Navigating the Future — Machine Learning for Wireless Sensing and Localization

Tian, Guoda

2024

Document Version:
Publisher's PDF, also known as Version of record

Navigating the Future — Machine Learning for Wireless Sensing and Localization

Doctoral Thesis

Guoda Tian

LUND
UNIVERSITY

Department of Electrical and
Information Technology

Lund, Nov 2024

Guoda Tian
Department of Electrical and Information Technology
Lund University
Ole Römers Väg 3, 223 63 Lund, Sweden

Series of Licentiate and Doctoral Theses
ISSN 1654-790X, No. 179
ISBN 978-91-8104-119-4 (printed)
ISBN 978-91-8104-120-0 (digital)

This thesis is typeset using LaTeX 2ε with the body text in Palatino and Goudy Initials, headings in Helvetica, text in figures in Arial.

Printed by Tryckeriet i E-huset, Lund University, Lund, Sweden.

Till min familj samt alla mina kära
kompisar och kollegor

Abstract

CELLULAR-based localization and sensing pave the way for a variety of applications across various domains, ranging from autonomous driving to emergency care and intelligent traffic management. Although traditional methods have been effective, they still face challenges such as the requirement for highly accurate models and the inherent complexity of the algorithms. This thesis explores the potential of integrating machine learning (ML) techniques to augment the performance of sensing and localization systems. Three main topics are covered by this thesis, namely, ML-aided channel estimation, sensing, and localization.

The first topic focuses on the calibration of the RF chain and the estimation of the propagation channel, which serves as essential prerequisites for numerous subsequent applications including arriving angle estimation, radio localization, digital beamforming, and sensing. We introduce a novel RF chain calibration algorithm for massive multiple-input multiple-output (MIMO) systems, using uplink signals. We derive the maximum likelihood estimator (MLE) and its corresponding Cramér-Rao Lower Bound (CRLB). Additionally, we propose a novel ML-powered channel estimation pipeline for orthogonal frequency division multiplex (OFDM) systems, which efficiently estimates channel coefficients for all OFDM grids based on only using a limited number of pilot signals.

The second topic addresses ML-based wireless sensing using massive MIMO systems. We introduce a novel pipeline that first employs tensor-decomposition algorithms to extract channel characteristics. Subsequently, a fully connected neural network is deployed to classify human movements.

Our pipeline is evaluated by a measurement campaign using a massive MIMO testbed in an indoor environment, providing empirical evidence of the system's efficacy in wireless sensing applications. Our work shows the sensing capacity of massive MIMO in such scenarios.

The third topic focuses on cellular localization, comprising three main sections. The first section introduces classical radio localization algorithms, including time of arrival (ToA), angle of arrival (AoA) and time difference of arrival (TDoA). The second main section addresses ML-based localization with a massive MIMO system. We introduce a novel pipeline composed of several parallel processing chains. Each chain is trained on distinct channel fingerprints, including the channel impulse response (CIR) and covariance matrices. To improve localization accuracy, we model the position error in each processing chain as a Gaussian distribution and combine the outputs to compute localization uncertainty. Furthermore, we investigate the required training density using the Nyquist theorem. Our pipeline is evaluated through indoor measurements, showing a centimeter-level localization accuracy. The third section investigates ML-aided cellular localization using a 5G new radio (NR) system that operates in beam space. We present an ML-based localization pipeline that integrates attention mechanisms and advanced uncertainty estimation algorithms. Unlike in the previous section, this uncertainty estimation approach is not restricted to a Gaussian assumption. Validation of our pipeline is conducted through an extensive outdoor measurement campaign encompassing both Line-of-Sight (LoS) and Non Line-of-Sight (NLoS) propagation scenarios. Measurement results indicate the sub-meter accuracy levels with our pipeline.

Popular Science Summary

The author believes that every reader has heard of the magic of ChatGPT. It is incredible to witness how machine learning (ML) has transformed our world, from recognizing speech and classifying images to translating languages, driving cars autonomously, and creating AI-based art. These ML-powered applications help us learn, share knowledge, and interact with people with different backgrounds. They also free us from repetitive tasks and provide insights for creative activities, like composing music. Inspired by the success of ML in these areas, this thesis explores how ML can enhance cellular communication and localization systems.

The fundamentals of communication systems are based on signal processing and information theory. Although traditional cellular technology has become quite advanced, it still has limitations and room for improvement. For example, many traditional communication techniques need precise mathematical models, which is challenging to develop, especially in complex environments with limited computational power. However, ML systems can learn directly from data, making them highly adaptable and capable of handling complex environments without requiring detailed mathematical models. This thesis focuses on three key areas: the use of ML for channel estimation, sensing, and localization. By applying ML, our aim is to improve the efficiency and accuracy of these processes, making cellular communication and localization more robust and reliable.

For most cellular systems, it is essential to understand how the channel between mobile phones and base stations influences wireless signals. To do this, a special signal is sent, known as a pilot signal. Both the sender and receiver know this signal, which helps them understand the channel.

However, using too many pilot signals can waste energy and slow down data transmission. So, can we use ML to better understand the wireless environment between our phones and the base station? To find a solution to this problem, we create a robust system that works well even when the signal is weak. We test our system by simulating various wireless environments, including dynamic environments caused by moving objects. Our system performs better than traditional methods, even in difficult conditions.

We also explore how ML can help detect and recognize activities using existing 4G cellular signals, similar to how radar works. This capability is important for future wireless systems, as combined sensing and communication is a key emerging technology. Radio signals have unique properties compared to other signals like sound or light, allowing radio devices to work well in quiet or dark environments. Our research focuses on using ML to identify different events or activities through these wireless systems. For example, if the received signal is much weaker, it may indicate that the signal is blocked by an object. Specifically, the base station uses ML to analyze the signals patterns sent by mobile phones, helping it understand the surrounding environment. We tested our methods in an indoor environment and our system successfully distinguishes between different types of human movement.

The third topic of this thesis focuses on ML-assisted cellular localization technologies, which are crucial in our daily lives. Most readers are familiar with global positioning systems (GPS) and might wonder what role cellular networks play in localization. In fact, cellular systems complement GPS, especially when GPS signals are weak or unavailable, such as in the indoor environment. We mainly investigate *uplink* localization, where the base station receives signals transmitted by user equipment, such as mobile phones. In essence, using ML for localization means learning how the features of the received signal relate to the positions of the user equipment. To achieve this, the thesis explores advanced ML techniques such as ensemble learning, attention mechanisms, and uncertainty estimation. Ensemble learning involves training multiple models and combining their outputs for better performance. The attention mechanism, widely used in language processing systems like ChatGPT, helps the model focus on important parts of the input data. Uncertainty estimation is crucial for applications where safety is critical, such as autonomous driving. It allows the model not only to make predictions but also to indicate its confidence level, reducing the risk of random, potentially dangerous guesses.

In general, this thesis demonstrates that significant performance improvements in wireless systems can be achieved with the aid of ML approaches.

Acknowledgment

Time flies, and as I write this page, I suddenly realize that my long PhD journey is nearing its end. I am deeply grateful to express my appreciation to all the wonderful people who have supported me throughout this process. Without your encouragement and guidance, achieving this challenging goal of earning a PhD would have been impossible.

First and Foremost, I express my sincere gratitude to my main supervisor Prof. Fredrik Tufvesson for providing me the opportunity to pursue PhD study and guiding me throughout the entire journey. Your unwavering support, insightful advice, and patience have been invaluable in shaping my path as an individual researcher. I truly appreciate all the stimulating discussions we have had during our individual meetings, where your feedback not only enhance the quality of my manuscripts but also help me to critically assess my own work. Your deep knowledge and clear vision have been pivotal in teaching me the correct attitude toward research and science.

My utmost gratitude goes to my co-supervisor assistant Prof. Xuesong Cai. Without you, my PhD journey would be much more miserable. I admire your deep knowledge and appreciate so much your patience and hard work to support me.

