations. Consistent with the results of Fig. 4.12, we observe marginal improvement over the OBMNet framework with additional regularization, for lower order M-QAM constellations. However, increasing constellation order brings out the increased resilience of our proposed approach over the OBMNet.

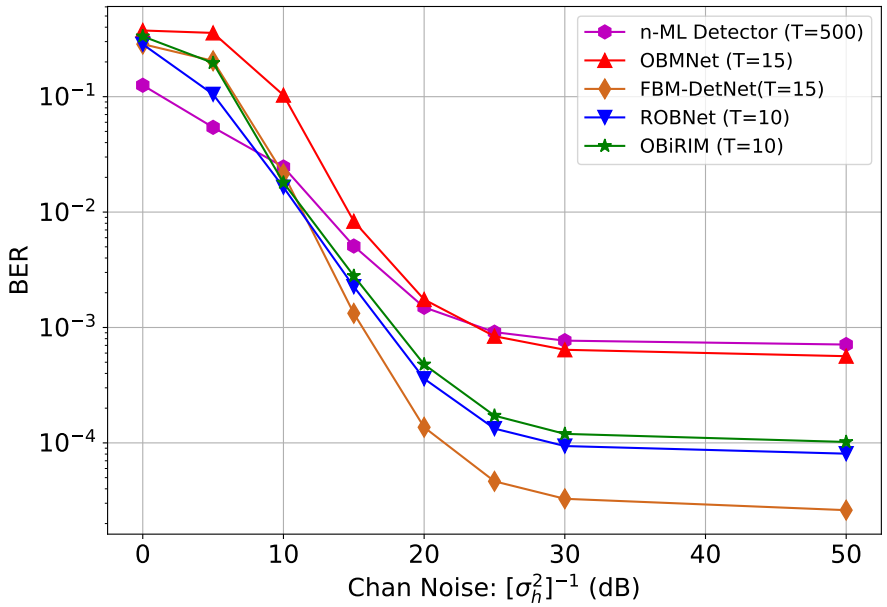


Figure 4.15: Performance comparison of improved networks for general channel detection with imperfect CSI for 16-QAM constellation with number of antennas $N = 128$ and the number of users $K = 8$.

The BER performance for 16-QAM constellation symbols as a function of the added channel estimation error is provided in Fig. 4.15. The detection performance is analyzed at the training SNR for 16-QAM symbols, i.e., 25 dB. There is a markedly increased performance gap in the performance of the regularized GD approach (both ROBNet and OBiRIM) for the 16-QAM constellation symbols. The increased network expressivity and training of our proposed approach enables accommodation of CSI estimation errors, in spite of the presence of higher order constellation symbols. However, the FBM-DetNet outperforms both the ROBNet as well as the OBiRIM in the resilience to channel estimation noise. As stated in Sec. 4.6.4, this is attributed to reduced cluster spread generated by the quantization-based projection to M-QAM symbols

of the FBM-DetNet.

We thus infer that the combination of the general parametric regularization, improved loss function and training on multiple Rayleigh-fading channel matrices, makes these extremely robust, over the unregularized OBMNet, one-bit detection networks. The observed resilience to channel estimation errors enables the use of these networks in conjunction with standard one-bit channel estimation algorithms, without affecting detection performance. Additionally, the proposed networks do not need to be separately trained for noisy channel estimates; existing ideally trained networks can be directly used with noisy channel data.

Remark. *Based on the observed results from Figs. 4.10-4.15, we can observe a difference in behavior for the unfolded ROBNet and recurrent OBiRIM. In particular, the OBiRIM is shown to perform marginally better for the lower order QPSK, whereas the ROBNet performs marginally better for the higher order 16-QAM. This highlights an important trade-off between (i) Capturing correlation through system memory and, (ii) Network expressivity through number of parameters. For the simpler QPSK system model, the system memory, through DNN recurrence, is slightly more efficient at capturing correlation among the different intermediate iterates* $\{\mathbf{x}^{(t)}\}_{t=1}^{T}$. *This is translates to marginally better performance for the simpler QPSK case. However, as we increase the constellation order, the recovery requires more network expressivity. Increasing the number of iterations the ROBNet increases the number of sub-networks and thus the number of trainable parameters. On the other hand, the OBiRIM, with more number of iterations, retains the same number of parameters due to parameter sharing. For the higher order 16-QAM symbols, we observe that the network expressivity and number of parameters wins out over the ability of the OBiRIM to capture correlation (with the same number of parameters). Thus the ROBNet now marginally outperforms the OBiRIM*[5].

4.7 Experimental Results: mmW-ROBNet

The following section presents the results for the mmW-ROBNet, specifically tailored to detection for the mmWave channel.

Simulation setup: The sectored mmWave channel [12] is considered, as described in Sec. 5.2. The signal from $K = 4$ users, transmitting QPSK symbols, is received at a BS ULA with $N = 64$ antennas.

DNN hyperparameters and training: We follow the same DNN training procedure, namely the minibatch description, optimizer and weight decay, as described in our previous work [36]. The mmW-ROBNet consists of $T = 5$ iterations. For the constellation-aware loss function (4.19)-(4.20), we assign $\lambda = 5$ and $\beta = 2$. All networks are trained at an intermediate SNR of 15 dB. Finally, the sequence of user masking

[5]Detailed analysis of this trade-off between recurrence and parameter richness falls outside the scope of this work.

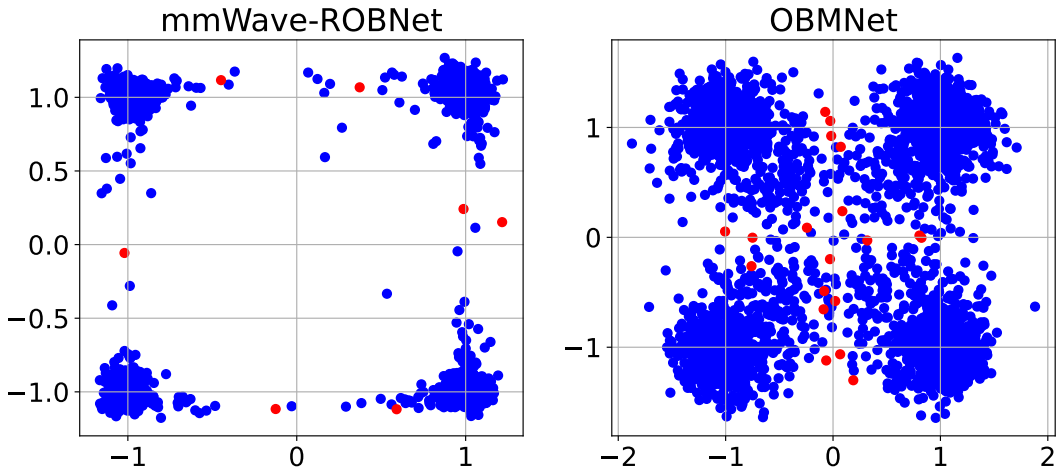


Figure 4.16: Recovered constellation for joint detection of all $K = 4$ users, received at ULA with $N = 64$ antennas at SNR $= 15\,\text{dB}$

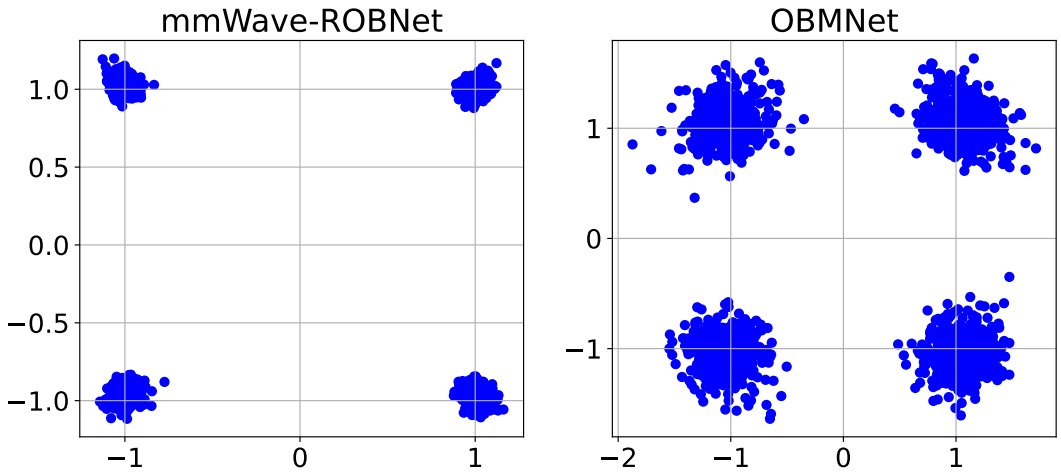


Figure 4.17: Recovered constellation for User 1 and User 2 only, received at ULA with $N = 64$ antennas at SNR $=$ $15\,\text{dB}$

coefficients $\{\kappa_t\}_{t=1}^{5}$, in (4.21), is given by $\{10, 1, 0.5, 0.1, 0\}$.

Performance benchmarks: The OBMNet [1], with $T = 20$ iterations is considered as our primary benchmark. We also compare our framework against the the n-ML [35] algorithm optimizing the cdf likelihood. The Rayliegh-ROBNet framework, from our previous work [36], is also considered. Finally, the ML detection, based on an exhaustive search, forms the lower limit of the BER performance.

4.7.1 Recovered constellation: Scatterplots

We begin with a qualitative performance comparison of the proposed mmW-ROBNet and the benchmark unregularized GD, i.e., the OBMNet [1]. The recovered constellation via joint detection of all

four users is provided in Fig. 4.16 (red dots represent incorrectly detected symbols). Based on these plots, it is evident that both the mmW-ROBNet as well as the OBMNet have poor recovery with bit errors for joint four-user detection. Since all the users experience different quality channels, the recovered constellation for the weaker users is evidently responsible for this performance limitation. Reinforcing this rationale, we also illustrate the scatterplots of the strongest two users, i.e., User 1 and User 2, in Fig. 4.17. Due to the better quality channel for these users, the mmW-ROBNet, with HieDet training, is able to generate more uniform constellation clusters with markedly reduced cluster spread compared to the OBMNet.

4.7.2 Detection performance for general mmWave channel

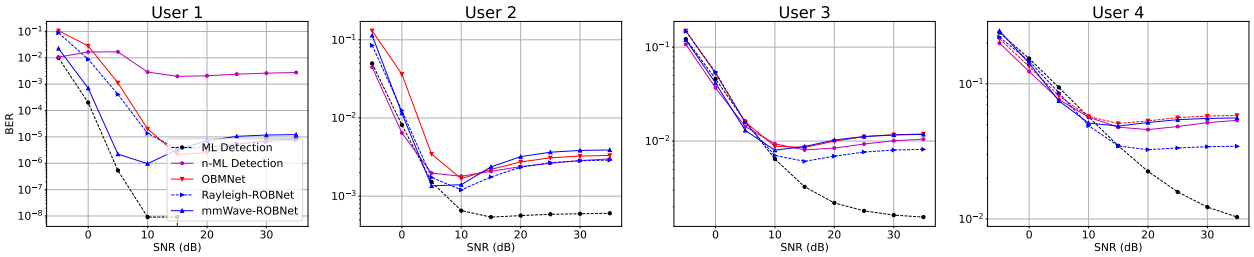


Figure 4.18: Comparison of per-user BER Vs SNR performance, from the strongest user, User 1, to the weakest, User 4. All $K = 4$ users, transmitting QPSK symbols, are jointly detected at the BS with $N = 64$ antennas

The per-user BER vs SNR performance is given in Fig. 4.18. As seen from these plots, the sigmoid-based likelihood (4.4) improves on the cdf-based n-ML detector. Although our Rayleigh-ROBNet [36] outperforms the benchmark detectors, it is also unable to captalize on the channel with the highest power, i.e., User 1. Note that the BER average across the users will not capture this difference in detector performance. The combination of the user-matched GD, along with the sequential HieDet training is able to capitalize on the users with a better channel without significantly affecting users with worse channel quality. Finally, all approaches saturate in BER for high SNR, and reducing the gap to ML is still part of our ongoing research.

4.8 Conclusions and future work

In this chapter we have proposed a regularized one-bit neural detector based on a novel regularized GD-based strategy for improving on the state-of-the-art OBMNet. The learnable DNN-based regularization is effectively able to improve on the OBMNet estimate on a per-iteration basis. To this end, we have developed two unique regularization networks: *(i)* ROBNet, using an unfolded DNN architecture, and *(ii)*

OBiRIM, using a RIM-based architecture. We also developed a novel constellation-aware loss function for DNN training, through which we are able to implicitly address bit errors. Through our model-aided DNN design as well as training for a general Rayleigh-fading channel, we are able to build a one-bit detector that doesn't need to be retrained for each new channel response. Finally, through our results we highlight the strength of the proposed approach, especially for the higher-order M-QAM constellations.

Additionally, we have presented a novel one-bit neural detection approach, mmW-ROBNet, specifically tailored to the mmWave channel model. We have illustrated the dependence of the detection performance, per user, on the channel power for that user. Addressing this limitation, we have modified the existing regularized GD framework, the ROBNet, to overcome these challenges. In particular, by means of the user-matched regularized GD and HieDet training, we are able to capitalize on the stronger user channel powers for an equitable performance among the multiple users.

Future work in this domain involves improving on the robustness of the HieDet training to further close the gap to ML detection. In addition, we also envision extending this detection to higher order M-QAM constellations, addressing the challenges therin.

Chapter 4, in part, is a reprint of the material as it appears in Aditya Sant, and Bhaskar D. Rao. "Regularized Neural Detection for One-Bit Massive MIMO Communication Systems." arXiv e-prints (2023): arXiv-2305. and, in part, a reprint of the material as it appears in Aditya Sant, and Bhaskar D. Rao. "Regularized Neural Detection for Millimeter Wave Massive MIMO Communication Systems with One-Bit ADCs." IEEE ICASSP 2023. The dissertation author was the primary investigator and author of these papers.

Appendices

4.A Proof of Lemma 3

We begin by considering the expression for the one-bit likelihood gradient $\nabla_{\mathbf{x}}$. For any general channel, the gradient expression is given by

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

For further analysis, we absorb the positive scalers $c\sqrt{2\rho}$ into $\mathbf{G}$ for simpler notation. These constants can be added back in the end and bear no change on the subsequent analysis. We thus simplify the gradient expression as

$$\nabla_{\mathbf{x}} = \mathbf{G}^{\mathrm{T}}\,\sigma(-\mathbf{G}\mathbf{x}). \tag{4.24}$$

Using the real-valued notations and transformations for DNN implementation, we write out each term as follows. We begin by considering the argument of the sigmoid expression.

$$\mathbf{G}\,\mathbf{x} = \begin{bmatrix} \mathbf{Y}_{\mathrm{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} \mathbf{H}_{\mathrm{R}} & -\mathbf{H}_{\mathrm{I}} \\ \mathbf{H}_{\mathrm{I}} & \mathbf{H}_{\mathrm{R}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\mathrm{R}} \\ \mathbf{x}_{\mathrm{I}} \end{bmatrix} \tag{4.25}$$

Using this for the sigmoid expression we have

$$\sigma(-\mathbf{G}\,\mathbf{x}) = \begin{bmatrix} \sigma\big(-\mathbf{Y}_{\mathrm{R}}(\mathbf{H}_{\mathrm{R}}\mathbf{x}_{\mathrm{R}} - \mathbf{H}_{\mathrm{I}}\mathbf{x}_{\mathrm{I}})\big) \\ \sigma\big(-\mathbf{Y}_{\mathrm{I}}(\mathbf{H}_{\mathrm{R}}\mathbf{x}_{\mathrm{I}} + \mathbf{H}_{\mathrm{I}}\mathbf{x}_{\mathrm{R}})\big) \end{bmatrix} \tag{4.26}$$

Now we re-write the gradient expression using these analytical expressions as

$$\begin{aligned}
\nabla_{\mathbf{x}} &= \mathbf{G}^{\mathrm{T}}\,\sigma(\mathbf{G}\mathbf{x}) \\
&= \begin{bmatrix} \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{I}}^{\mathrm{T}} \\ -\mathbf{H}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{\mathrm{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_{\mathrm{I}} \end{bmatrix} \begin{bmatrix} \sigma\big(-\mathbf{Y}_{\mathrm{R}}(\mathbf{H}_{\mathrm{R}}\mathbf{x}_{\mathrm{R}} - \mathbf{H}_{\mathrm{I}}\mathbf{x}_{\mathrm{I}})\big) \\ \sigma\big(-\mathbf{Y}_{\mathrm{I}}(\mathbf{H}_{\mathrm{R}}\mathbf{x}_{\mathrm{I}} + \mathbf{H}_{\mathrm{I}}\mathbf{x}_{\mathrm{R}})\big) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{I}}^{\mathrm{T}} \\ -\mathbf{H}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{Y}_{\mathrm{R}}\sigma\big(-\mathbf{Y}_{\mathrm{R}}(\mathbf{H}_{\mathrm{R}}\mathbf{x}_{\mathrm{R}} - \mathbf{H}_{\mathrm{I}}\mathbf{x}_{\mathrm{I}})\big) \\ \mathbf{Y}_{\mathrm{I}}\sigma\big(-\mathbf{Y}_{\mathrm{I}}(\mathbf{H}_{\mathrm{R}}\mathbf{x}_{\mathrm{I}} + \mathbf{H}_{\mathrm{I}}\mathbf{x}_{\mathrm{R}})\big) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{I}}^{\mathrm{T}} \\ -\mathbf{H}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \tilde{\sigma}_{\mathrm{R}}(\mathbf{x}) \\ \tilde{\sigma}_{\mathrm{I}}(\mathbf{x}) \end{bmatrix}
\end{aligned} \quad . \tag{4.27}$$

We now write out the real-representation channel matrix as

$$\begin{bmatrix} \mathbf{H}_{\mathrm{R}} & -\mathbf{H}_{\mathrm{I}} \\ \mathbf{H}_{\mathrm{I}} & \mathbf{H}_{\mathrm{R}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathrm{R}} & -\mathbf{A}_{\mathrm{I}} \\ \mathbf{A}_{\mathrm{I}} & \mathbf{A}_{\mathrm{R}} \end{bmatrix} \begin{bmatrix} \bar{\alpha}_{\mathrm{R}} & -\bar{\alpha}_{\mathrm{I}} \\ \bar{\alpha}_{\mathrm{I}} & \bar{\alpha}_{\mathrm{R}} \end{bmatrix} \tag{4.28}$$

with $\bar{\alpha} = \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_K \end{bmatrix}$.

Based on this we have

$$\begin{aligned}
\mathbf{H}_{\mathrm{R}}^{\mathrm{T}} &= \bar{\alpha}_{\mathrm{R}}\mathbf{A}_{\mathrm{R}}^{\mathrm{T}} - \bar{\alpha}_{\mathrm{I}}\mathbf{A}_{\mathrm{I}}^{\mathrm{T}} \\
\mathbf{H}_{\mathrm{I}}^{\mathrm{T}} &= \bar{\alpha}_{\mathrm{R}}\mathbf{A}_{\mathrm{I}}^{\mathrm{T}} + \bar{\alpha}_{\mathrm{I}}\mathbf{A}_{\mathrm{R}}^{\mathrm{T}}
\end{aligned} \quad . \tag{4.29}$$

Using this in the expression for the gradient we simplify the gradient as

$$\begin{aligned}
\nabla_{\mathbf{x}} &= \begin{bmatrix} \nabla_{\mathrm{R}} \\ \nabla_{\mathrm{I}} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{I}}^{\mathrm{T}} \\ -\mathbf{H}_{\mathrm{I}}^{\mathrm{T}} & \mathbf{H}_{\mathrm{R}}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \tilde{\sigma}_{\mathrm{R}}(\mathbf{x}) \\ \tilde{\sigma}_{\mathrm{I}}(\mathbf{x}) \end{bmatrix} \\
&= \begin{bmatrix} \bar{\alpha}_{\mathrm{R}}\,\mathbf{t}(\mathbf{x}) + \bar{\alpha}_{\mathrm{I}}\,\mathbf{u}(\mathbf{x}) \\ \bar{\alpha}_{\mathrm{R}}\,\mathbf{u}(\mathbf{x}) - \bar{\alpha}_{\mathrm{I}}\,\mathbf{t}(\mathbf{x}) \end{bmatrix},
\end{aligned} \tag{4.30}$$

where we have

$$\begin{aligned}\mathbf{t}(\mathbf{x}) &= \mathbf{A}_{\mathrm{R}}^{\mathrm{T}}\,\tilde{\sigma}_{\mathrm{R}}(\mathbf{x}) + \mathbf{A}_{\mathrm{I}}^{\mathrm{T}}\,\tilde{\sigma}_{\mathrm{I}}(\mathbf{x}) \\ \mathbf{u}(\mathbf{x}) &= -\mathbf{A}_{\mathrm{I}}^{\mathrm{T}}\,\tilde{\sigma}_{\mathrm{R}}(\mathbf{x}) + \mathbf{A}_{\mathrm{R}}^{\mathrm{T}}\,\tilde{\sigma}_{\mathrm{I}}(\mathbf{x}).\end{aligned} \tag{4.31}$$

Putting all this together, we an write the complex valued gradient as follows,

$$\nabla_{\mathbf{x}} = \nabla_{\mathrm{R}} + j\,\nabla_{\mathrm{I}} = \left(\bar{\alpha}_{\mathrm{R}} + j\,\bar{\alpha}_{\mathrm{I}}\right)\left(\mathbf{t}(\mathbf{x}) + j\,\mathbf{u}(\mathbf{x})\right) \tag{4.32}$$

Chapter 5

DNN-aided Dequantization for Few-bit MIMO Detection

5.1 Introduction

Massive MIMO communications is integral for building a network of high-speed communications and interconnected devices[3,4]. A key component of the wireless RF and analog chain is the high-resolution and high-speed Analog-to-Digital Converter (ADC). However, this component significantly contributes to the overall system cost and complexity[23, 120]. To this end, few-bit ADCs, using two or more bits, enable balancing the tradeoff between high system cost and complexity, and excessive signal distortion, seen in one-bit ADCs. Additionally, with several advances in algorithm design and machine learning, research into robust communication system design on the backbone of low resolution ADCs has garnered increased interest[20–25].

The application of DNNs to wireless communication systems has greatly improved receiver performance and robustness. The general parametric structure of DNNs, coupled with their advantage as universal functional approximators [121, 122], makes them an integral part of the future of robust wireless communications. DNN-aided wireless communications are applied to beamformer design [52–54], channel estimation [123–125] as well as end-to-end detection [55–59]. In this work, we exploit this general parametric structure of DNNs to develop an iterative dequantizer for signal recovery from few-bit measurements.

5.1.1 Prior work

Few-bit MIMO systems have been extensively studied, from channel estimation systems to detector design. To begin with, the use of one-bit ADCs in multi-user MIMO for both channel estimation and data detection is of particular research interest. One-bit MIMO was first used for sensing and channel estimation algorithms[26–28]. Following this, data detection for one-bit MIMO gained a valuable advance with the application of Bussgang's theorem to linearize the input-output relation [126]. Through means of this relation, a class of Bussgang-based receivers was developed for detection from one-bit data [29–31], and extended to include more robust receiver structures for one-bit systems[32, 33].

The current state-of-the-art detection frameworks for one-bit MIMO detection is based on the maximum likelihood (ML) framework. The ML optimization, using the Gaussian cumulative distribution function (CDF), has been formulated in [34]. Utilizing this formulation the work in [35] introduced a near maximum likelihood (n-ML) detector. Other works applying the Gaussian CDF likelihood formulation have also been used extending this idea [130, 131]. The authors in [1] designed the detector, the OBMNet, an unfolded DNN, implementing the GD algorithm for approximate likelihood. The work in [20] improved on the OBMNet by introducing a learnable M-QAM projection over the GD iterations, the FBM-DetNet. Going beyond model-based methods, the different DNN-based receivers have also been designed to using the ML optimization and unrolling the resulting GD algorithm [36, 133, 138]. The work in [156] analytically addresses ML detection for one-bit MIMO, as well as incorporating a faster accelerated GD-based detector using an unfolded DNN, i.e., the A-PrOBNet. Other DNNs for one-bit MIMO detection, not relying on the likelihood framework have also been developed [134–137].

Going beyond one-bit, few-bit ADCs have also gained increased interest for channel estimation and pilot design algorithms[21,22,157–159]. Data detection for few-bit MIMO systems is also a heavily researched area in building robust MIMO systems[33, 160–163]. The current state-of-the-art few-bit MIMO detector is the FBM-DetNet[20]. This extends the ML framework, derived for one-bit MIMO to the multi-bit case, unfolding the resultant GD algorithm as a T-layer DNN. However, to the best of the authors' knowledge, no work has addressed the functional mapping from the discrete M-QAM transmitted signal space to the quantized received signal space. This approach, i.e., signal dequantization forms the basis of the analysis in this work.

5.1.2 Contributions of this work

The detector design in this work utilizes a DNN-aided framework to dequantize the signal, In particualr, the following enlist the main contributions of this work.

1. *Role of quantizer design:* This work details the impact of the different parameters of the few-bit quantizer in the signal detection. In particular, the tradeoff for resolution vs clipping in signal acquisition is further elaborated.

2. *Insights into signal dequantization:* The functional mapping from the discrete transmitted M-QAM space to the received quantized measurements is elucidated. This analysis enables the formulation of a probability of perfect dequantization, and the role of massive MIMO systems on such systems.

3. *Baseline projected ZF detection:* An initial baseline approach for signal dequantization is introduced using the zero forcing (ZF) based projection for ideal unquantized measurements. This is based the procedure for alternating optimization for signal recovery, and also serves as a lower bound on the performance evaluation for the signal recovery.

4. *DNN-aided iterative dequantizer:* A DNN-aided framework is presented to learn a general dequantization mapping, beginning with the role of such DNN-aided systems in signal dequantization. The network, DQuantNet, is implemented as a T-block unfolded DNN, with each block implementing a learnt dequantization procedure to iteratively estimate the unquantized measurements.

The experimental results in this work compare the proposed DNN-aided framework to the current state-of-the art detection for few-bit MIMO. The presented results show this to be a promising approach for detector design using dequantization.

Organization: This work is organized as follows. Section 5.2 introduces the system model, quantizer design and the resulting optimization for signal detection. Following this, Sec. 5.3 provides insights into the role of the quantizer design, the general dequantization process, and concludes by introducing the projected ZF detector. Improving on the projected ZF detector, Sec. 5.4 details the general DNN-aided dequantizer, along with the implementation as an unfolded DNN. The experimental validation is provided in Sec. 5.5 and Sec. 5.6 concludes the paper.

Notation: We use lower-case boldface letters $\mathbf{a}$ and upper case boldface letters $\mathbf{A}$ to denote complex valued vectors and matrices respectively. The notation $\mathfrak{Re}(\cdot)$ and $\mathfrak{Im}(\cdot)$ denote the real and imaginary parts, respectively. The operation $(\cdot)^{\mathrm{T}}$ denotes the transpose of the array or matrix. Unless otherwise specified,

all scalar functions like tanh(·) or sign(·), when applied to arrays or matrices, imply element-wise operation. The diagonalization operator, denoted by diag(·), when applied to an array $\mathbf{a}$, creates a diagonal matrix with the diagonal entries given by $\mathbf{a}$. The notation $\mathbf{x}^{(t)}$ is used to denote the value of the variable $\mathbf{x}$ at iteration t of the algorithm. For the DNN training, the size of the training set is given by N_{train} and the notation $\hat{\mathbf{x}}_{n,\text{train}}$ denotes the n^{th} sample from this set. Unless otherwise specified, the norm $||\cdot||$ represents the ℓ_2-norm for a vector or matrix.

5.2 System Model and Few-bit Likelihood

This section begins by introducing the multi-user wireless uplink system model. This is followed by the signal quantization model. Finally the optimization framework is presented for the signal detection.

5.2.1 Uplink wireless system model

The Rayleigh fading channel with block flat-fading, as used in most past works, e.g. [12, 139] is utilized here. The K single antenna users transmit to a multi-antenna base-station (BS) with N receive antennas. The MIMO channel $\bar{\mathbf{H}} \in \mathbb{C}^{N\times K}$ consists of i.i.d entries drawn from $\mathcal{CN}(0,1)$. This work assumes perfect unquantized channel state information (CSI) at the BS.