I express my gratitude to Prof. Bo Bernhardsson, Prof. Liang Liu, and Prof. Ove Edfors, Dr. Basuki Priyanto, Dr. Sha Hu, for their invaluable supports. It has been a great pleasure collaborating with all of you, and I have gained so much from the experiences. I appreciate the help of Leif Wilhelmsson and Juan Vidal Alegría for providing me valuable feedbacks of my thesis. I would like to thank Michail Triantafyllidis for providing the photo used on the cover of this thesis.

I deeply appreciate all my colleges and friends who accompany my Phd journey. My sincere thanks to Ilayda Yaman and Dino Pjanić. I'm so grateful

that we've been able to collaborate on extremely challenging projects and measurements together. Thank you to my two office mates, Michiel Sandra and Muhammad Umar Farooq. It is always a pleasure to talk and hang out with both of you every day. I truly appreciate all the help you've given me, whether it's fixing things related to research or everyday issues I couldn't solve on my own. Thank you, Tianyi Bai and Ziliang Xiong, for teaching me valuable tools to tackle research problems effectively. Thanks Juan Vidal Alegría, I really enjoy the time that we play table tennis together! Thanks Zhongyunshen Zhu for all the Fika and lunch time together, especially when I encounter difficulties in my life. I wish you enjoy your time at MIT and have a wonderful career in the future! Thanks my other university colleges, collaborators and friends Andreia, Makambi, Anders J Johansson, Maria Jenset Petrén, Erik Ottander, Per Ola Wiking, Erik Danielsson, Simon Hyttfors, João, Erik Mårtensson, Sara, Christian, Harsh Tataria, Ali Al-Ameri, Muris, Fredik Rusek, Jesús, Meifang, Aleksei, Naharika, Jiawang, Yingjie, Xuhong, Wei Zhou, Arthur, Juan Sanchez, Hedieh, Ashkan, Sijia, Russell, Junshi, Vincent, Tian Zhou, Weinan Wang, Manoj, Erik Bengtsson, Marcus Sandberg, Ben Nel, Yuyan, Qiuyan, Paula, Stefan, Emil, Nina, Sofie, Kristoffer, William, Hannes, Karthik, Ufuk, André, Max Halbwachs, and those I may forget to include. All of you have played a significant role in my success.

I would like to sincerely thank my piano teacher Prof. Marianne Jacobs. Though I only play piano as a hobby, every lesson with you has felt like attending a master class. Your professionalism and direct feedback (possibly coming from Dutch directness), such as "Du verstehst Beethoven überhaupt nicht", have been incredibly helpful in identifying my weaknesses and improving my skills. These comments, more direct than anything I've heard from academic supervisors, have been invaluable. The sense of accomplishment I get from music, especially when research feels tough, is something I truly appreciate.

Finally, my deepest gratefulness goes to my parents, who always support me with your endless love. I know it has not been easy for you as I pursue my studies thousands of kilometers away from home, but your constant love and encouragement have carried me through to this success.

Guoda Tian
Lund, Oct 2024.

Content

Preface

This thesis is the culmination of more than five years of work conducted between autumn 2018 and spring 2024. Thanks to the funding provided by an Ericsson project *Massive MIMO technologies and applications*, I was enrolled as a student in the *communications engineering* research group at Lund University in Lund, Sweden. The main focus of this work is to investigate the possibility of leveraging machine learning and signal processing technologies to enhance the performance of various wireless communication systems, particularly massive MIMO systems. The work has been supervised by Professor *Fredrik Tufvesson*, Professor *Ove Edfors*, Professor *Bo Bernhardsson* and assistant Professor *Xuesong Cai*.

STRUCTURE OF THE THESIS

This thesis is composed of an introduction section along with six research papers. The introduction section begins with a high-level overview of the research topic in the first chapter. The following chapter presents preliminary knowledge and provides a detailed discussion of three key aspects: machine-learning-aided channel estimation, sensing, and localization. The concluding chapter summarizes the thesis and provides a vision of future research within this domain. The introduction section is self-contained and includes sufficient material for readers interested in the research topic. The six research papers, published or submitted to scientific journals or conference proceedings, are included in this thesis with permission from the publishers. These articles offer detailed methodologies, analyses, and results related to specific research aspects.

- **INTRODUCTION**
  This thesis addresses an interdisciplinary research field between machine

learning and wireless communication, with a focus on deep learning and physical layers. Chapter 1 offers a high-level overview of the relationship between traditional signal processing and machine learning. It briefly presents potential aspects within the physical layer of wireless communication that could benefit from ML techniques. Chapter 2 presents preliminary knowledge of machine learning and massive multiple-input multiple-output systems, which are essential for comprehending the subsequent chapters of the thesis. Chapter 3 addresses the research topic of wireless propagation channel estimation, a prerequisite for sensing and localization systems. This chapter delves into two crucial fields: RF chain calibration and channel estimation for OFDM systems. Chapter 4 explores the potential of applying machine learning to wireless sensing systems, with a focus on passive sensing of human activity. Chapter 5 applies signal processing and machine learning techniques to various wireless localization systems, including indoor localization with MaMIMO, and outdoor commercial cellular base stations operating in beam spaces. Chapter 6 concludes the introduction part and provides a vision for future work.

- **PAPERS**
  This thesis includes six papers [1–6], research and personal contributions are listed as follows:

INCLUDED PAPERS

Paper I: G. Tian, H. OdetAlla, B. E. Priyanto, S. Hu, and F. Tufvesson, "Modified Gold Sequence for Positioning Enhancement in NB-IoT", *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, 2019, doi: 10.1109/WCNC.2019.8886076.

► **Research Contributions**: This paper proposes a novel positioning reference signal for the Narrowband Internet of things NB-IoT system, which has a limited bandwidth and sampling rate. To address those limitations, we modified the legacy signal to improve cross-correlation properties between transmitted signals from different BSs. Compared to the legacy signal, our proposed method can achieve a $15\% - 30\%$ improvement in terms of positioning accuracy, under various channels defined by the 3GPP standard.

► **Personal Contributions**: This work results from my Master's thesis at Sony Communication AB during the spring of 2018. I am the main

contributor to this work. I developed the main idea and performed the mathematical derivation. I developed the simulation framework for evaluating the performance together with my colleagues, Mr. OdetAlla and Dr. Priyanto. I was the main author of this paper.

Paper II: G. Tian, H. Tataria, and F. Tufvesson, "Amplitude and Phase Estimation for Absolute Calibration of Massive MIMO Front-Ends", *2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, doi: 10.1109/ICC40277.2020.9148962.

▶ **Research Contributions**: This paper addresses the research problem of array calibration. We proposed a novel method for estimating the RF chain amplitude and phase using uplink pilot signals, which can be utilized to calibrate the massive MIMO array. In addition, our method does not necessarily have to be implemented inside an anechoic chamber. Two *closed-form* estimators, namely the MLE and the moment estimators, are derived to estimate the amplitude scalings and phase drifts of RF chains, respectively. To evaluate the proposed estimator, the corresponding *closed-form* CRLB is also derived. This bound can indicate the performance of our method under different propagation scenarios and different signal-to-noise ratios. A simulation framework is also developed to evaluate our method.

▶ **Personal Contributions**: This work was written in an early phase of my Ph.D. study. I was the main contributor to this work. I developed the main idea with my supervisors, performed the mathematical derivation, and validated the proposed method through synthetic data. Finally, I wrote the article with the support of my supervisors.

Paper III: G. Tian, X. Cai, T. Zhou, Weinan. Wang, and F. Tufvesson, "Deep-Learning Based Channel Estimation for OFDM Wireless Communications", *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, Oulu, Finland, 2022, doi: 10.1109/SPAWC51304.2022.9834008.