As a part of the multi-user uplink, the k^{th} user transmits the signal $\bar{x}_k$ drawn from the M-QAM constellation. The multi-user transmitted signal is $\bar{\mathbf{x}} = \begin{bmatrix}\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_K\end{bmatrix}^{\text{T}}$. Additionally, the M-QAM symbol vector is normalized such that $\mathbb{E}\,||\bar{\mathbf{x}}||^2 = K$ The unquantized received signal at the BS is

$$\bar{\mathbf{r}} = \bar{\mathbf{H}}\bar{\mathbf{x}} + \bar{\mathbf{n}}, \tag{5.1}$$

where $\bar{\mathbf{n}}$ is the AWCGN with noise variance depending on the system signal-to-noise ratio (SNR) $\rho = \frac{\mathbb{E}(||\bar{\mathbf{H}}\bar{\mathbf{x}}||^2)}{\mathbb{E}(||\bar{\mathbf{n}}||^2)}$.

In order to express the algorithm design as a function of real-valued inputs, we convert the received signal and the observed channel matrix into real-valued forms as

$$\begin{gathered}\mathbf{H} = \begin{bmatrix}\mathfrak{Re}(\bar{\mathbf{H}}) & -\mathfrak{Im}(\bar{\mathbf{H}}) \\ \mathfrak{Im}(\bar{\mathbf{H}}) & \mathfrak{Re}(\bar{\mathbf{H}})\end{bmatrix},\ \mathbf{x} = \begin{bmatrix}\mathfrak{Re}(\bar{\mathbf{x}}) \\ \mathfrak{Im}(\bar{\mathbf{x}})\end{bmatrix}, \\ \mathbf{r} = \begin{bmatrix}\mathfrak{Re}(\bar{\mathbf{r}}) \\ \mathfrak{Im}(\bar{\mathbf{r}})\end{bmatrix},\ \mathbf{n} = \begin{bmatrix}\mathfrak{Re}(\bar{\mathbf{n}}) \\ \mathfrak{Im}(\bar{\mathbf{n}})\end{bmatrix}.\end{gathered} \tag{5.2}$$

The received unquantized signal at the BS is thus given by

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{5.3}$$

However, the BS does not have access to this unquantized signal. The received signal (5.3) is passed through the b-bit quantizer at the BS. This is further elaborated in the subsequent subsection.

5.2.2 Signal quantization

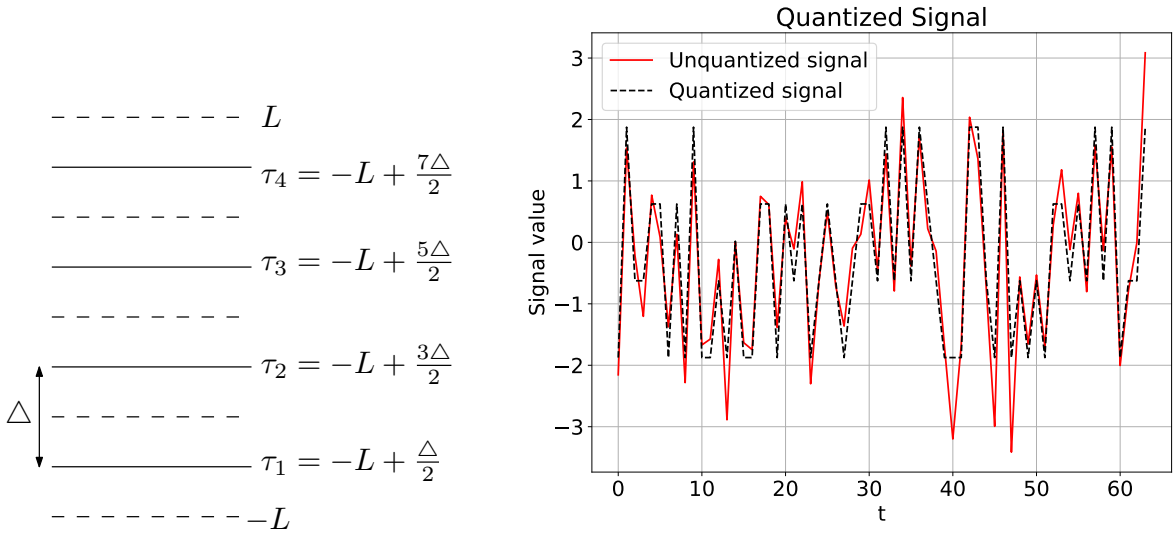


Figure 5.1: Illustration of 2-bit quantizer design and signal quantization. (a) Quantizer levels shown wrt the saturation limits $[-L, L]$ (b) Illustrating the quantized signal output for $L = 2.5$ and input signal from 32×4 MU-MIMO system with QPSK symbols at SNR $= 30$ dB.

At the BS, the b-bit quantizer is characterized by the quantizer saturation limits[1] $[-L, L]$. The different quantizer levels and inter-level spacing are given by $\tau_l = -L + (2l-1)\frac{\triangle}{2}$ and $\triangle = \frac{2L}{2^b}$, respectively, for $l = 1, 2, \ldots, 2^b$. Based on this, the element-wise b-bit quantization at the BS is

$$\mathbf{y}[i] = \mathcal{Q}_b(\mathbf{r}[i]), \forall\, i = 1, 2, \ldots, N \tag{5.4a}$$

$$= \begin{cases} \tau_l, \text{ if } \tau_l - \frac{\triangle}{2} \leq \mathbf{r}[i] < \tau_l + \frac{\triangle}{2},\ l \neq \{1, 2^b\} \\ \tau_1, \text{ if } \mathbf{r}[i] < \tau_1 + \frac{\triangle}{2} \\ \tau_{2^b}, \text{ if } \mathbf{r}[i] \geq \tau_{2^b} - \frac{\triangle}{2}. \end{cases} \tag{5.4b}$$

The quantizer levels and quantized signal output are shown in Fig. 5.1 for a two-bit quantizer. Fig. 5.1(a)

[1] Values of the input signal for which there is no clipping in the quantizer

shows the quantizer levels with respect to the saturation limits $[-L, L]$. As seen from the plots in Fig. 5.1(b), the signal undergoes amplitude clipping for certain values. This is dependent on the choice of the quantizer limits L. The design of the quantizer limits for a given signal analyzed further in Sec. 5.3.1.

5.2.3 Optimization for received signal detection

We represent the received quantized signal (5.4) via the additive quantization noise model given as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} + \mathbf{e}. \tag{5.5}$$

Here the quantization noise $\mathbf{e}$ is not independent of the received signal $\mathbf{r}$. This specifically exploits the fact that the few bit signal slightly deviates from the linear system model and approaches linearity with increasing quantizer bits. Hence, this can be utilized to iteratively dequantize the observed quantized measurements. The linear MSE detection (for the unqunatized case) can be appropriately modified to detect the signal from the quantized measurements. In particular, the both the transmitted symbols $\mathbf{x}$ and quantization noise $\mathbf{e}$ are jointly estimated. This resulting optimization is presented as

$$[\mathbf{x}^*, \mathbf{e}^*] = \underset{\mathbf{x}\in\mathcal{M}^{2K}, \mathbf{e}\in\mathcal{E}^{2N}}{\operatorname{argmin}} ||\mathbf{y} - \mathbf{H}\mathbf{x} - \mathbf{e}||^2, \tag{5.6}$$

where $\mathcal{M}^{2K}$ represents the set of the $2K$-dimensional vectors, consisting of the real-valued representation (see eq. (5.2)) of the K-dimensional vectors of M-QAM symbols. The set $\mathcal{E}^{2N}$ represents the set of quantization noise vectors. Clearly, this constrained optimization is not convex. This work addresses solving this detection problem via the alternating optimization framework. The insights into this MIMO detection, along with the proposed DNN-aided dequantizer, are presented in the subsequent sections.

5.3 Insights into Few-bit MIMO Detection

This section begins by considering the different factors affecting quantizer design in the MIMO RF chain. This is followed by the subspace analysis of the quantized signal and the subsequent dequantization of this quantized signal. Finally, a baseline ZF detector is presented, that forms the basis for the improved DNN-aided dequantizer.

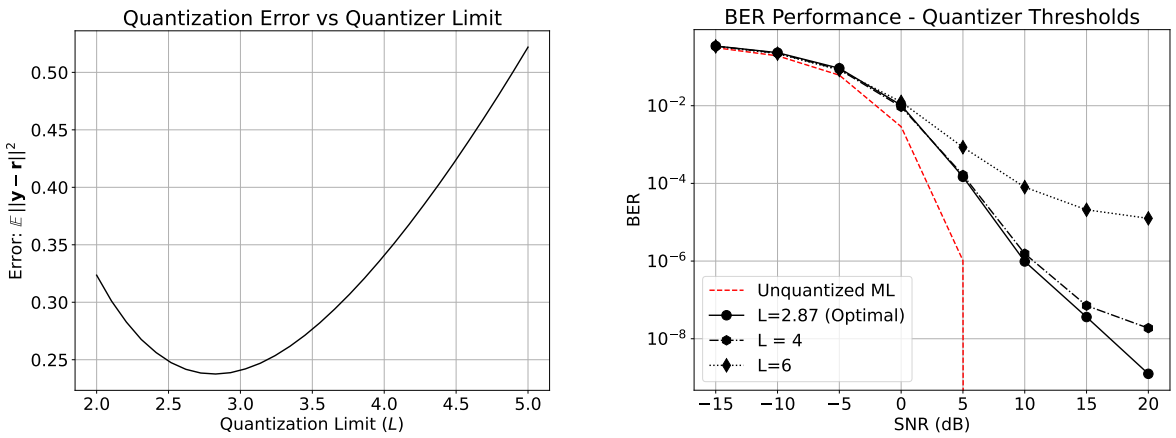


Figure 5.2: Illustrating the resolution vs clipping tradeoff for a 4×32 MU-MIMO with each user transmitting QPSK symbols at SNR $= 30$ db (a) Quantization error $||\mathbf{y}-\mathbf{r}||^2$ vs the quantizer limit L (b) Performance of proj-ZF detection baseline (Algorithm 7) for different quantizer limits

5.3.1 Impact of quantizer design on detection

The parameters of the quantizer influence the best possible performance for the few-bit MIMO detector. Tuning the quantizer (before detection) to ensure optimal performance requires information about the received signal power, gain control at the receiver and the quantizer tradeoff between clipping and increased signal resolution. This is elaborated below.

Unquantized received signal power

Consider the unquantized received signal (5.3) that is input to the quantizer. The signal power per antenna element is given by $\mathbb{E}\,||\mathbf{r}[i]||^2 = \mathbb{E}\,||\mathbf{h}_i^{\mathrm{T}}\mathbf{x}||^2 + \sigma_{\mathbf{z}}^2$, where $\mathbf{h}_i$ is the i^{th} row of the channel matrix $\mathbf{H}$. This received signal power clearly depends on *(i)* The number of users K, *(ii)* The power of transmitted symbols per user as well as the modulation order of the symbols, and *(iii)* The noise variance $\sigma_{\mathbf{z}}^2$, which is a function of the SNR.

Signal gain control

At the BS receiver the ADC scales the received unquantized signal, using an automatic gain control (AGC), to ensure that the input signal to the ADC remains within an optimal range to avoid clipping and nonlinear distortion. However, for ease of analysis in this work (analyzing received signal SNR), the power control is realized at the transmitter side. This gain control is realized through scaling of the transmitted symbols, i.e., $\mathbb{E}||\mathbf{x}||^2 = p\,K$, with p as the per-user transmit power, ensuring same power is transmitted by the UE, invariant of modulation. The value of p can be adjusted to control the received signal power at

the BS. Without loss of generality, in the subsequent analysis and algorithm development in this work, the K-user transmitted symbol vector is scaled such that $\mathbb{E}\,||\mathbf{x}||^2 = K$.

Resolution vs clipping tradeoff

In addition to the signal scaling, the quantizer limit L also plays an important role in system performance. Once the power control for the received signal has been fixed, the quantizer parameter can be independently tuned to control allowable signal clipping. For a given received unquantized signal $\mathbf{r}$, a lower value of L ensures better resolution, but increases the signal clipping. Conversely, increasing L will reduce the signal clipping, at the expense of reduced signal resolution at lower magnitudes. This is empirically shown by analyzing the quantization error of a 2-bit quantizer for the 4-user QPSK transmission at 30 dB SNR in Fig. 5.2(a). This plot shows that an appropriately chosen $L = 2.87$ correctly balances this resolution-clipping tradeoff.

This choice of the quantizer limit L is further reinforced through the performance plots illustrated in Fig. 5.2(b). The BER performance for the projected ZF baseline, explained in 5.3.3. The maximum likelihood (ML) detection for the unquantized signal is the lower limit on the bit error rate (BER). As seen from these plots, by increasing L beyond the optimal, the performance deteriorates due to lower resolution at lower signal amplitudes. The plots in Fig. 5.2 illustrate that for optimal signal detection performance, the quantizer limit L should be chosen to balance the resolution vs clipping tradeoff for the received unquantized signal.

Remark. *In practical system deployment the quantizer performance and efficiency is affected the received unquantized signal power. Thus, the quantizer limit L and the transmitted symbol power $\mathbb{E}\,||\mathbf{x}||^2$ should be jointly optimized to balance the resolution vs clipping tradeoff. However, for ease of analysis in this work, we assume an ideal quantizer that can operate at any input power level. To this end, the symbol powers are kept fixed, and the quantizer limit L is tuned to balance this resolution vs clipping tradeoff.*

Quantizer parameters for subsequent analysis

Based on the performance plots in Fig. 5.2, the optimum two-bit quantizer limit to balance the resolution vs clipping tradeoff is chosen as $L = 2.87$. The subsequent DNN-aided detection algorithms are trained and tested using this value of the quantizer limit.