▶ **Research Contributions**: This paper addresses the research problem of deep-learning-based channel estimation for multi-carrier systems. It provides a novel deep-learning pipeline to estimate channel transfer functions for an OFDM system by leveraging pilot signals. A simulation

framework is developed to evaluate our pipeline. The results show that our channel estimation pipeline performs significantly better than the traditional signal processing algorithm based on an LMMSE estimator. In addition, our algorithm shows its robust behavior under low SNR scenarios.

► **Personal Contributions**: I was one of the main contributors to this work. I developed and implemented the channel estimation algorithm and validated the proposed method using synthetic data together with coauthors. Finally, I took major responsibility for the writing.

Paper IV: B. R. Manoj, G. Tian, S. Willhammar, F. Tufvesson and E. G. Larsson, "Sensing and Classification Using Massive MIMO: A Tensor Decomposition-Based Approach", *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2649-2653, Dec. 2021, doi: 10.1109/LWC.2021.3110463.

► **Research Contributions**: This paper tackles the problem of device-free wireless sensing using uplink signals. This work presents a novel activity classification framework, consisting of a tensor-decomposition block for dimension reduction and an ML block for classifying activities. We validated our framework by indoor activity sensing measurements. The results demonstrate the strong sensing ability of massive MIMO systems in comparison to systems with fewer antennas. This work is also one of the first papers that investigates wireless activity sensing using massive MIMO systems.

► **Personal Contributions**: This paper is the result of a cooperation with the communication group at Linköping University. I was one of the main contributors to this work. I was responsible for developing the framework to implement tensor decomposition and feature extraction. I also participated in the preparation and conducting of the measurement campaign. Finally, I wrote section II and sections III. A and III. B and contributed to other sections.

Paper V: G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu and F. Tufvesson, "Deep-Learning-Based High-Precision Localization With Massive MIMO", *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 19-33, 2024, doi: 10.1109/TMLCN.2023.3334712.

► **Research contribution**: This paper tackles the high-precision user UE localization problem for massive MIMO systems using deep learning. We investigated different channel fingerprints for UE localization, namely the truncated channel impulse response and covariance matrices. To merge different channel fingerprints, we present a novel pipeline that combines different channel fingerprints by applying ensemble learning methods. This pipeline can predict both the UE position and error variance as a Gaussian distribution, indicating the prediction uncertainty that is critical for many safety-critical systems such as autonomous driving. We also investigated the necessary training density using the Nyquist sampling theorem, which provides insight to adequately sample and average training data. In general, this is the first paper on machine learning wireless localization that addresses fingerprint investigation, ensemble learning, uncertainty estimation, and training density altogether. Our pipeline is evaluated by an indoor measurement campaign with a massive MIMO testbed. The measurement results show that our pipeline can reach very good localization accuracy (centimeter level).

► **Personal Contributions**: This paper results from a collaboration project with the Lund University Humanities Lab and the LTH robotic lab. I was the main contributor to this paper. I developed and implemented the localization pipeline. I took the main responsibility for the radio system part of the measurement campaign. In addition, I wrote most of the manuscript and was responsible for communicating with the editorial board of the journal.

Paper VI: G. Tian, D. Pjanić, X. Cai, B. Bernhardsson, and F. Tufvesson, "Attention-aided Outdoor Localization in Commercial 5G NR Systems", submitted to *IEEE Transactions on Machine Learning in Communication and Networking*, second round, after major revision.

► **Research contributions**: This paper addresses the research topic of cellular localization with commercial fifth-generation (5G) new radio (NR) systems operating in the beam domain. For this specific system, a novel machine-learning-based single-snapshot localization pipeline is presented, which utilizes attention-aided neural networks as its core. As a continuation of *paper V*, a more advanced uncertainty estimation tech-

nology is applied, which does not necessarily model prediction error as a Gaussian distribution. Different uncertainty estimation techniques are compared as well. A Kalman filter is applied to our pipeline to address multiple snapshot localization. Our pipeline is also evaluated with an outdoor measurement campaign with real mobile phones and 5G base stations. The measurement campaign covers a variety of common outdoor propagation scenarios. In summary, this is one of the first few wireless localization papers, using attention-aided technologies, investigating various types of uncertainty estimation techniques, and verifying the main algorithm with a commercial-grade 5G base station.

► **Personal Contributions**: This paper is a collaboration with Ericsson Lund. Dino Pjanic and I contributed equally to this paper. My main duty was the development and implementation of the algorithm. I also participated in the measurement campaign. We are both responsible for writing the manuscript with the support of our supervisors.

RELATED WORK

During my Ph.D. study, I have also presented my research results as a temporary documents (TD) in the European Cooperation in Science and Technology (COST) action CA20120 and contributed the following publications, which are not included in my thesis.

- B. R. Manoj, G. Tian, S. Willhammar, F. Tufvesson and E. G. Larsson, "Moving Object Classification with a Sub-6 GHz Massive MIMO Array Using Real Data", *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, doi:10.1109/ICASSP39728.2021.9414952.
- M. Sandra, G. Tian, X. Cai, A. J. Johansson, "Antenna Array Configuration for Reliable Communications in Maritime Environments", *2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring)*, Helsinki, Finland, 2022, doi: 10.1109/VTC2022-Spring54318.2022.9860440.
- M. Sandra, G. Tian, A. Fedorov, X. Cai, A. J. Johansson, "Measurement-Based Wideband Maritime Channel Characterization", *2023 17th European Conference on Antennas and Propagation (EuCAP)*, Florence, Italy, 2023, doi: 10.23919/EuCAP57121.2023.10133372.
- G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu and F. Tufvesson, "High-Precision Machine-Learning Based Indoor Localization with

Massive MIMO System", *2023 IEEE International Conference on Communications (ICC)*, Rome, Italy, 2023, doi: 10.1109/ICC45041.2023.10278664.

- I. Yaman, G. Tian, M. Larsson, P. Persson, M. Sandra, A. Dürr, E. Tegler, N. Challa, H. Garde, F. Tufvesson, K. Åström, O. Edfors, S. Malkowsky, L. Liu "The LuViRA Dataset: Synchronized Vision, Radio, and Audio Sensors for Indoor Localization", *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, 2024.
- I. Yaman, G. Tian, E. Tegler, J. Gulin, N. Challa, F. Tufvesson, O. Edfors, K. Åström, S. Malkowsky, L. Liu "LuViRA Dataset Validation and Discussion: Comparing Vision, Radio, and Audio Sensors for Indoor Localization", *IEEE Journal of Indoor and Seamless Positioning and Navigation, doi: 10.1109/JISPIN.2024.3429110.*
- G. Tian, I. Yaman, X. Cai, L. Liu and F. Tufvesson, "High accuracy indoor wireless localization using a real Massive MIMO Testbed", *in COST CA20120 Technical Meeting, TD(22)03063, Valencia, Spain, September 20-22, 2022*

Acronyms and Symbols

ACRONYMS AND ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5G	5th Generation
AI	Artificial Intelligence
AoA	Angle of Arrival
AUSE	Area Under the Sparsification Error
AWGN	Additive White Gaussian Noise
BMSE	Bayesian Mean Squared Error
BS	Base Station
CNN	Convolutional Neural Network
CP	Cyclic Prefix
CRLB	Cramér–Rao Lower Bound
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCT	Discrete Cosine Transform
DL	Deep Learning
DSP	Digital Signal Processing
FCNN	Fully Connected Neural Network
FFT	Fast Fourier Transform
FIM	Fisher Information Matrix
GPU	Graphics Processing Unit
IFFT	Inverse Fast Fourier Transform
LMMSE	Linear Minimum Mean Square Error Estimation
LoS	Line of Sight