Remark. *The choice of the quantizer parmeters, like the quantizer limit L, can also be tuned based on the SNR. The SNR affects the percentage occupancy of the different quantization levels differs with amount*

of noise added, attributed to increased received signal variance due to the AWGN. Considering this, the quantizer design and receiver algorithms can be tuned for different SNR values separately. However, this SNR-dependent analysis falls outside the scope of this current work.

Following the quantizer design, the subsequent subsections delve into transmitted signal recovery algorithms and DNN-aided methods via signal dequantization.

5.3.2 Signal dequantization

Signal recovery from quantized measurements involves inverting a nonlinear transformation $\mathbf{y} = f(\mathbf{x}) : \mathbb{R}^{2K} \to \mathbb{R}^{2N}$, as seen in eqs. (5.3)-(5.4). However, the specific properties of the input and output sets, as well as the nature of the transformation $f(\cdot)$ enables establishing bounds on transmitted symbol recovery from received measurements $\mathbf{y}$. This subsection delves deeper into the framework for evaluating these recovery bounds using received signal dequantization.

Defining the sets and quantities

The following are the different sets of interest in this analysis, defined for a given $\mathbf{H} \in \mathbb{R}^{2N \times 2K}$ and a b-bit ADC $\mathcal{Q}_b(\cdot)$:

$$\mathcal{Y} = \{\mathbf{y} \,|\, \mathbf{y} = \mathcal{Q}_b(\mathbf{r}), \forall\, \mathbf{r} \in \mathbb{R}^N\} \tag{5.7a}$$

$$\mathcal{Y}_{\mathbf{H}} = \{\mathbf{y} \,|\, \mathbf{y} = \mathcal{Q}_b(\mathbf{r}), \forall\, \mathbf{r} \in \mathcal{R}(\mathbf{H})\} \tag{5.7b}$$

$$\mathrm{R}_{\mathrm{QAM}} = \{\mathbf{r} \,|\, \mathbf{r} = \mathbf{H}\mathbf{x},\, \mathbf{x} \in \mathcal{M}^K\} \tag{5.7c}$$

$$\mathcal{Y}_{\mathrm{QAM}} = \{\mathbf{y} \,|\, \mathbf{y} = \mathcal{Q}_b(\mathbf{r}), \forall\, \mathbf{r} \in \mathrm{R}_{\mathrm{QAM}}\}. \tag{5.7d}$$

The cardinality of these discrete sets (5.7) plays a key role in the further analysis. These quantities are given by

$$|\mathcal{Y}| = 2^{bN} \tag{5.8a}$$

$$|\mathcal{Y}_{\mathbf{H}}| = N_{\mathbf{H}} \leq 2^{bN} \tag{5.8b}$$

$$|\mathrm{R}_{\mathrm{QAM}}| = M^K \tag{5.8c}$$

$$|\mathcal{Y}_{\mathrm{QAM}}| = N_{\mathbf{H},\,\mathrm{QAM}} \leq N_{\mathbf{H}}. \tag{5.8d}$$

Note that, barring $\mathrm{R}_{\mathrm{QAM}}$, the cardinality of all other sets will increase with increasing N, due to the increasing dimension of the subspace of received quantized measurements $\mathbf{y}$.

Illustration of the different sets

In this analysis, a low dimensional 2-bit quantized MIMO model is considered to pictorially illustrate the different sets in (5.7), as well as the role of quantizer design, discussed in the earlier sub-section. For this example, consider the real-valued problem such that $\mathbf{H} \in \mathbb{R}^{N \times 1}$, $M = 4$ and $b = 2$. This is pictorially shown in Fig. 5.3.

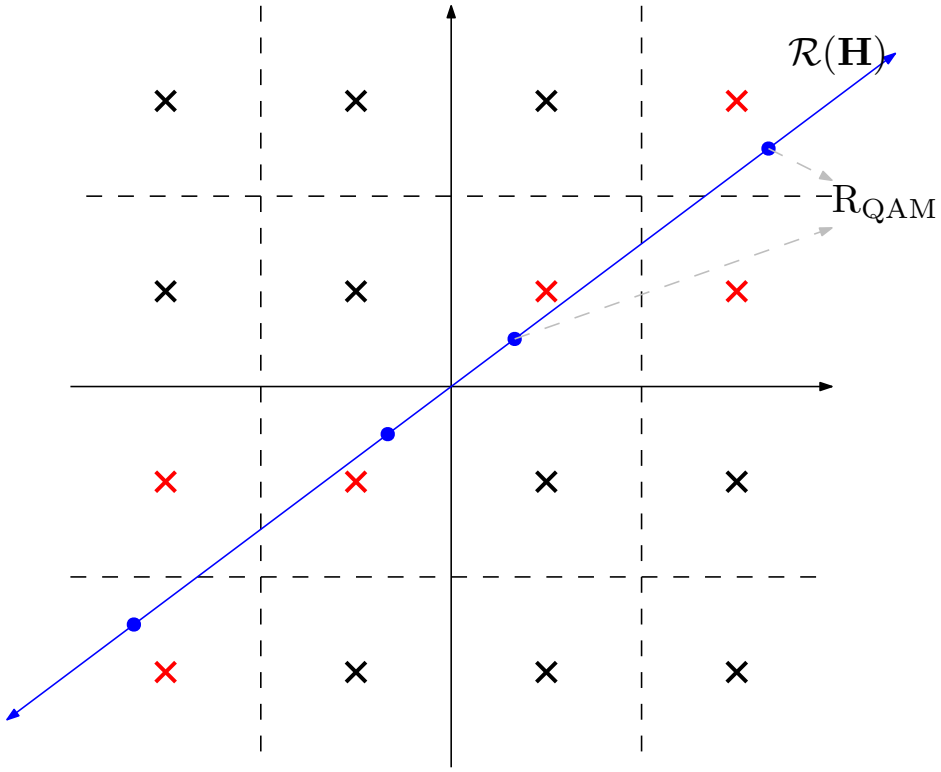


Figure 5.3: Illustration for the 2-D quantized space with the different sets (5.7)

Based on the illustrated subspace analysis in Fig. 5.3, the different square grids partition $\mathbb{R}^2$ using 2-bit quantization along each axis. The centroid of each square partition, i.e., the quantized value, is represented using the 'x' symbol. For a given channel matrix $\mathbf{H}$, the blue line passing through the origin represents $\mathcal{R}(\mathbf{H})$. The different grid squares that are intersected by $\mathcal{R}(\mathbf{H})$ are part of the set $\mathcal{Y}_{\mathbf{H}}$, represented by the red crosses. The discrete set $\mathrm{R}_{\mathrm{QAM}}$ is the set of blue dots on the line $\mathcal{R}(\mathbf{H})$. The set $|\mathcal{Y}_{\mathrm{QAM}}|$ will be a subset of $\mathcal{Y}_{\mathbf{H}}$, with $N_{\mathbf{H}} = 6$ and $N_{\mathbf{H},\mathrm{QAM}} = 4$.

Resolution vs clipping tradeoff: Since the sets $\mathcal{R}(\mathbf{H})$ and $\mathrm{R}_{\mathrm{QAM}}$ are independent of the quantized

grid space, the quantizer parameter L, controlling the sizes of the squares in the grid, determines the set $\mathcal{Y}_{\text{QAM}}$. From the illustration in Fig. 5.3, it is evident that a very large value of L, or a very small value of L will result in the selection of the two quantized centroids close to the origin, or at the edges of the grid, respectively. This is the resolution vs clipping tradeoff seen in Sec. 5.3.1. Thus, the optimum choice of the quantizer limit L balances this tradeoff by maximizing the overlap between the sets $\mathcal{Y}_{\mathbf{H}}$ and $\mathcal{Y}_{\text{QAM}}$. This is expressed via the optimization

$$L^* = \underset{L \in \mathbb{R}^+}{\text{argmin}} \, \mathbb{E}_{\mathbf{H}} \, |N_{\mathbf{H}} - N_{\mathbf{H},\text{QAM}}|. \tag{5.9}$$

The above analytically ensures that the discrete set R_{QAM} is evenly spread over all the intersecting quantization grid points. The optimization (5.9) is evaluated by averaging over different channel realization, for a given M-QAM setup. However, owing to the intractable nature of the quantities $N_{\mathbf{H}}$ and $N_{\mathbf{H},\text{QAM}}$, for a general $\mathbf{H}$, the optimum quantizer limit L is empirically evaluated using the quantization error minimization, as given in 5.3.1. All subsequent analysis is developed for this optimal value of the quantizer limit L.

This analysis is now extended to the general N-dimensional case; using the definitions in (5.7) and (5.8), the general probability of perfect dequantization can be evaluated.

Towards perfect dequantization

The few-bit quantization (5.3)-(5.4) presents a nonlinear functional mapping from the set $\mathcal{M}^{2K}$ to $\mathcal{Y}_{\text{QAM}}$. This is invertible, if this mapping is one-one. Therefore, the probability for perfect dequantization $\text{P}_{\text{dequant}}$ is the probability that this mapping is one-one. We make an important assumption for the ensuing analysis. Specifically, the number of antennas is chosen large enough, such that $N_{\mathbf{H}} > M^K$. This is possible since $N_{\mathbf{H}}$ increases with increasing N.

The probability $\text{P}_{\text{dequant}}$ can be evaluated using the combinatorial method of putting M^K balls in $N_{\mathbf{H}}$ boxes, such that each box has at most 1 ball. This implies that there is a unique one-one mapping between the observed discrete unquantized values $\mathbf{r} \in \text{R}_{\text{QAM}}$ and quantized observations $\mathbf{y} \in \mathcal{Y}_{\mathbf{H},\text{QAM}}$. This is evaluated as

$$\text{P}_{\text{dequant}}(N_{\mathbf{H}}) = \frac{N_{\mathbf{H}} * (N_{\mathbf{H}} - 1) \ldots (N_{\mathbf{H}} - M^K + 1)}{(N_{\mathbf{H}})^{M^K}}. \tag{5.10}$$

Before highlighting the utility of evaluating this probability of dequantization, two specific properties of signal dequantization are shown.

Firstly, it can be shown that (5.10) is a monotonically increasing function of N_H. To show this,

consider two values M_1 and M_2, such that $M_1 < M_2$. We can then show

$$\begin{aligned}\frac{\mathrm{P}_{\text{dequant}}(M_2)}{\mathrm{P}_{\text{dequant}}(M_1)} &= \prod_{i=0}^{M^K-1} \Big(\frac{M_2 - i}{M_1 - i}\Big)\frac{M_1}{M_2} \\ &= \prod_{i=0}^{M^K-1} \frac{M_1 M_2 - M_1 i}{M_1 M_2 - M_2 i} \\ &= \prod_{i=0}^{M^K-1} \Big(1 + \frac{(M_2 - M_1)i}{M_2(M_1 - i)}\Big) \\ &> 1.\end{aligned} \tag{5.11}$$

Clearly, as $N \to \infty$, $\mathrm{P}_{\text{dequant}}(N_{\mathbf{H}}) \to 1$; this provides an analytical understanding for motivating the use of massive MIMO for few-bit systems.

Secondly, we can upper bound the dimensionality of the subspace, to ensure that dequantization is definitely not possible. This will occur if,

$$\begin{aligned}M^K &> \text{supp}(N_{\mathbf{H},K}) \\ M^K &> 2^{bN} \\ K &> \frac{bN}{\log_2 M}.\end{aligned} \tag{5.12}$$

This restricts both the number of users as well as the modulation order for a given MIMO system.

The significance of evaluating the probability or dequantization is elaborated below.

- The probability of dequantization connects the quantizer parameters with the parameters of the transmitted constellation symbols.
- This metric characterizes the effect of the number of MIMO antennas on the possible performance of the few-bit MIMO detection algorithm, establishing the probability of a one-one inverse mapping to the M-QAM subspace $\mathcal{M}^{2K}$.
- Conversely, the minimum number of BS antennas can be selected to achieve a desired dequantization probability; this enables resource-efficient design of few-bit MIMO systems.

This analysis is a probability of one-one mapping only, and does not provide insights for effective algorithms for dequantization. Different detectors can be utilized to implement this operation. We begin by describing the baseline detection strategy, which will be refined further to learn an efficient dequantization-based detector.

Remark. *The above analysis and probability for perfect dequantization relies on knowing $N_{\mathbf{H}}$ for any channel $\mathbf{H} \in \mathbb{R}^{2N \times 2K}$. The computation of these bounds, although relevant for evaluating the dequantization probabilities, falls outside the scope of this work. Here, we primarily focus on the detection algorithm design for few bit MIMO. The detailed analysis on the evaluating this probability analytically or numerically is left for future work.*

5.3.3 Projected ZF detection

The projected ZF detection algorithm is based on solving the joint optimization (5.6) using alternating optimization. Intuitively, this consists of two main steps *(i)* Update the estimate of the transmitted symbols using the current estimate of the dequantized value, *(ii)* Update the dequantized value using the current estimate of the transmitted symbols.

Alternating optimization for projected ZF

This iterative approach successively updates the values of each parameter as follows.

1. Updating $\mathbf{x}^{(t)}$

$$\mathbf{x}_{\mathrm{ZF}}^{(t)} = \mathbf{H}^{\dagger}(y - \mathbf{e}^{(t-1)}) \tag{5.13a}$$

$$\mathbf{x}^{(t)} = \mathcal{P}_{\mathcal{X}}(\mathbf{x}_{\mathrm{ZF}}^{(t)}). \tag{5.13b}$$

2. Updating $\mathbf{x}^{(t)}$

$$\mathbf{e}_{\mathrm{ZF}}^{(t)} = \mathbf{y} - \mathbf{H}\mathbf{x}^{(t)} \tag{5.14a}$$

$$\mathbf{e}^{(t)} = \mathcal{P}_{\mathcal{E}}(\mathbf{e}_{\mathrm{ZF}}^{(t)}). \tag{5.14b}$$

This alternating update strategy involves successively refining the estimate of the quantization noise $\mathbf{e}^{(t)}$ to get a better estimate of the current unquantized signal $\mathbf{r}^{(t)} = y - \mathbf{e}^{(t)}$. This, in turn, is used to get the current MMSE estimate of $\mathbf{x}^{(t)}$ using the current estimate of the unquantized signal. This is used to again re-estimate the quantization noise, and so on.