LSTM	Long Short-Term Memory
LTE	Long-Term Evolution
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MSE	Mean Squared Error
MUSIC	Multiple Signal Classification
NB-IoT	Narrowband Internet of Things
NLL	Negative Log-Likelihood
NLP	Natural Language Processing
NN	Neural Network
NPRS	Narrowband Positioning Reference Signal
NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
PAPR	Peak-to-Average Power Ratio
PCA	Principal Component Analysis
PCI	Physical Cell ID
QPSK	Quadrature Phase Shift Keying
RbC	Regression by Classification
RF	Radio Frequency
RNN	Recurrent Neural Network
SAGE	Space-Alternating Generalized Expectation-Maximization
SNR	Signal-to-Noise Ratio
SRS	Sounding Reference Signals
SVM	Support Vector Machine
TDoA	Time Difference of Arrival
ToA	Time of Arrival
TPU	Tensor Processing Unit
UE	User Equipment

INTRODUCTION

1

Overview

The concept of applying machine learning (ML) to improve wireless communication system performance dates back to the late 1990s [7]. The authors in [7] explored the use of simple and standard ML algorithms, i.e., support vector machines, for performing channel equalization tasks. However, hardware limitations at that time prevented the widespread adoption of ML-based algorithms on a large scale. In recent years, advances in artificial intelligence (AI) related hardware have significantly improved computing power, enabling the application of advanced learning algorithms in many fields, including image recognition [8], speech recognition [9], natural language processing (NLP) [10, 11], and autonomous driving [12, 13]. Similarly, in the field of wireless communication, various tasks such as beam management [14], sensing [15], localization and navigation [16], and channel estimation [17] have been addressed using ML, yielding substantial results.

This chapter first explores the relationship between ML and traditional signal processing algorithms. Then it introduces three categories of ML algorithms: supervised learning, unsupervised learning, and reinforcement learning. Finally, it discusses the potential applications of ML in wireless communication systems.

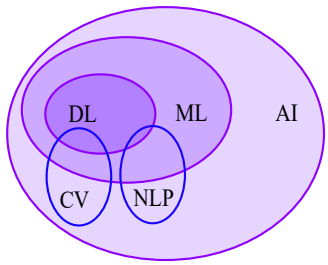


Figure 1.1: Relationships between AI, ML and DL.

1.1 THE ROAD FROM DIGITAL SIGNAL PROCESSING TO DEEP LEARNING

1.1.1 AN OVERVIEW OF DSP AND ML

Looking back at the history of computer development, it is a fascinating journey for society to witness the evolution of various data processing techniques. The principles of digital signal processing (DSP) were formulated in the 1960s, consisting of spectral analysis and digital filter design [18]. These techniques were widely applied for tasks such as denoising, data compression, feature extractions, parameter estimations, etc. One of the milestones was the development of the fast Fourier transform (FFT) algorithm in 1965 by Cooley and Tukey [19], which significantly reduced the computational burden and increased the efficiency of many algorithms used for spectral analysis and waveform generation, including widely used techniques like orthogonal frequency division multiplexing (OFDM). Although the advantages of DSP, such as flexibility, scalability, and robustness to environmental changes, were well recognized at that time, its usage was still limited to some specific applications, mainly military, due to its costs. Later, the advancement of integrated circuit technology enabled the massive implementation of more complex algorithms, leading to the development of more advanced DSP applications [20].

Another transformative area in computing that emerged in the 1950s is ML. Many readers may be confused by the connections between ML, deep learning (DL), and AI; therefore, we depict their relationships in Fig. 1.1. ML is a subset of AI. Various fields in AI, including computer vision (CV), and NLP, encompass concepts that extend beyond ML, although they can undoubtedly leverage ML to enhance their capabilities and performances. The idea of ML was proposed in the 1950s [21], which could identify alphabets. However, due to the constraint of computational power, the usage of ML was not pushed forward until the 1990s. At that time, early methods including

linear regressions, decision trees, support vector machines (SVM), and neural networks (NN) paved the way for more sophisticated techniques that emerged in the 21st century [22]. SVM was introduced in the 1990s as a powerful classification algorithm aiming at finding the optimal hyperplane to separate data points into different classes. NN methods were inspired by the structure of human brains and pushed the frontiers of various fields with their ability to learn hierarchical representations and highly non-linear relationships from raw data. Today, thanks to the further enhancement of computational power by more advanced devices such as the graphic processing unit (GPU) and the tensor processing unit (TPU), NNs are much deeper and more complex, containing gigantic sets of parameters. This further enables advances such as image recognition, natural language processing, etc.

1.1.2 COMPARISON BETWEEN DSP AND ML METHODS

Fruitful results have already been harvested by applying traditional signal processing methods, but there are still some situations that are hard for their usages. In certain cases, traditional DSP algorithms are used for processing structured and/or well-defined signals. Well-defined signals have mathematical descriptions. To be more specific, specific patterns or regularities can be seen in structured signals, which may even repeat over time, frequency, or space. However, a vast majority of the signals in real life may not have a very clear pattern, or it is difficult to extract the pattern due to practical reasons. As an example, we consider the following two tasks: (a) The detection of signal 1: $y_1(t) = a_1 sin(\omega_1 t + \phi_1)$ and signal 2: $y_2(t) = a_2 sin(\omega_2 t + \phi_2)$ in a noisy environment, where the signal-to-noise-ratio (SNR), amplitudes a_1 and a_2, frequencies ω_1 and ω_2 and phases ϕ_1 and ϕ_2 are known. (b) Image identification of a cat or a dog. We can see that task (a) contains a well-defined signal while in task (b) it is challenging to write exact mathematical formulas to describe cats and dogs. Therefore, task (a) is more suitable for DSP methods and a possible solution is to apply the Neyman-Person detector [23], while task (b) can in an easier way be addressed by ML methods.

However, we need to realize that the implementation of the ML algorithm normally requires a large amount of resources and budget. Therefore, before applying ML algorithms, it is imperative to undertake some critical assessments. Can we harvest the potential benefits by applying ML? In the following, short summaries of the advantages and disadvantages of ML methods are given. From my perspective, the suitable cases for involving ML-based algorithms are, but not limited to,

- ML can help reduce the computational complexity of an algorithm.
- ML algorithms can achieve significantly better results than traditional methods. In extreme cases, there is no clear traditional method.

- The task itself is strongly related to the handling of large and diverse data with complex patterns and relationships.
- The task itself involves an optimization procedure, where it is non-trivial to mathematically find a clear optimal solution.
- When it is challenging to establish an accurate mathematical model.

In contrast, it is challenging for ML methods to offer meaningful contributions, provided that the tasks fall into the following categories:

- There exist well-grounded traditional methods and it is unnecessary or not possible to further reduce the complexity. One example is a parameter estimation problem where a simple closed-form estimation already exists that can reach the optimal solution.
- Analysis of encrypted traffic. The difficulty mainly arises due to the inherent limitations imposed by the encryption itself. As a result, it is challenging to recognize the data pattern, as well as to collect enough labeled training data for the ML algorithms.

1.1.3 HOW CAN ML BENEFIT FROM DSP METHODS?

Although ML is better suited for processing signals that are not well defined, one should not forget the benefits of traditional DSP methods that still offer valuable contributions to ML algorithms. For example, [24] demonstrates that applying the Discrete Cosine Transform (DCT) to an image can reduce the size of input data while simultaneously improving classification accuracy. This is achieved by transforming the image into the frequency domain, where less important high-frequency components are discarded, enabling more efficient and accurate image recognition. From my perspective, the advantages of combining DSP with ML algorithms include:

- Noise and interference reduction and signal-to-noise quality enhancement, examples are anti-aliasing filters, out-of-band noise filters, beamformers, etc.
- Extracting specific patterns. For example, both [5] and [25] demonstrate that extracting and training on distinct channel features, such as the angle-delay profile or covariance matrices, can significantly enhance localization accuracy compared to training on raw transfer functions.

1.2 CATEGORIES OF ML AND TYPICAL TASKS

Existing ML algorithms may be mainly categorized into three categories based on their learning approaches [22], that is, i) supervised learning, ii) unsupervised learning and iii) reinforcement learning. Each category is associated with typical tasks and applications.

1.2.1 SUPERVISED LEARNING

As one of the fundamental ML categories, supervised learning requires a *labeled* dataset, where each input is associated with a corresponding output. Supervised learning is used to learn a mapping from inputs to outputs using labeled training data. Both the quality and quantity of the training data significantly affect the performance. To evaluate the performance of learning algorithms, one should test new data not used during the training phase. If the training data covers only limited scenarios, the algorithm's performance will typically be poor when it encounters use cases that are not adequately represented in the training phase.

Supervised learning technologies can be used for two main tasks, namely, classification and regression [22]. The classification task aims to categorize different input data by assigning a label (or multiple labels). In simple terms, performing a classification task is similar to solving a multiple choice question. For example, email spam detection requires a binary response, where the answer is either affirmative or negative. The image detection task aims to classify images. Similarly, within the domain of medical diagnosis, the object is to detect diseases. The other important task under the umbrella of supervised learning is regression, which aims to predict continuous numerical values rather than labels based on input data. Examples of regression tasks include predicting house prices, estimating the propagation channel or the localization of the user equipment (UE), etc.

1.2.2 UNSUPERVISED LEARNING

Different from supervised learning, unsupervised learning algorithms focus on identifying patterns and structures of datasets, which are not labeled. Typical tasks of unsupervised learning include clustering and dimensionality reduction [22]. Clustering involves grouping similar data points based on their inherent patterns, ensuring that objects within the same group exhibit significant similarity compared to those in other groups. Popular clustering algorithms include K-means, density-based spatial clustering of applications with noise (DBSCAN), etc [26]. The applications of this category vary from market segmentation to social network analysis. Clustering is also widely applied in the design of wireless networks, see [26] for further details. Dimensional reduction is also an important technique in unsupervised learning, which aims to transform high-dimensional data into a lower-dimensional representation while still preserving essential information [27, 28] without relying on labeled groundtruth data. A representative algorithm is principal component analysis (PCA), which reduces the data dimension by extracting the principal components associated with the largest singular values or

eigenvalues. One of the main applications of dimension reduction in wireless communication is the so-called channel charting [29, 30]. In summary, this technique creates a channel chart that captures and preserves the spatial geometry information of the UE from the received channel, ensuring that points close in space are also close in the channel chart. Dimension reduction algorithms can also extract important channel features for wireless sensing [4].

1.2.3 REINFORCEMENT LEARNING

As another important subset of ML, reinforcement learning trains agents or algorithms to interact with different environments autonomously [31, 32]. Classic reinforcement learning algorithms include the Markov decision process [31], the Monto-Carlo method [31, 32], etc. Unlike supervised learning, reinforcement learning operates in scenarios where clear labels or predetermined values may not be readily available. Instead, learning performance is improved by adjusting the algorithm parameters to maximize cumulative rewards, based on the feedback from each iteration. Feedback can range from binary success/failure indicators to complex criteria-based scoring.

Diverse applications utilize reinforcement learning techniques, including strategic gameplay, for example, AlphaGo and Alphastar [33, 34]. Reinforcement learning has shown significant use in real-world challenges, particularly in robotics, by enabling precise control of complex mechanical movements [35]. In wireless communication systems, reinforcement learning is a powerful tool for optimizing transmission schemes [36] and efficiently mitigating spectrum jamming [37]. Specifically, [36] proposes a distributive, model-free deep reinforcement learning algorithm to optimize the transmit power control scheme in wireless networks. [37] proposes a deep reinforcement learning algorithm based on a recursive convolutional neural network that optimizes anti-jamming strategies without requiring prior knowledge of jamming patterns. As seen from the aforementioned examples, reinforcement learning plays a crucial role in advancing cutting-edge technological landscapes.

1.3 APPLYING ML TO WIRELESS COMMUNICATION SYSTEMS

Wireless systems have made significant advances in the past 50 years, progressing beyond their initial purpose of simple message transmission. Driven by society's demand, researchers and engineers have actively explored more advanced functionalities for wireless technology, such as modeling and estimating complex communication channels, advanced sensing, and high-precision localization [38]. In this section, we focus on the physical layer and briefly explain the benefits of using ML in wireless communication systems.

We provide examples of ML-aided techniques in channel estimation, sensing, and localization. More details will be introduced in the following chapters.

1.3.1 ESTIMATING CHANNEL TRANSFER FUNCTION

Acquiring propagation channel knowledge plays a pivotal role in designing wireless communication, sensing, and localization systems. However, it is challenging to estimate a precise channel transfer function due to the complex behavior of the wireless channel. The wireless channel can vary rapidly and contain excessive information in the frequency and spatial domains, necessitating more effective channel estimation algorithms capable of handling massive amounts of data and highly dynamic environments. Additionally, the non-linear effects of radio frequency (RF) chains add to these challenges. These factors make classic parameter estimation algorithms inherently time-consuming and less efficient [39]. To address these challenges, data-driven-based methods offer promising advancements due to their efficiency and ability to handle nonlinearity effectively [17, 40]. Specifically, [17] explores a DL-based approach that effectively mitigates channel non-linear distortion, demonstrating robust performance in simulations compared to conventional methods, particularly when fewer training pilots are employed. [40] shows that by integrating expert knowledge to the proposed OFDM channel estimation network, one can achieve more accurate channel estimation and higher data recovery accuracy compared to conventional methods.

1.3.2 SENSING AND LOCALIZATION

The role of wireless sensing is to remotely acquire and exploit information and characteristics of interesting objects through received radio signals that change with the movement of objects. For certain applications such as object speed estimation, there already exist mature signal processing algorithms: one can attain the speed estimation by analyzing the Doppler spectrum. However, it is more suitable to apply ML methods for tasks such as object classification, since attaining an accurate model for the propagation scenario can be a non-trivial task, especially when the base station (BS) cannot obtain the size and moving speeds of objects. In contrast, this relationship can be learned by exploiting large datasets [41, 42].

Localization services have been identified as a key function in modern cellular systems, and relevant localization algorithms have been investigated for a few decades. Several localization algorithms, based on proximity, angle of arrival (AoA), and time of arrival (ToA), have been integrated into the current 3rd Generation Partnership Project (3GPP) standards [43]. To further enhance localization accuracy and reduce algorithmic complexity, an

ML-based method can be applied, which learns the non-linear relationship between the received channel state information and the UE location ,see [44, 45].

2

Preliminaries

This chapter briefly presents fundamental concepts and technologies in ML algorithms and multiple antenna systems. We begin with introducing two important tasks, namely regression and classification, followed by illustrating fundamentals of NNs. Next, we illustrate the basics of massive multiple-input multiple-output (MIMO) systems, and introduce a MIMO system model and directional beamforming.

2.1 MACHINE LEARNING BASICS

This section begins with explaining two representative classic ML algorithms under the umbrella of supervised learning, namely linear regression and logistic regression. These two algorithms can be applied to solve regression and classification tasks, respectively, see [22, 46]. We also use them as examples to illustrate the general process of supervised learning, where a crucial consideration is to use an appropriate cost function. The cost functions introduced here will be used in the following chapters to address wireless channel estimation, sensing, and localization tasks. Next, we introduce the basic structures of NNs, attention mechanisms, and ensemble learning concepts.