Projection operators

The projection operators (5.13b) and (5.14b) are used to incorporate the constraints $\mathbf{x} \in \mathcal{M}^{2K}$ and $\mathbf{e} \in \mathcal{E}^{2N}$, respectively, in the joint optimization (5.6). Each of these is further elaborated below.

1. Projection of $\mathbf{x}^{(t)}$: The Gaussian projection is utilized for projecting the ZF estimate onto the symbol space $\mathcal{M}^{2K}$, as seen in the previous works [156]. Following the same theoretical framework as seen in this work, the estimated symbol at the t^{th} iteration, $\mathbf{x}_{\text{ZF}}^{(t)}$, is modeled as the ideal M-QAM symbol vector $\mathbf{s}$ with a Gaussian noise $\triangle\mathbf{s}$. Based on the uniform distribution over all the symbols, the Gaussian denoiser is derived as the posterior mean $\mathbb{E}_{\mathbf{s}|\mathbf{x}}(\mathbf{s})$. Thus, the projection is derived as

$$\mathcal{P}_{\mathcal{X}}(\mathbf{x}_{\text{ZF}}^{(t)}) = c^{(t)} \sum_{i=1}^{M^K} \mathbf{s}_i \exp\Big(- \frac{||\mathbf{x}_{\text{ZF}}^{(t)} - \mathbf{s}_i||^2}{(\sigma^{(t)})^2} \Big), \tag{5.15}$$

   where $\mathbf{s}_i$ is the i^{th} element in the set $\mathcal{M}^{2K}$, and $c^{(t)} = \Big(\sum_{i=1}^{M^K} \exp\Big(- \frac{||\mathbf{x}_{\text{ZF}}^{(t)} - \mathbf{s}_i||^2}{2(\sigma^{(t)})^2} \Big) \Big)^{-1}$. The term $\sigma^{(t)}$ is a projection hyperparameter, denoting the confidence of symbol estimation.

2. Projection of $\mathbf{e}^{(t)}$: Different from the M-QAM symbol set $\mathcal{M}^K$, the noise distribution for few bit systems cannot be accurately analyzed. This is also attributed to the clipping vs resolution tradeoff for the quantizer design. The presence of signal clipping prevents bounding all the samples of the quantization noise, and the exact distribution of the bounded noise and clipped components is intractable. Considering these constraints, and utilizing empirical testing of different projection approaches, the identity operator is chosen for stable recovery. Thus the quantization noise projection is given by

$$\mathcal{P}_{\mathcal{E}}(\mathbf{e}_{\text{ZF}}^{(t)}) = \mathbf{e}_{\text{ZF}}^{(t)}. \tag{5.16}$$

The entire projected ZF algorithm is given in Algorithm 7. This algorithm forms the baseline detector benchmark for this work. The subsequent section improves on this baseline algorithm, by utilizing a DNN-aided iterative dequantizer.

Remark. *With regards to the projection $\mathcal{P}_{\mathcal{X}}(\cdot)$ in* (5.15)*, the approach in [156] utilizes a learnable denoiser (learning the variance parameters $\{\sigma^{(t)}\}_{t=1}^{T}$) for for detection in one-bit MIMO systems. However, for this work, utilizing signal dequantization-based detection, both the learnable denoiser as well as the statically chosen denoiser are observed to have the same performance. Hence, a statically chosen denoiser is utilized, for ease of implementation.*

Algorithm 5: Projected ZF Algorithm

Input: T, $\mathbf{H}$, $\mathbf{e}^{(0)} = 0$, $\{\sigma^{(t)}\}_{t=1}^{T}$
Output: $\mathbf{x}^{(T+1)}$

for $t = 1$ **to** T **do**
  Evaluate $\mathbf{x}_{\text{ZF}}^{(t)}$ using (5.13a);
  Project $\mathbf{x}_{\text{ZF}}^{(t)}$ using (5.15) to get $\mathbf{x}^{(t)}$;
  Evaluate $\mathbf{e}_{\text{ZF}}^{(t)}$ using (5.14a) & update $\mathbf{e}^{(t)} \leftarrow \mathbf{e}_{\text{ZF}}^{(t)}$;
end
Evaluate $\mathbf{x}_{\text{ZF}}^{(T+1)}$ using (5.13a);
Project $\mathbf{x}_{\text{ZF}}^{(T+1)}$ using (5.15) to get $\mathbf{x}^{(T+1)}$;

5.4 DNN-aided Iterative Dequantizer

This section begins by motivating the need for DNN-aided methods for solving this dequantization problem. Following this, the general learnable iterative dequantizer framework is presented. Finally, the specific DNN-based implementation, i.e., DQuantNet, is elaborated for this detector design.

5.4.1 Motivating DNN-aided detection

Building on the projected ZF baseline Algorithm 7, a DNN-aided detector is introduced here, motivated by two main aspects: *(i)* Addressing the limitations of projected ZF, and *(ii)* Learning a nonlinear dequantization mapping. This is elaborated below.

Limitations of the baseline projected ZF algorithm

Algorithm 7 introduces the framework for iteratively fine-tuning the estimate of the quantization noise. However, this approach faces a few limitations

- *Projecting* $\mathbf{e}_{\text{ZF}}^{(t)}$*:* As stated earlier, the intractability in analyzing the quantization noise statistics poses challenges for the design of this quantization noise projection. The identity projection operation (5.16) has been selected for stable recovery, based on empirical testing. However, this can be further improved by data-driven methods.

- *Higher SER for higher order M-QAM:* The Gaussian projection (5.15) is empirically observed to aggressively reduce symbol cluster spread, resulting in hard thresholding of symbols at each iteration. This results in recovered constellation clusters with low cluster spread, but doesn't correct symbol errors efficiently. This is unfavorable from a MIMO detection point of view. Thus, there is a need to balance

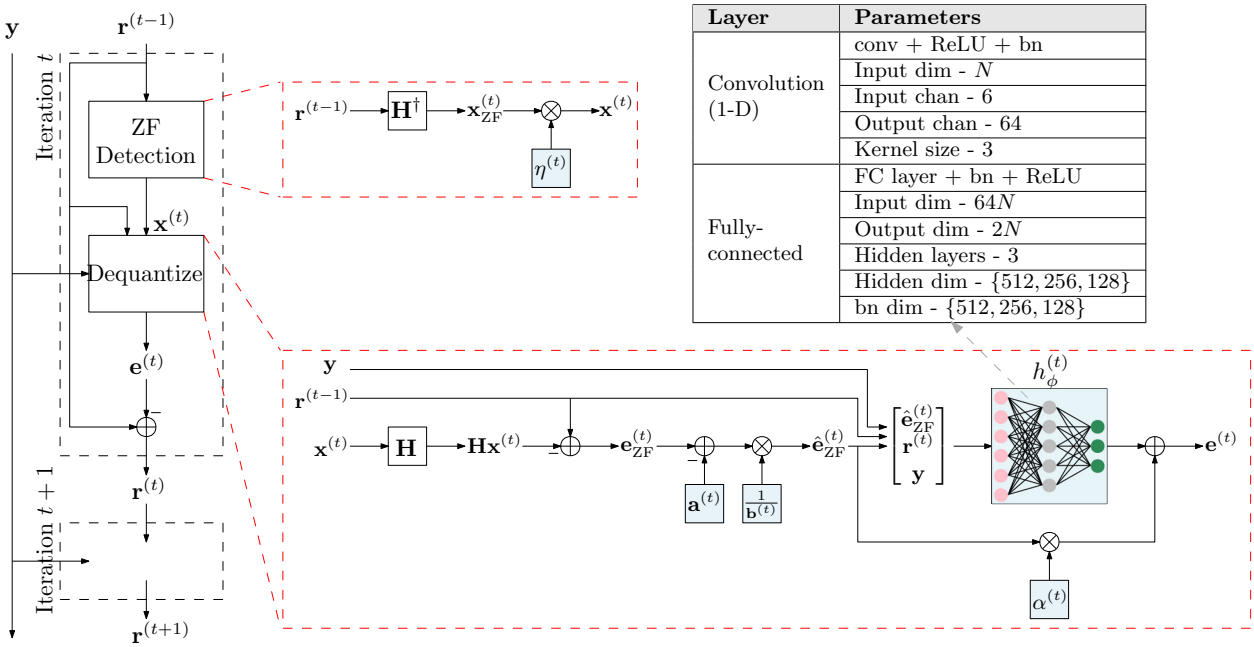
symbol error correction and M-QAM cluster spread reduction over iterations for the dequantization process.

Both these limitations can be effectively addressed through the use of a DNN-aided methods, as will be explained in the next subsection.

Learning dequantization

As stated earlier, we presented the conditions and probability for one-one mapping between the discrete M-QAM space $\mathcal{M}^{2K}$ and the observed quantized measurements $\mathcal{Y}_{\text{QAM}}$. However, the probability of dequantization does not establish any methods for executing the inverse process. Inverting the quantization operation is a nonlinear intractable function $f(\cdot) : \mathcal{Y}_{\text{QAM}} \to \mathcal{M}^{2K}$, i.e., the dequantization function. The general parametric nature of DNNs and the ability for universal functional approximation [121] makes them ideally suited to model and learn this dequantization function.

5.4.2 Iteratively learning the dequantizer



Layer	Parameters
Convolution (1-D)	conv + ReLU + bn
	Input dim - N
	Input chan - 6
	Output chan - 64
	Kernel size - 3
Fully-connected	FC layer + bn + ReLU
	Input dim - $64N$
	Output dim - $2N$
	Hidden layers - 3
	Hidden dim - $\{512, 256, 128\}$
	bn dim - $\{512, 256, 128\}$

Figure 5.4: Block diagram of the T-stage DQuantNet. Within each stage, the shaded blue boxes represent the learnable parameters. The accompanying table in the figure details the parameters and dimension of the subnetwork $h_{\phi}^{(t)}$. The parameters depend on the received signal input dimension N.

In order to learn a general, nonlinear mapping $f(\cdot) : \mathcal{Y}_{\text{QAM}} \to \mathcal{M}^{2K}$, the proposed dequantization approach works on iteratively enhancing the estimate of the unquantized signal $\mathbf{r}^{(t)}$, beginning with the

quantized measurements $\mathbf{y}$. These unquantized measurements are used to estimate the transmitted symbols $\mathbf{x}$. This entire operation is executed via a three-stage update process, described below.

1. ZF estimate: Multi-user M-QAM signal estimation

$$\mathbf{x}_{\text{ZF}}^{(t)} = \mathbf{H}^{\dagger}\mathbf{r}^{(t-1)} \tag{5.17a}$$

$$\mathbf{x}^{(t)} = \eta^{(t)}\mathbf{x}_{\text{ZF}}^{(t)} \tag{5.17b}$$

2. Dequantization: Learning quantization correction

$$\mathbf{e}_{\text{ZF}}^{(t)} = \mathbf{r}^{(t)} - \mathbf{H}\mathbf{x}^{(t)} \tag{5.18a}$$

$$\hat{\mathbf{e}}_{\text{ZF}}^{(t)} = \frac{\mathbf{e}_{\text{ZF}}^{(t)} - \mathbf{a}^{(t)}}{\mathbf{b}^{(t)}} \tag{5.18b}$$

$$\mathbf{e}^{(t)} = \alpha^{(t)}\hat{\mathbf{e}}_{\text{ZF}}^{(t)} + h_{\phi}^{(t)}(\hat{\mathbf{e}}_{\text{ZF}}^{(t)}, \mathbf{r}^{(t-1)}, \mathbf{y}) \tag{5.18c}$$

3. Updating unquantized signal

$$\mathbf{r}^{(t)} = \mathbf{r}^{(t-1)} - \mathbf{e}^{(t)} \tag{5.19}$$

The following points further elaborate on each stage of the proposed algorithm (5.17)-(5.19).

- ZF *estimate* - The step (5.17a) is similar to Algorithm 7. However, differently, the Gaussian projection $\mathcal{P}_{\mathcal{X}}(\cdot)$ from (5.15) is not used. Instead, an iteration-dependent learnt scalar $\eta^{(t)}$ is used to scale the M-QAM signal amplitudes, as seen in (5.17b).

- *Dequantization* - Instead of directly subtracting the quantization noise from the quantized measurements, as seen in (5.14), a general error correction term is learnt from the data. First, the ZF quantization error is evaluated, as given in (5.18a). In step (5.18b), the learnt parameters $a^{(t)}$ and $b^{(t)}$ normalize this error. In step (5.18c), this is utilized, along with previous unquantized estimate $\mathbf{r}^{(t-1)}$ and the quantized measurements $\mathbf{y}$ to create a DNN-aided regularized correction $h_{\phi}^{(t)}(\cdot)$ to the quantization error term $\mathbf{e}_{\text{ZF}}^{(t)}$. This is added to a scaled normalized error $\hat{\mathbf{e}}_{\text{ZF}}^{(t)}$, with learnt parameter $\alpha^{(t)}$, to create the final error correction term $\mathbf{e}^{(t)}$. The details of the DNN implementation $h_{\phi}^{(t)}(\cdot)$ are provided in Sec. 5.4.3.

- *Unquantized signal update* - The learnt quantization correction $\mathbf{e}^{(t)}$ is subtracted from the current unquantized estimate to get the new value $\mathbf{r}^{(t)}$.

5.4.3 Implementing the iterative dequantizer

This subsection introduces the DNN implementation of the iterative dequantizer: DQuantNet (De-Quantization Network). The block diagram for the DQuantNet is provided in Fig. 5.4. Explaining this further, the details of the DNN architecture and training are provided below.

DNN architecture

Based on the block diagram in Fig. 5.4, we have the following.