2.1.1 LINEAR REGRESSION

We begin by introducing the basics of linear regression algorithms, which model the system output as a linear combination of the inputs. For simplicity, we consider a system that takes a vector $\mathbf{x} \in \mathbb{R}^N$ as input and outputs a scalar y, although the fundamental principles can be easily extended to systems with

multiple outputs. To this end, y is represented as [22, 46]:

$$y = b + \sum_{n=1}^{N} \omega_i x_i + \epsilon, \tag{2.1}$$

where b denotes the optional bias term and ω_i represents the weight of the i-th element of the vector $\mathbf{x}$. In addition, $\hat{y} = \sum_{n=1}^{N} \omega_i x_i + b$ denotes the predicted value of the linear regression algorithm and $\epsilon = y - \hat{y}$ the prediction error. Observe that if the input and output relationship is strongly non-linear, it might be inappropriate to directly apply a linear regression to learn this relationship. Instead, one can apply various non-linear base function(s) $\phi_j(.)$ to $\mathbf{x}$, formulating a vector $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{N'} = \big[\phi_1(\mathbf{x}), ..., \phi_{N'}(\mathbf{x})\big]^T$. An example of a base function is $\phi_j(x) = x^j$, which formulates the idea of polynomial regression [22]. Let $\boldsymbol{\omega} = \big[\omega_1, ..., \omega_N\big]^T \in \mathbb{R}^{N'}$ denote the new weight vector, (2.1) can then be modified as

$$y = b + \boldsymbol{\omega}^T \boldsymbol{\phi}(\mathbf{x}) + \epsilon. \tag{2.2}$$

During the training phase, the task is to find all $\boldsymbol{\omega}$ so that the model output approaches the ground truth. To do this, it is required to collect enough training data. We assume that a total of P input vectors $\mathbf{x}$ and corresponding P output scalars are collected. The j-th input vector and the j-th output scalar are denoted as $\mathbf{x}_j$ and y_j respectively. We then define a matrix $\boldsymbol{\Phi} \in \mathbb{R}^{P \times N'}$ with its j-th row as $\boldsymbol{\phi}^T(\mathbf{x}_j)$, and a new vector $\mathbf{y} \in \mathbb{R}^P$ with the j-th element as y_j. The error ϵ is usually supposed to be Gaussian distributed with variance σ^2, and all elements $\mathbf{x}_j$ are statistically independent, while all elements of $\boldsymbol{\omega}$ are treated as fixed unknown values rather than random variables. Considering this, the overall likelihood function can be written as [22]

$$p(\mathbf{y}|\boldsymbol{\omega}) = \prod_{j=1}^{P} \exp\Big\{-\frac{(y_j - \boldsymbol{\omega}^T \boldsymbol{\phi}_j(\mathbf{x}_j) - b)^2}{2\sigma^2}\Big\}. \tag{2.3}$$

In (2.3), $\boldsymbol{\omega}$ can be estimated by maximizing the likelihood function (2.3) or minimizing the following mean square error (MSE) cost function [22]

$$E(\boldsymbol{\omega}) = \sum_{j=1}^{P} \big\{y_j - \boldsymbol{\omega}^T \boldsymbol{\phi}_j(\mathbf{x}_j) - b\big\}^2. \tag{2.4}$$

Equation (2.4) is a measure of the mean square error between the predicted output and the ground truth, which is a popular cost function in regression

tasks. The closed-form solution of $\boldsymbol{\omega}$ and b can be found by taking the derivative of $E(\boldsymbol{\omega})$ w.r.t. each ω_i and b [22], specifically,

$$\boldsymbol{\omega} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{y}, \quad b = \frac{1}{P}\sum_{i=1}^{P} y_i - \frac{1}{N'}\sum_{i=1}^{N'}\omega_i \sum_{j=1}^{P}\boldsymbol{\Phi}_{ij}. \tag{2.5}$$

2.1.2 FUNDAMENTALS OF ML-BASED CLASSIFICATION ALGORITHMS

When it comes to classification tasks, we can again apply the concept of maximum likelihood. However, one needs to use another cost functions instead of (2.4). In this section, we first present the basics of classification tasks and provide the derivation of a widely-used cost function, namely the cross-entropy function. We consider that a classification task that is associated with K possible output classes labeled as $\mathcal{C}_1, ..., \mathcal{C}_K$ and the prior probability of the k-th event is written as $p(\mathcal{C}_k)$. The posterior probability of the k-th class $p(\mathcal{C}_k|\mathbf{x})$ given an input vector $\mathbf{x}$, can be calculated using Bayes' theorem as follows [22]:

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)\,p(\mathcal{C}_k)}{\sum_m p(\mathbf{x}|\mathcal{C}_m)\,p(\mathcal{C}_m)} = \frac{\exp(a_k)}{\sum_m \exp(a_m)}, \tag{2.6}$$

where $a_k = \ln p((\mathbf{x}|\mathcal{C}_k)\,p(\mathcal{C}_k))$ is the log probability. Our aim is to find the relationship between a_k and $\mathbf{x}$. There are various ways to approach this, one common way is to assume that a_k and $\mathbf{x}$ have linear relationships and this refers to the *logistic regression* algorithm*. Specifically, a_k is expressed as $a_k = \tilde{\boldsymbol{\omega}}_k^T\mathbf{x}$, where $\tilde{\boldsymbol{\omega}}_k \in \mathbb{R}^N, k = 1, ..., K$ denotes the weight vector.

Now, consider that we have collected N input vectors $\mathbf{x}_n, n = 1, ..., N$ and that a logistic regression algorithm is applied. Further, we denote further $p(C_k|\mathbf{x}_n) = \frac{\exp(a_{n,k})}{\sum_m \exp(a_{n,m})}$ as the posterior probability that the output label is C_k when $\mathbf{x}_n$ is the input vector, while $a_{n,k} = \tilde{\boldsymbol{\omega}}_k^T\mathbf{x}_n$ is the corresponding log probability. We further denote a matrix $\mathbf{T} \in \mathbb{R}^{K\times N} = [\mathbf{t}_1, ..., \mathbf{t}_N]$ that records the ground truth label for all N input vectors. The element $\mathbf{T}_{k,n} = 1$ if for the n-th input vector, the event label is C_k. Otherwise, $\mathbf{T}_{k,n} = 0$. During the training process, our object is that $p(C_k|\mathbf{x}_n)$ approaches $\mathbf{T}_{k,n}$. Therefore, we define the following likelihood function $p(\mathbf{T}|\tilde{\boldsymbol{\omega}}_1, ..., \tilde{\boldsymbol{\omega}}_k)$ as

$$p(\mathbf{T}|\tilde{\boldsymbol{\omega}}_1, ..., \tilde{\boldsymbol{\omega}}_k) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\phi_n)^{t_{nk}}. \tag{2.7}$$

*Logistic regression is a classification algorithm that predicts a continuous outcome representing the probability of a class.

We aim to find all $\tilde{\omega}_k$ to maximize (2.7), or to minimize the cross-entropy $E(\tilde{\omega}_1, ..., \tilde{\omega}_k)$ between t_{nk} and y_{nk}, specifically,

$$E(\tilde{\omega}_1, ..., \tilde{\omega}_k) = -\ln p(\mathbf{T}|\tilde{\omega}_1, ..., \tilde{\omega}_k) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln p(\mathcal{C}_k|\phi_n). \tag{2.8}$$

Equation (2.8) is the widely used cross-entropy cost function to address classification tasks. All weights $\tilde{\omega}_j$ can be estimated by applying gradient descent algorithms, see [22, 46].

2.1.3 AN INTRODUCTION TO NEURAL NETWORKS

NNs have been widely applied to solve complex tasks such as image classification [8], speech recognition [47], owing to their ability to learn nonlinear models, as highlighted by [46]. Typically, NN structures can be represented by multivariate functions $f : \mathbb{R}^{M_1} \to \mathbb{R}^{M_2}$, where M_1 and M_2 denote the dimensions of learning characteristics and desired outcomes, respectively. In this chapter, we focus on introducing the fully connected neural network (FCNN) as an illustrative example. The reader is referred to [22] for more detailed information on other typical NNs, such as the convolutional neural network (CNN), recurrent neural network (RNN), long-short-term memory (LSTM), etc. Fig. 2.1 presents an example of a typical FCNN architecture, comprising an input layer, several hidden layers, and an output layer. The number of nodes in the input and output layers is M_1 and M_2, respectively.