- The iterative dequantization algorithm is unfolded as a T-stage DNN. For each iteration t, a distinct subnetwork is utilized.
- At each iteration, the unquantized estimate $\mathbf{r}^{(t-1)}$ (starting with $\mathbf{r}^{(0)} = \mathbf{y}$) and quantized measurements $\mathbf{y}$ are fed in as input. The output of each iteration is the refined unquantized estimate $\mathbf{r}^{(t)}$.
- Each iteration t subnetwork block consists of three stages, explained in equations (5.17)-(5.19). The specific steps and flow of information in each of the stages is further elaborated in the red dashed line boxes.
- Within each iteration subnetwork, the learanble parameters are shown in the shaded blue boxes. For the dequantization stage, the regularization subnetwork $h_{\phi}^{(t)}$ consists of 1-D convolutional and fully connected layers. The specific details of the parameters within $h_{\phi}^{(t)}(\cdot)$ are detailed in the accompanying table in Fig. 5.4

Loss function and training

The loss function for DNN training $\mathcal{L}$ is designed as a weighted sum of individual sub-losses at each iteration $\mathcal{L}^{(t)}$. This is given by the form below.

$$\mathcal{L}^{(t)} = \frac{1}{\lambda + 1}\left[||\mathbf{r}^{(t)} - \mathbf{r}_{\text{train}}||^2 + \lambda||\mathbf{x}_{\text{ZF}}^{(t)} - \mathbf{x}_{\text{train}}||^2\right] \tag{5.20a}$$

$$\mathcal{L} = \sum_{t=1}^{T} \exp(-\beta(T-t))\,\mathcal{L}^{(t)}. \tag{5.20b}$$

Based on this formulation, we have the following

- Each sub-loss at iteration t, i.e. (5.20a), consists of a weighted sum of the dequantization error as well as the symbol recovery error. This weighted sum balances both these factors, important for robust symbol recovery. The value of λ is selected through hyperparameter tuning.

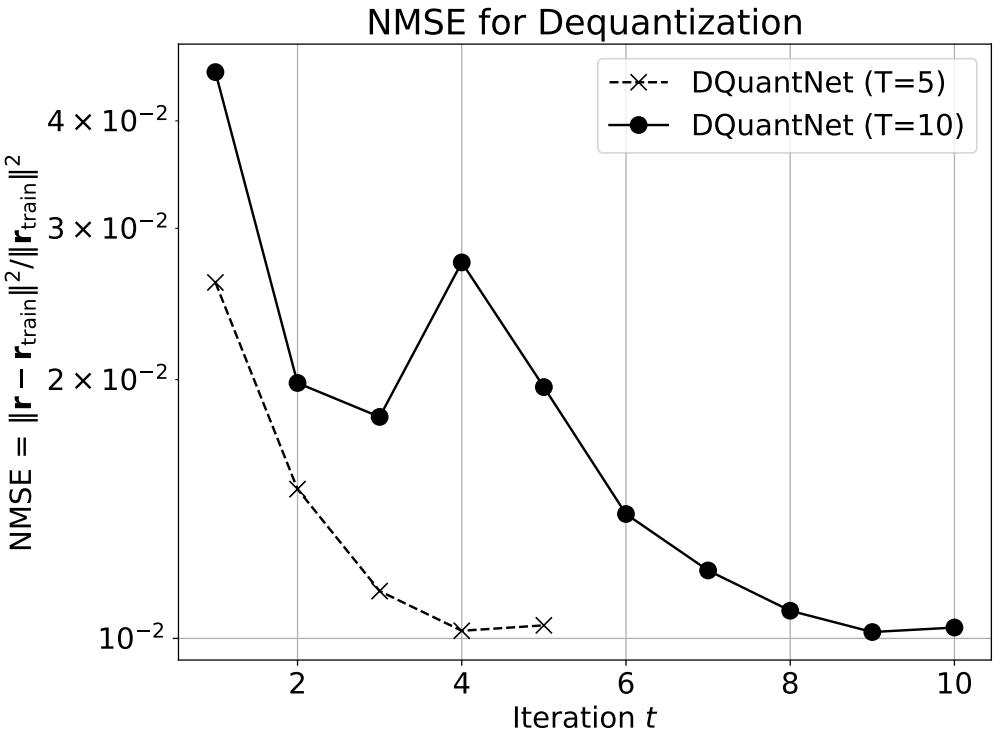


Figure 5.5: Dequantization NMSE evaluated over each iteration t. The $K = 4$ users transmit QPSK symbols for a $N = 32$ antenna BS. The SNR is 30 dB.

- Each sub-loss $\mathcal{L}^{(t)}$ is exponentially weighted based on the iteration. Thus the total loss is not excessively penalized by the estimation error in the initial stages. Additionally, this also creates a trajectory for the DNN-aided framework, reducing the dequantization loss $||\mathbf{r}^{(t)} - \mathbf{r}_{\text{train}}||^2$ over each iteration.

The DNN training follows from the same steps as the different standard works in this domain [1, 20, 36, 138]. The network training is carried out via minibatch gradient descent, with the chosen batch size $N_{\text{train}} = 32$. In order to train the DQuantNet on the set of randomly generated Rayleigh channel matrices, each minibatch is generated from a different channel matrix $\mathbf{H}$, denoted by $\mathcal{B}_{\mathbf{H}}$. Based on the described system model (5.1)-(5.4), the minibatch set is generated as $\mathcal{B}_{\mathbf{H}} = \{\bar{\mathbf{x}}_n, \bar{\mathbf{z}}_n, \bar{\mathbf{y}}_n\}_{n=1}^{N_{\text{train}}}$. The loss function (5.20) is used for training. We practically implement minibatch gradient descent with the Adam update [144] for each training minibatch to keep a check on the learning rate. For regularization of DNN weights, we utilize weight decay to further increase resilience by preventing exploding network weights.

5.4.4 Discussion

Based on the introduced iterative dequantization framework, i.e., the DQuantNet, the following points elaborate further on the process of iteratively learning the dequantization function.

Iterative dequantization vs projected ZF

The baseline projected ZF Algorithm 7 works on estimating the quantization noise iteratively. This estimate is refined over iterations and, at each stage, subtracted from the quantized measurements to get the current unquantized estimate. Differently, the DQuantNet takes a small step towards the final unquantized measurement, using each intermediate estimate $\mathbf{r}^{(t)}$ for ZF detection of the M-QAM signal $\mathbf{x}^{(t)}$. This approach is more resilient to SER correction, and balances the same with reduced M-QAM cluster spread. In addition, the update process for learning a stage-dependent quantization correction $\mathbf{e}^{(t)}$, i.e., (5.18), is a generalized projection to the quantization noise space, learnt from unquantized training data.

Significance of learning regularized error correction

The key learning step in the DNN-aided iterative dequantization is (5.18b) and (5.18c). As stated in Sec. 5.4.1, the general dequantization mapping is an intractible function. By learning a regularized error corection step, *(i)* The framework utilizes the information in the ZF estimate, i.e., $\mathbf{e}_{\text{ZF}}^{(t)}$, and *(ii)* It uses the unquantized training data $\mathbf{r}_{\text{train}}^{(t)}$ to learn the appropriate correction term to this ZF quantization error estimate.

Dequantization convergence

The convergence plots in Fig. 5.5 show the dequantization performance for QPSK transmitted symbols. As seen from the performance plots, increasing T beyond a certain value doesn't significantly improve performance. It is evident that the dequantization error is bounded below for a given choice of the quantizer. This is because *(i)* For a finite antenna case, the dequantization probability $\text{P}_{\text{dequant}}$ is not exactly 1, implying that the recovery performance is bounded below, and *(ii)* The DQuantNet is trained for the entire distribution of Rayleigh fading channels $\mathbf{H}$, not just optimized for a single channel matrix.

Few-bit MIMO vs one-bit MIMO

One-bit MIMO is the limiting case of the few-bit MIMO system. Applying the additive noise model (5.5) to one-bit quantizers clearly shows that the quantization noise $\mathbf{e} = \pm 1 - \mathbf{r}$, i.e., the unquantized signal (up to a scalar bias). Specifically, the magnitude of the quantized measurements do not provide any information about the input unquantized measurements, beyond the sign. To this end, the different state-of-the-art detectors for one-bit MIMO (see Sec. 5.1) can only match the sign($\mathbf{r}$) to the true value. Beyond this, the detector uses additional projection or scaling steps to estimate the M-QAM symbols. By just adding

an additional bit in the MIMO system, the signal there is additional information about the magnitudes of the unquantized measurements (see Sec. 5.3.1) and a probability for one-one mapping, i.e., $\mathrm{P}_{\mathrm{dequant}}$. The probability of learning this intractable (probably) one-one mapping makes DNN-aided detection much better suited for robust detector design, as compared to the one-bit MIMO system.

5.5 Experimental Results

5.5.1 Simulation setup

The different parameters used for the simulations as well as the parameters of the different algorithms are provided below.

Channel model and quantized measurements

The M-QAM symbols $\mathbf{x}$ are transmitted by $K = 4$ users to a BS antenna with $N = 32$. For this work, we consider the transmission of QPSK and 16-QAM symbols. The input SNR $= \frac{\mathbb{E}(||\mathbf{Hx}||^2)}{\mathbb{E}(||\mathbf{n}||^2)}$ is taken in the range $[-15, 20]$ dB and $[10, 45]$ dB for QPSK and 16-QAM symbols, respectively. For consistency in evaluation, the same b-bit quantizer is used for the MU-MIMO system, irrespective of the modulation order. To ensure this condition is satisfied, the trasnmitted symbols $\mathbf{x}$ are normalized such that $\mathbb{E}\,||\mathbf{x}[i]||^2 = 1$. In the tests within this section, one-bit and two-bit ADCs are used as few-bit quantizers. Beyond two-bits, the quantization noise is more accurately modeled using independent noise. Various state-of-the-art detectors already exist for these systems.

Performance benchmarks

The proposed DQuantNet is compared to the current state-of-the-art detector for few-bit MIMO systems, specifically the FBM-DetNet [20]. For the QPSK symbols, we utilize the ML detection with unquantized measurements as the theoretical upper bound for detection. We also compare the results with the introduced projected ZF detector, i.e., Algorithm 7, to get the dequantization-based detection baseline.

DQuantNet parameters

Based on the convergence plots for Fig. 5.5, the DQuantNet is run for $T = 5$ iterations for both QPSK and 16-QAM symbols. The training procedure is explained in Sec. 5.4.3. For the loss function (5.20), the hyperparameter values are taken to be $\lambda = 5$ and $\beta = 0.5$. The DQuantNet is trained for high

SNR=30 dB for QPSK symbols and 25 dB for 16-QAM symbols. This specializes the network for high-SNR dequantization, efficiently removing the quantization noise at high SNR. It also follows from the inference of similar works [28, 36, 138].

5.5.2 Detection performance for general channel

This section now compares the performance of the proposed DQuantNet to the state-of-the-art detectors for a general channel drawn from a distribution of Rayleigh fading channels.

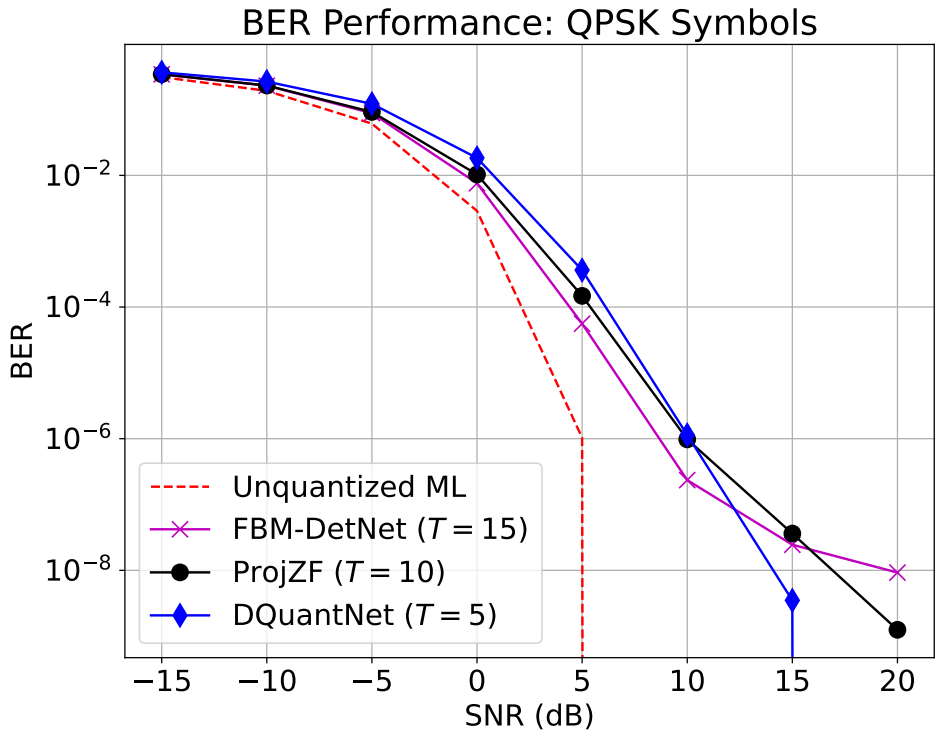


Figure 5.6: Performance comparison of DQuantNet for 2-bit MIMO. The $K = 4$ users transmit QPSK symbols for a $N = 32$ antenna BS.

The recovery performance for QPSK transmitted symbols is shown in Fig. 5.6. Based on these performance plots it is evident that the unquantized ML is comparable to two-bit detection for very low SNR but quickly deviates beyond 5 dB. This shows very little scope for further optimization of two-bit MIMO detection at very low SNR. The FBM-DetNet slightly outperforms our proposed dequantization detection for low to moderate SNRs. However, *(i)* This framework uses three times as many iterations as the DQuantNet, *(ii)* The DQuantNet is specifically trained for high-SNR dequantization, and *(iii)* The FBM-DetNet saturates to a higher BER for higher SNR. Differently, there is no saturation observed for the proposed DQuantNet at high SNR; showing a more efficient framework for symbol error correction through dequantization. The

similar orders of BER for QPSK shows this modulation to be inherently robust to quantization. This is not the case for higher-order M-QAM modulation.