In the process of training a NN, two fundamental steps are necessary, namely forward propagation and backward propagation. Specifically in the forward propagation step, input signals are introduced through the NN's input layer, traversing multiple hidden layers before ultimately reaching the output layer. The output of the NNs vary based on individual tasks; for regression tasks, the NN yields predicted values, while in classification tasks, it produces the probabilities associated with each label. At each layer, the output of a node depends on inputs from preceding layers, the corresponding weights and biases, and a distinctive nonlinear activation function exclusive to that node. To elaborate the process, we consider an FCNN with γ_i nodes in the i-th layer, and the values of these nodes are aggregated into a signal vector $\mathbf{x}^i = [x_1^i, ..., x_{\gamma_i}^i] \in \mathbb{R}^{\gamma_i}$. The value of the k-th node can be calculated by applying a weight vector $\hat{\mathbf{w}} = [\hat{w}_1^{i-1}, ..., \hat{w}_{\gamma_i-1}^{i-1}] \in \mathbb{R}^{\gamma_{i-1}}$ to the signal vector $\mathbf{x}^{i-1}$ from the previous layer. This computation is specifically expressed as

$$x_k^i = g_i\left(\sum_{j=1}^{\gamma_{i-1}} x_j^{i-1}\hat{w}_j^{i-1}\right) + b_i. \tag{2.9}$$

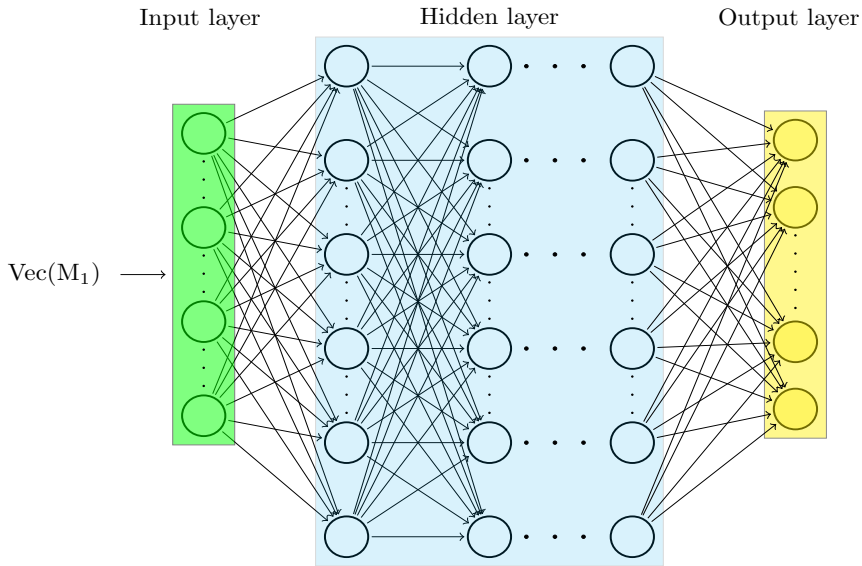


Figure 2.1: A typical structure of an FCNN.

Here, b_i denotes an optional bias term, and $g_i(.)$ symbolizes the activation function applied to the nodes in the i-th layer. The same propagation pattern repeats for each subsequent layer, ultimately generating the predicted values or probabilities of each label, depending on the task being addressed. For supervised learning, it is important to select an appropriate training criterion, often referred to as the loss function, since this function quantifies the disparity between the predicted value and the ground truths. As previously stated, the Mean Squared Error (MSE) in (2.4) is a widely used cost function to perform regression tasks, while the cross entropy in (2.8) to perform classification tasks. Once the training criterion is chosen, the optimization process involves fine-tuning all hyperparameters, specifically the weights and bias terms within each layer, as denoted in (2.9), to minimize the losses. This optimization procedure is executed through backward propagation, where the final loss based on the selected criteria is propagated backward through each layer of the NN to facilitate weight update. More relevant material can be found in [22, 46].

2.1.4 ATTENTION-AIDED MECHANISM

The attention mechanism was originally proposed by [11] to address tasks related to language translation. The concept of the attention mechanism is presented in Fig. 2.2. As shown, a matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M] \in \mathbb{R}^{N \times M}$ is fed into the attention block as input, resulting in another matrix $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_M] \in$

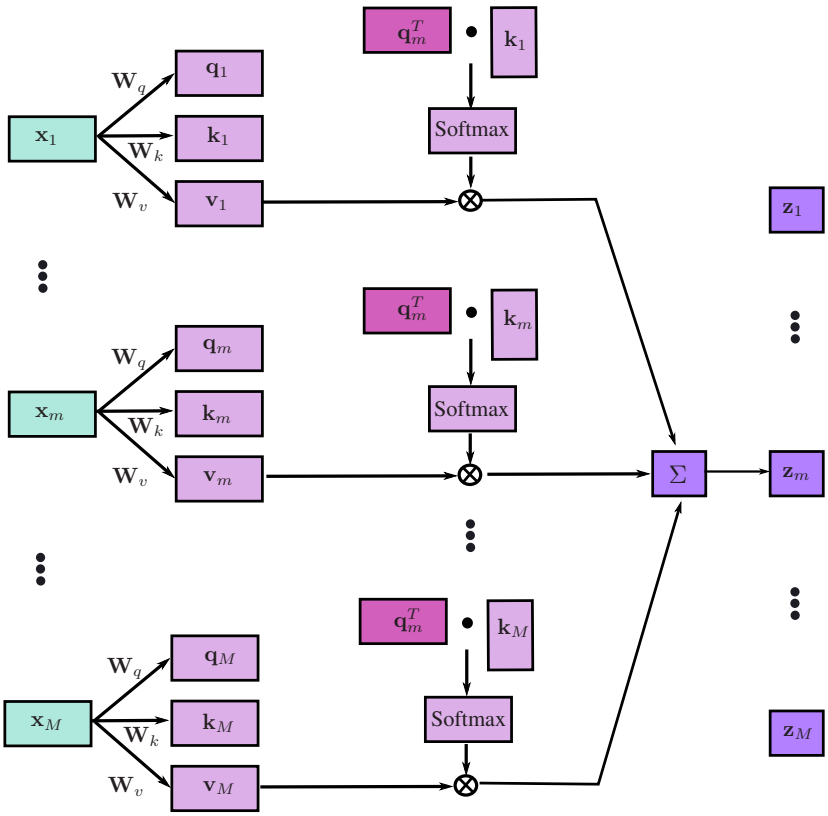


Figure 2.2: An illustration of basic attention mechanism to generate $\mathbf{z}_j$ and same mechanism can be applied to generate $\mathbf{Z}$.

$\mathbb{R}^{N' \times M}$ as output. We define three matrices: the query matrix $\mathbf{W}_q \in \mathbb{R}^{N \times N}$, the key matrix $\mathbf{W}_k \in \mathbb{R}^{N \times N}$ and the value matrix $\mathbf{W}_v \in \mathbb{R}^{N' \times N}$. All elements of these three matrices are parameters that need to be optimized in the training phase. By multiplying $\mathbf{X}$ by those three matrices, we obtain

$$\mathbf{W}_q \mathbf{X} = \mathbf{Q} \in \mathbb{R}^{N \times M}, \ \mathbf{W}_k \mathbf{X} = \mathbf{K} \in \mathbb{R}^{N \times M}, \ \mathbf{W}_v \mathbf{X} = \mathbf{V} \in \mathbb{R}^{N' \times M}. \tag{2.10}$$

We then conduct the "attention" operation, which is the pairwise correlations between all columns of matrices $\mathbf{Q}$ and $\mathbf{K}$ to obtain matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$. Specifically,

$$\mathbf{A} = \frac{1}{\sqrt{N}} \mathbf{Q}^T \mathbf{K}. \tag{2.11}$$

The matrix $\mathbf{A}$ is then normalized by *softmax* operation to attain $\hat{\mathbf{A}} \in \mathbb{R}^{M \times M}$. Each element $\hat{\mathbf{A}}_{i,j}$ is expressed as

$$\hat{\mathbf{A}}_{i,j} = \frac{\exp \mathbf{A}_{i,j}}{\sum_k \exp \mathbf{A}_{i,k}}. \tag{2.12}$$