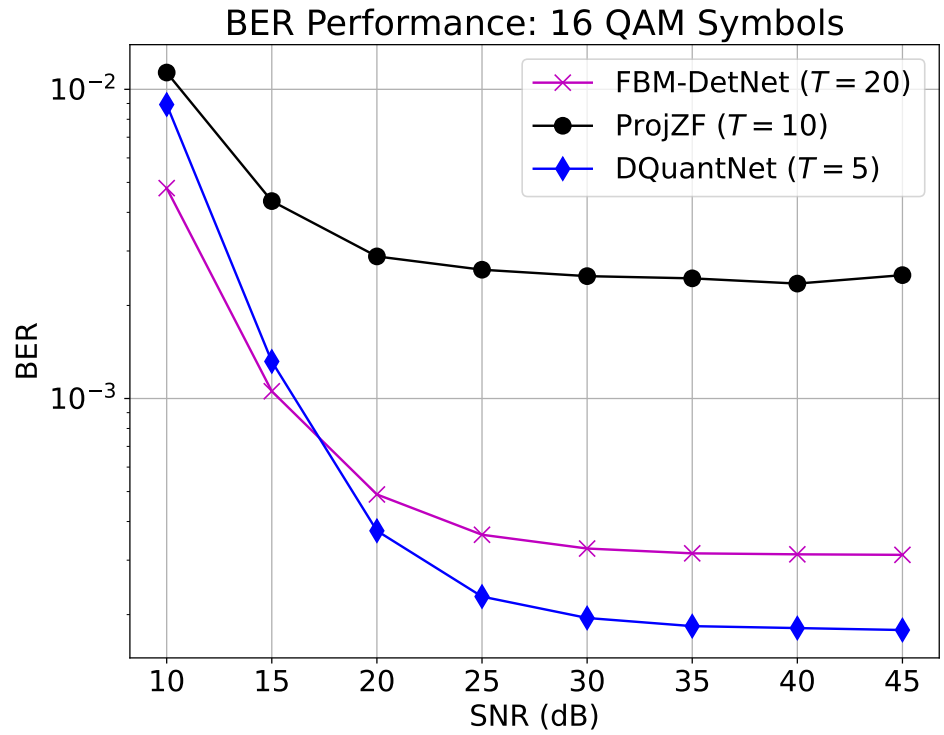


Figure 5.7: Performance comparison of DQuantNet for 2-bit MIMO. The $K = 4$ users transmit 16-QAM symbols for a $N = 32$ antenna BS.

The recovery performance for 16-QAM transmitted symbols is shown in Fig. 5.7. Based on these plots it is evident that the gap to the projected ZF increases with increasing modulation order. Further, the DQuantNet improves on the FBM-DetNet for higher SNR values, using significantly fewer iterations. This shows the proposed dequantization-based detection to be an efficient framework for recovery of higher order M-QAM symbols.

5.6 Conclusions

Through this work, we introduce a novel DNN-aided receiver based on iterative signal dequantization. The theoretical insights into signal dequantization provides an expression for the probability for perfect dequantization, that can be empirically evaluated. This also provides guidance on the design of the massive MIMO system, number of users and modulation order to get a desired performance in terms of probability of dequantizing the received signal. Finally, this work presents the DQuantNet, an unfolded DNN to learn

this dequantization mapping using an three stage iterative procedure. The iteration-weighted loss function enables efficient training by utilizing the intermediate estimates. Future work in this domain involves more detailed analysis into theoretical limits of signal dequantization. In addition, this work can also be extended beyond the rich-fading Rayleigh channel to the mmWave case with its own challenges for non-uniform user power distribution.

Chapter 5, in part, is a reprint of the material being prepared for submission, Aditya Sant, and Bhaskar D. Rao. "Few-bit MIMO Detection Using DNN-aided Dequantization". The dissertation author is the primary investigator and author of this paper.

Chapter 6

Contributions and Future Work

This chapter summarizes the contributions of this dissertation, building robust next generation MIMO communication systems through the following two verticals: *(i)* Block-sparse signal modeling for mmWave channels, and *(ii)* Building robust receiver algorithms for few-bit MIMO systems.

Block-sparse signal recovery

Motivated by the need to robustly model the heterogeneous angular scattering in mmWave channels, this work introduced a robust signal recovery framework for a general block-sparse signal, with unknown block sizes and locations. The underlying framework for block-sparse signal recovery worked on modifying the sparse Bayesian learning algorithm, a type-II estimation algorithm utilizing a parametric Gaussian prior for the underlying sparse signal, and consequently estimating the hyperparameters of this prior distribution. The main contribution of this work was the introduction of a novel class of total-variation based regularizers. Unlike conventional regularizers operating in the signal space, the use of the block-sparse regularizers in the hyperparameter space, as laid out by the SBL framework, proved to be more resilient for block-sparse signal recovery, especially for multiple measurement vectors. This work introduced three specific block-sparse regularizers in the SBL hyperparameter space, *(i)* Linear TV, *(ii)* Log TV, and *(iii)* Difference of Logs (DoL) TV. A convex optimization framework, based on majorizing the underlying SBL cost function, was presented for the first two classes, and an expectation maximization (EM) algorithm was presented for the third. The proof of convergence was provided for the underlying block-sparse signal recovery using expectation maximization. The experimental results presented the strength of the TV-SBL algorithm for a huge range of block-sparse signals. Unlike conventional block-sparse signal recovery algorithms, the TV-SBL

did not compromise on the sparse signal recovery, at the cost of block-sparse signal recovery. This makes this framework especially resilient for signal recovery of hybrid block-sparse signals, containing both isolated sparse and block-sparse signal components, that could potentially be observed for mmWave channels.

This work also provided a basis for extending block-sparse signal recovery for both 1-D and 2-D signals. The synergistic use of DNN-aided methods, building on the SBL framework, can overcome the challenges for tractably modeling the signal prior. Future directions and developments for both model-based and DNN-aided methods are promising. For model-based frameworks, the exploration of a broader range of block-sparse enforcing signal priors promises to advance the theoretical underpinnings of block-sparse signal recovery. Concurrently, DNN-aided methodologies are poised to mitigate issues of model mismatch and computational complexity as this framework is extended to more complex systems.

Insights into receiver detection algorithms for few-bit MIMO system

Motivated by the need to build robust detection algorithms for few-bit MIMO systems, this work provided theoretical insights as well as novel DNN-aided recovery algorithms for few-bit MIMO systems. The one-bit MIMO, a specific case of the few-bit MIMO systems, was studied in more detail here. The theoretical analysis of the receiver algorithms began by the characterization of the properties of the maximum likelihood optimization. The insights into in the convexity and smoothness of the likelihood function enabled characterization of the convergence of the GD algorithm for signal recovery. The accelerated GD was presented as an improved receiver detection algorithm, capitalizing on the smoothness properties of the likelihood. The analysis was extended to surrogate likelihood measures for signal recovery, specifically the logistic regression based likelihood, a framework used in recent research works with much success. Through the analysis of the underlying Hessian, the strengths of this surrogate was elucidated; an analysis that hasn't been provided in recent research works utilizing this function for one-bit MIMO signal recovery. The role of projection was elaborated as well, considering that signal recovery for communication systems is constrained to finite M-QAM constellations. This work proposed the A-PrOBNet, a DNN-aided projected accelerated GD algorithm, that matched the state-of-the-art recovery algorithms for one-bit MIMO for Rayleigh fading channels.

DNN-aided methods were analyzed in detail, through the proposed regularized GD detection. This DNN-aided regularization, applied to the GD, enabled increased flexibility for signal recovery, which was exploited for signal recovery in mmWave channels. Different from the Rayleigh fading channel, the mmWave channel presented more challenges to joint user detection owing to greater channel power variation among the users. The regularized GD was exploited to develop a novel hierarchical detection training strategy for

this channel, that resulted in equitable performance among the different users.

Finally, the DNN-aided framework was also utilized for signal recovery for few-bit MIMO systems beyond one-bit, specifically detection for two-bit MIMO signals. Different from one-bit MIMO, the design of the quantizer plays a key role in detection performance by balancing the trade-off between signal resolution and clipping. Further, the transmission of symbols from a finite M-QAM constellation enables the evaluation of a probability of perfect dequantization. The framework for dequantization, however, requires analysis of the quantization noise statistics, that are not tractably analyzed. This makes DNN-aided frameworks ideally suited for the signal recovery. The proposed framework, DQuantNet, was used to learn a general projection strategy for the intractable quantization noise for DNN-aided dequantization, that resulted in improved system performance over the existing state-of-the-art.

Future directions in few-bit MIMO detection algorithm design, for one-bit and beyond are very promising. The introduced A-PrOBNet, using Gaussian denoising can be extended to handle the challenges of the mmWave channel. Similarly, improved DNN architectures and training strategies can enhance performance for the generalized projection strategy. Incorporating SNR into detection frameworks, where the balance between quantization noise and AWGN is effectively handled, is also a promising direction for both one-bit as well as two-bit MIMO detection. The underlying synergy between model-based and DNN-aided methods will be key to continual development of robust detection algorithms.

Bibliography

[1] L. V. Nguyen, A. L. Swindlehurst, and D. H. Nguyen, "Linear and deep neural network-based receivers for massive mimo systems with one-bit adcs," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7333–7345, 2021.

[2] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas *et al.*, "On the road to 6g: Visions, requirements, key technologies and testbeds," *IEEE Communications Surveys & Tutorials*, 2023.

[3] A.-S. Bana, E. De Carvalho, B. Soret, T. Abrao, J. C. Marinello, E. G. Larsson, and P. Popovski, "Massive mimo for internet of things (iot) connectivity," *Physical Communication*, vol. 37, p. 100859, 2019.

[4] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (iot) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5g-iot scenarios," *Ieee Access*, vol. 8, pp. 23 022–23 040, 2020.

[5] T. Wild, V. Braun, and H. Viswanathan, "Joint design of communication and sensing for beyond 5g and 6g systems," *IEEE Access*, vol. 9, pp. 30 845–30 857, 2021.

[6] A. Dogra, R. K. Jha, and S. Jain, "A survey on beyond 5g network with the advent of 6g: Architecture and emerging technologies," *IEEE access*, vol. 9, pp. 67 512–67 547, 2020.

[7] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE communications magazine*, vol. 49, no. 6, pp. 101–107, 2011.

[8] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5g cellular: It will work!" *IEEE access*, vol. 1, pp. 335–349, 2013.

[9] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE journal on selected areas in communications*, vol. 32, no. 6, pp. 1164–1179, 2014.

[10] N. Docomo *et al.*, "5g channel model for bands up to100 ghz," Technical report, Tech. Rep., 2016.

[11] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2002.

[12] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[13] E. Björnson, J. Hoydis, L. Sanguinetti *et al.*, "Massive mimo networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.

[14] S. Hur, S. Baek, B. Kim, Y. Chang, A. F. Molisch, T. S. Rappaport, K. Haneda, and J. Park, "Proposal on millimeter-wave channel modeling for 5G cellular system," vol. 10, no. 3, pp. 454–469, April 2016.

[15] C. Gustafson, K. Haneda, S. Wyne, and F. Tufvesson, "On mm-wave multipath clustering and channel modeling," vol. 62, no. 3, pp. 1445–1455, March 2014.

[16] K. Haneda, "Channel models and beamforming at millimeter-wave frequency bands," *IEICE Transactions*, vol. 98-B, pp. 755–772, 2015.

[17] Z. Zhang, J. Ryu, S. Subramanian, and A. Sampath, "Coverage and channel characteristics of millimeter wave band using ray tracing," in *2015 IEEE international conference on communications (ICC)*. IEEE, 2015, pp. 1380–1385.

[18] M. Bensebti, J. McGeehan, and M. Beach, "Indoor multipath radio propagation measurements and characterisation at 60 ghz," in *1991 21st European Microwave Conference*, vol. 2. IEEE, 1991, pp. 1217–1222.

[19] T. S. Rappaport, E. Ben-Dor, J. N. Murdock, and Y. Qiao, "38 ghz and 60 ghz angle-dependent propagation for cellular & peer-to-peer wireless communications," in *2012 IEEE international conference on communications (ICC)*. IEEE, 2012, pp. 4568–4573.

[20] L. V. Nguyen, D. H. Nguyen, and A. L. Swindlehurst, "Deep learning for estimation and pilot signal design in few-bit massive mimo systems," *IEEE Transactions on Wireless Communications*, 2022.

[21] D. H. Nguyen, "Neural network-optimized channel estimator and training signal design for mimo systems with few-bit adcs," *IEEE Signal Processing Letters*, vol. 27, pp. 1370–1374, 2020.

[22] J. Mo, P. Schniter, and R. W. Heath, "Channel estimation in broadband millimeter wave mimo systems with few-bit adcs," *IEEE Transactions on Signal Processing*, vol. 66, no. 5, pp. 1141–1154, 2017.

[23] J. Mo, A. Alkhateeb, S. Abu-Surra, and R. W. Heath, "Hybrid architectures with few-bit adc receivers: Achievable rates and energy-rate tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2274–2287, 2017.

[24] L. Fan, S. Jin, C.-K. Wen, and H. Zhang, "Uplink achievable rate for massive mimo systems with low-resolution adc," *IEEE Communications Letters*, vol. 19, no. 12, pp. 2186–2189, 2015.

[25] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive mimo uplink with low-resolution adcs," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 4038–4051, 2017.

[26] C. Stöckle, J. Munir, A. Mezghani, and J. A. Nossek, "Channel estimation in massive mimo systems using 1-bit quantization," in *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2016, pp. 1–6.

[27] C.-L. Liu and P. Vaidyanathan, "One-bit sparse array doa estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3126–3130.

[28] A. Sant and B. D. Rao, "Doa estimation in systems with nonlinearities for mmwave communications," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4537–4541.

[29] A. Mezghani, M.-S. Khoufi, and J. A. Nossek, "A modified mmse receiver for quantized mimo systems," *Proc. ITG/IEEE WSA, Vienna, Austria*, pp. 1–5, 2007.

[30] D. K. 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.

[31] Q. Wan, J. Fang, H. Duan, Z. Chen, and H. Li, "Generalized bussgang lmmse channel estimation for one-bit massive mimo systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4234–4246, 2020.

[32] 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.

[33] S. S. Thoota and C. R. Murthy, "Variational bayes' joint channel estimation and soft symbol decoding for uplink massive mimo systems with low resolution adcs," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3467–3481, 2021.

[34] J. Choi, D. J. Love, D. R. Brown, and M. Boutin, "Quantized distributed reception for mimo wireless systems using spatial multiplexing," *IEEE Transactions on Signal Processing*, vol. 63, no. 13, pp. 3537–3548, 2015.

[35] 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.

[36] A. Sant and B. D. Rao, "Regularized neural detection for one-bit massive mimo communication systems," *arXiv e-prints*, pp. arXiv–2305, 2023.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[39] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[40] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[41] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE international conference on robotics and automation*. IEEE, 2010, pp. 4372–4378.

[42] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[43] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[45] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[46] K. Lønning, P. Putzky, J.-J. Sonke, L. Reneman, M. W. Caan, and M. Welling, "Recurrent inference machines for reconstructing heterogeneous mri data," *Medical image analysis*, vol. 53, pp. 64–78, 2019.

[47] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *jama*, vol. 316, no. 22, pp. 2402–2410, 2016.

[48] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[49] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, "Sparks of artificial general intelligence: Early experiments with gpt-4. arxiv," *arXiv preprint arXiv:2303.12712*, 2023.

[50] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[51] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th international conference on international conference on machine learning*, 2010, pp. 399–406.

[52] F. Sohrabi, Z. Chen, and W. Yu, "Deep active learning approach to adaptive beamforming for mmwave initial alignment," *IEEE Journal on Selected Areas in Communications*, 2021.

[53] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Reinforcement learning of beam codebooks in millimeter wave and terahertz mimo systems," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 904–919, 2021.

[54] A. Sant, A. Abdi, and J. Soriaga, "Deep sequential beamformer learning for multipath channels in mmwave communication systems," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5198–5202.

[55] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based mimo communications," *arXiv preprint arXiv:1707.07980*, 2017.

[56] T. Diskin, N. Samuel, and A. Wiesel, "Deep mimo detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–5.

[57] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for mimo detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1702–1715, 2020.

[58] 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.

[59] K. Pratik, B. D. Rao, and M. Welling, "Re-mimo: Recurrent and permutation equivariant neural mimo detection," *IEEE Transactions on Signal Processing*, vol. 69, pp. 459–473, 2020.

[60] Ö. T. Demir and E. Björnson, "Channel estimation in massive mimo under hardware non-linearities: Bayesian methods versus deep learning," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 109–124, 2019.

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

[62] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and iot networks: Potentials, current solutions, and open challenges," *IEEE communications surveys & tutorials*, vol. 22, no. 2, pp. 1251–1275, 2020.

[63] X. Chen, C. Wu, T. Chen, H. Zhang, Z. Liu, Y. Zhang, and M. Bennis, "Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective," *IEEE Transactions on wireless communications*, vol. 19, no. 4, pp. 2268–2281, 2020.

[64] A. Paulraj, R. Nabar, and D. Gore, *Introduction to space-time wireless communications*. Cambridge university press, 2003.

[65] E. J. Candés, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," vol. 52, no. 2, pp. 489–509, Feb. 2006.

[66] D. L. Donoho, "Compressed sensing," vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[67] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[68] E. Van Den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2009.

[69] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," vol. 56, no. 4, pp. 1982–2001, 2010.

[70] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," vol. 58, no. 6, pp. 3042–3054, 2010.

[71] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," vol. 57, no. 8, pp. 3075–3085, 2009.

[72] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity." *Journal of Machine Learning Research*, vol. 12, no. 11, 2011.

[73] T. Peleg, Y. C. Eldar, and M. Elad, "Exploiting statistical dependencies in sparse representations for signal recovery," vol. 60, no. 5, pp. 2286–2303, May 2012.

[74] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, p. 211–244, Sep. 2001. [Online]. Available: https://doi.org/10.1162/15324430152748236

[75] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[76] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning," vol. 5, no. 5, pp. 912–926, Sep. 2011.

[77] ——, "Extension of SBL algorithms for the recovery of block sparse signals with intra-block correlation," vol. 61, no. 8, pp. 2009–2015, April 2013.

[78] L. Yu, H. Sun, J.-P. Barbot, and G. Zheng, "Bayesian compressive sensing for cluster structured sparse signals," *Signal processing*, vol. 92, no. 1, pp. 259–269, 2012.

[79] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," vol. 63, no. 2, pp. 360–372, Jan 2015.

[80] J. Dai, A. Liu, and H. C. So, "Non-uniform burst-sparsity learning for massive MIMO channel estimation," vol. 67, no. 4, pp. 1075–1087, Feb 2019.

[81] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," vol. 25, no. 6, pp. 131–146, November 2008.

[82] L. Wang, L. Zhao, S. Rahardja, and G. Bi, "Alternative to extended block sparse Bayesian learning and its relation to pattern-coupled sparse Bayesian learning," vol. 66, no. 10, pp. 2759–2771, May 2018.

[83] M. Shekaramiz, T. K. Moon, and J. H. Gunther, "Bayesian compressive sensing of sparse signals with unknown clustering patterns," *Entropy*, vol. 21, no. 3, 2019.

[84] L. Wang, L. Zhao, L. Yu, J. Wang, and G. Bi, "Structured Bayesian learning for recovery of clustered sparse signal," *Signal Processing*, vol. 166, p. 107255, 2020.

[85] A. Sant, M. Leinonen, and B. Rao, "General total variation regularized sparse Bayesian learning for robust block-sparse signal recovery," Toronto, Canada, Jun. 6–11 2021, pp. 5604–5608.

[86] R. M. Neal, *Bayesian Learning for Neural Networks*. Berlin, Heidelberg: Springer-Verlag, 1996.

[87] D. J. MacKay and C. Laboratory, "Bayesian non-linear modelling for the prediction competition," in *In ASHRAE Transactions, V.100, Pt.2*. ASHRAE, 1994, pp. 1053–1062.

[88] D. P. Wipf and B. D. Rao, "An empirical Bayesian strategy for solving the simultaneous sparse approximation problem," vol. 55, no. 7, pp. 3704–3716, 2007.

[89] P. Pal and P. P. Vaidyanathan, "Pushing the limits of sparse support recovery using correlation information," vol. 63, no. 3, pp. 711–726, 2015.

[90] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems*, 2002, pp. 383–389.

[91] R. R. Pote and B. D. Rao, "Robustness of sparse Bayesian learning in correlated environments," Barcelona, Spain, May 4-8, 2020, pp. 9100–9104.

[92] M. Al-Shoukairi, P. Schniter, and B. D. Rao, "A GAMP-based low complexity sparse Bayesian learning algorithm," vol. 66, no. 2, pp. 294–308, 2018.

[93] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," vol. 4, no. 2, pp. 317–329, 2010.

[94] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational em algorithms for non-gaussian latent variable models," *Advances in neural information processing systems*, vol. 18, p. 1059, 2006.

[95] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using Laplace priors," vol. 19, no. 1, pp. 53–63, 2010.

[96] D. Wipf, J. Palmer, B. Rao, and K. Kreutz-Delgado, "Performance evaluation of latent variable models with sparse priors," vol. 2, 2007, pp. II–453–II–456.

[97] T. A. Srikrishnan and B. D. Rao, "Addressing the noise variance problem in sparse Bayesian learning," 2018, pp. 1974–1979.

[98] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005. [Online]. Available: https://doi.org/10.1137/040605412

[99] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259 – 268, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016727899290242F

[100] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 813–824, 1998.

[101] J. Liu, T.-Z. Huang, I. W. Selesnick, X.-G. Lv, and P.-Y. Chen, "Image restoration using total variation with overlapping group sparsity," *Information Sciences*, vol. 295, pp. 232–246, 2015.

[102] J. Huang and F. Yang, "Compressed magnetic resonance imaging based on wavelet sparsity and nonlocal total variation," in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2012, pp. 968–971.

[103] X. Zhang, T. Bai, H. Meng, and J. Chen, "Compressive sensing-based ISAR imaging via the combination of the sparsity and nonlocal total variation," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 5, pp. 990–994, 2014.

[104] M. Eickenberg, E. Dohmatob, B. Thirion, and G. Varoquaux, "Grouping total variation and sparsity: Statistical learning with segmenting penalties," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 685–693.

[105] H. K. Aggarwal and A. Majumdar, "Hyperspectral unmixing in the presence of mixed noise using joint-sparsity and total variation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp. 4257–4266, 2016.

[106] W. He, H. Zhang, and L. Zhang, "Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing," vol. 55, no. 7, pp. 3909–3921, 2017.

[107] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 6, pp. 797–829, 2006.

[108] E. J. Candés, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008.

[109] Y. Shen, J. Fang, and H. Li, "Exact reconstruction analysis of log-sum minimization for compressed sensing," vol. 20, no. 12, pp. 1223–1226, 2013.

[110] D. P. Wipf and S. S. Nagarajan, "A new view of automatic relevance determination," in *Advances in Neural Information Processing Systems*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 1625–1632.

[111] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[112] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[113] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[114] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[115] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, no. 1, pp. 3–34, 2015.

[116] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341–362, 2012.

[117] J. Bezdek, R. Hathaway, R. Howard, C. Wilson, and M. Windham, "Local convergence analysis of a grouped variable version of coordinate descent," *Journal of Optimization Theory and Applications*, vol. 54, no. 3, pp. 471–477, 1987.

[118] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[119] N. Chinchor, "MUC-4 evaluation metrics," in *In Proceedings of the Fourth Message Understanding Conference*, 1992, pp. 22–29.

[120] B. Murmann *et al.*, "Adc performance survey 1997-2020," in *IEEE Int. Solid-State Circuits Conf.(ISSCC) Dig. Tech. Papers VLSI Symp*, 2020.

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

[122] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[123] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2017.

[124] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[125] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[126] J. J. Bussgang, "Crosscorrelation functions of amplitude-distorted gaussian signals," 1952.

[127] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "One-bit massive mimo: Channel estimation and high-order modulations," in *2015 IEEE International Conference on Communication Workshop (ICCW)*. IEEE, 2015, pp. 1304–1309.

[128] C. Mollen, J. Choi, E. G. Larsson, and R. W. Heath, "Uplink performance of wideband massive mimo with one-bit adcs," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 87–100, 2016.

[129] 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*, vol. 65, no. 15, pp. 4075–4089, 2017.

[130] 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.

[131] Y.-S. Jeon, N. Lee, and H. V. Poor, "Robust data detection for mimo systems with one-bit adcs: A reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1663–1676, 2019.

[132] D. Ho, "Channel estimation and data detection methods for 1-bit massive mimo systems," Ph.D. dissertation, University of California San Diego, 2022.

[133] S. Khobahi, N. Shlezinger, M. Soltanalian, and Y. C. Eldar, "Lord-net: Unfolded deep detection network with low-resolution receivers," *IEEE Transactions on Signal Processing*, vol. 69, pp. 5651–5664, 2021.

[134] E. Balevi and J. G. Andrews, "One-bit ofdm receivers via deep learning," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4326–4336, 2019.

[135] S. Kim, M. So, N. Lee, and S. Hong, "Semi-supervised learning detector for mu-mimo systems with one-bit adcs," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[136] S. Kim, J. Chae, and S.-N. Hong, "Machine learning detectors for mu-mimo systems with one-bit adcs," *IEEE Access*, vol. 8, pp. 86 608–86 616, 2020.

[137] Y.-S. Jeon, D. Kim, S.-N. Hong, N. Lee, and R. W. Heath, "Artificial intelligence for physical-layer design of mimo communications with one-bit adcs," *IEEE Communications Magazine*, 2022.

[138] A. Sant and B. D. Rao, "Regularized neural detection for millimeter wave massive mimo communication systems with one-bit adcs," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[139] P. Barsocchi, "Channel models for terrestrial wireless communications: a survey," *CNR-ISTI technical report*, vol. 83, 2006.

[140] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media, 2003, vol. 87.

[141] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate o\bigl(k^2\bigr)," in *Doklady Akademii Nauk*, vol. 269, no. 3. Russian Academy of Sciences, 1983, pp. 543–547.

[142] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[143] 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.

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

[145] Z. Li, W. Xu, X. Zhang, and J. Lin, "A survey on one-bit compressed sensing: Theory and applications," *Frontiers of Computer Science*, vol. 12, no. 2, pp. 217–230, 2018.

[146] E. Knill and R. Laflamme, "Power of one bit of quantum information," *Physical Review Letters*, vol. 81, no. 25, p. 5672, 1998.

[147] P. T. Boufounos and R. G. Baraniuk, "1-bit compressive sensing," in *2008 42nd Annual Conference on Information Sciences and Systems*. IEEE, 2008, pp. 16–21.

[148] S. Khobahi, N. Naimipour, M. Soltanalian, and Y. C. Eldar, "Deep signal recovery with one-bit quantization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2987–2991.

[149] L. V. Nguyen, D. H. Nguyen, and A. L. Swindlehurst, "Dnn-based detectors for massive mimo systems with low-resolution adcs," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[150] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Deep learning for massive mimo with 1-bit adcs: When more antennas need fewer pilots," *IEEE Wireless Communications Letters*, vol. 9, no. 8, pp. 1273–1277, 2020.

[151] P. Putzky and M. Welling, "Recurrent inference machines for solving inverse problems," *arXiv preprint arXiv:1706.04008*, 2017.

[152] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *2019 IEEE International Workshop on Signal Processing Systems (SiPS)*. IEEE, 2019, pp. 266–271.

[153] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[154] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[155] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[156] A. Sant and B. D. Rao, "Insights into maximum likelihood detection for one-bit massive mimo communications," *IEEE Transactions on Wireless Communications (submitted)*.

[157] Y. Ding, S.-E. Chiu, and B. D. Rao, "Bayesian channel estimation algorithms for massive mimo systems with hybrid analog-digital processing and low-resolution adcs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 3, pp. 499–513, 2018.

[158] J. Zicheng, G. Shen, L. Nan, P. Zhiwen, and Y. Xiaohu, "Deep learning-based channel estimation for massive-mimo with mixed-resolution adcs and low-resolution information utilization," *IEEE Access*, vol. 9, pp. 54 938–54 950, 2021.

[159] O. Dabeer and U. Madhow, "Channel estimation with low-precision analog-to-digital conversion," in *2010 IEEE International Conference on Communications*. IEEE, 2010, pp. 1–6.

[160] 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.

[161] 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.

[162] 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.

[163] L. V. Nguyen, A. L. Swindlehurst, and D. H. Nguyen, "Variational bayes for joint channel estimation and data detection in few-bit massive mimo systems," *arXiv preprint arXiv:2212.01717*, 2022.