This normalization ensures that all elements of $\hat{\mathbf{A}}$ are positive, and that the sum of all elements in each column equals 1. Finally, the output matrix $\mathbf{Z}$ can be calculated as

$$\mathbf{Z} = \mathbf{V}\hat{\mathbf{A}}. \tag{2.13}$$

Bear in mind that the attention mechanism shown in (2.10)-(2.13) processes the input vectors in $\mathbf{X}$ in a parallel way, which discards the sequence of order of vectors. Therefore, if important information is presented in the order of vector arrangement, it is essential to use a *positioning encoding* technique to incorporate and preserve this sequential information. One solution is to add a fixed positioning encoding matrix $\mathbf{X}_k \in \mathbb{R}^{N \times M}$ to $\mathbf{X}$ [11]. A standard example of $\mathbf{X}_k$ (also used in [11]) is

$$\begin{aligned} \mathbf{X}_k(x, y) &= \sin\left(\frac{x}{10000^{y/N}}\right), \text{for odd } y; \\ \mathbf{X}_k(x, y) &= \cos\left(\frac{x}{10000^{(y-1)/N}}\right), \text{for even } y. \end{aligned} \tag{2.14}$$

2.1.5 ENSEMBLE LEARNING APPROACHES

As a popular ML approach, ensemble learning aims to enhance performance by first training multiple base learners and then combining their outputs to target a better performance than any individual learner [48]. To develop a successful ensemble learning algorithm, two key factors need to be addressed, namely training several different models (either different training inputs or training on different datasets) and effectively combining those learners. As illustrated in Fig. 2.3, ensemble learning algorithms can be mainly classified into three categories, namely boosting, bagging, and stacking [48].

1) Bagging. Bagging, or Bootstrap Aggregating, follows a series of obvious steps aimed at enhancing the accuracy of ML models. We first formulate M subsets of training data, with each subset randomly selecting N data points from the overall dataset. This allows certain data points to appear in multiple subsets. Each subset is then used to independently train a base learner. All base learners use models that share the same structure, and the training of all M learners can be performed in parallel. Once trained, each of these models makes predictions and the prediction results are then fused through methods such as max voting or averaging. The fusion improves model stability and accuracy by reducing the variance.

2) Boosting. In the first step, the M subsets are formulated in the same way as in bagging. However, boosting trains basic learners sequentially, which

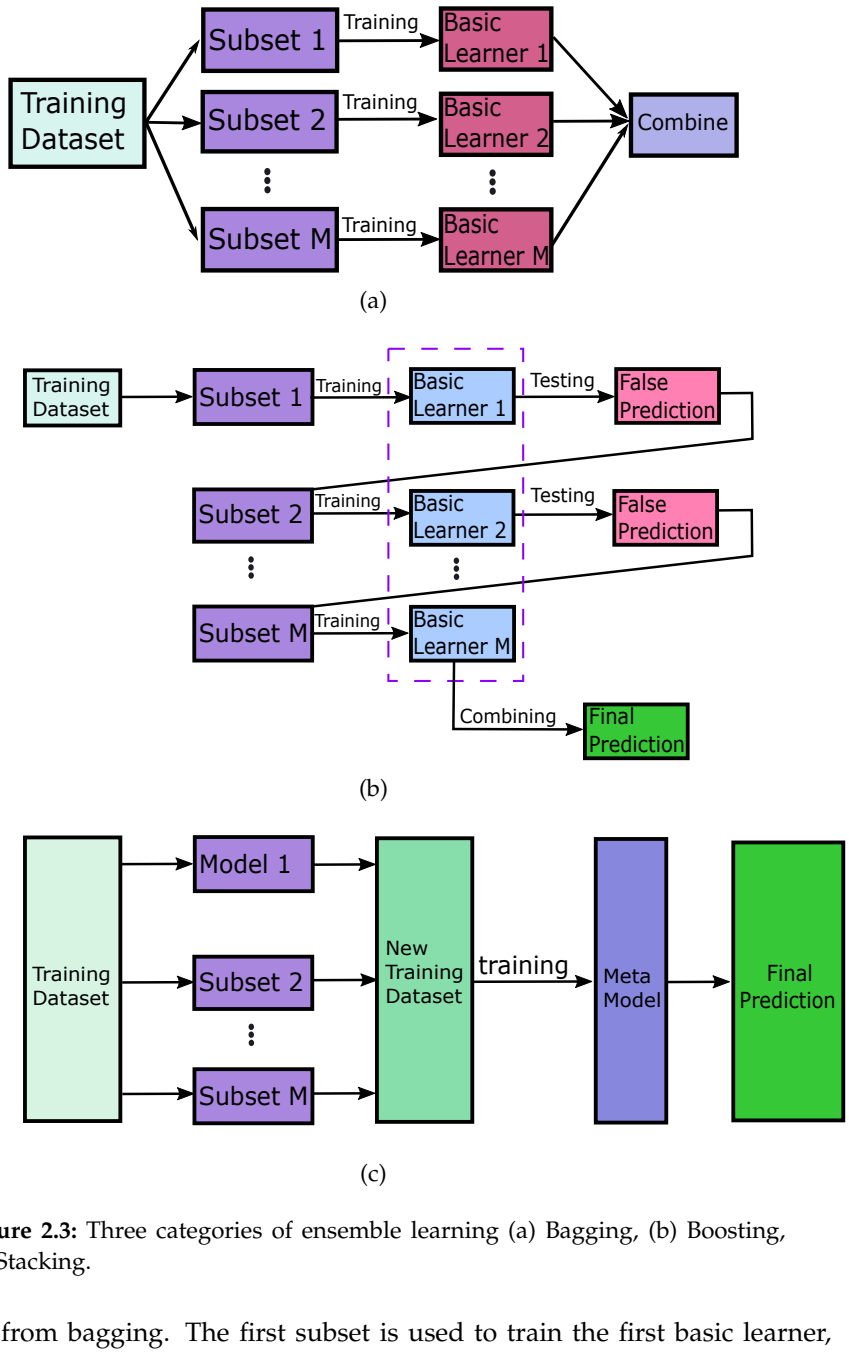


Figure 2.3: Three categories of ensemble learning (a) Bagging, (b) Boosting, (c) Stacking.

differs from bagging. The first subset is used to train the first basic learner, which is then tested on other training data. Misclassified or weakly performing data points are identified and added to the second subset. Following the

first step, the second basic learner is trained on this updated subset and tested again. We repeat this process until all the subsets M have been used. In the final step, the overall prediction can either use the output of the last learner or be adjusted by giving more weight to the performance of the most recent learner.

3) Stacking. Unlike bagging and boosting methods, stacking combines several heterogeneous models to create a meta-model. Specifically, M different models are trained using the initial dataset. The outputs of these models are then used to form a new training set, which is utilized to train a meta-model that integrates the results of the initial algorithms and produces the final prediction.

2.2 FOUNDATIONS OF MASSIVE MIMO TECHNOLOGIES

In this section, we provide a short introduction to a key technology used in modern cellular communication systems, namely massive MIMO. We begin by presenting a massive MIMO signal model, followed by illustrating the concept of beamforming. The art of massive MIMO technology enables the BS to simultaneously serve different users using the same frequency and time resource; therefore it helps to increase the spectral efficiency. This advantage becomes more pronounced with a massive number of antennas, as it increases further the Shannon channel capacity.

2.2.1 MASSIVE MIMO SIGNAL MODEL

To gain a deep comprehension of massive MIMO, it is important to understand the corresponding signal model. Here, we briefly introduce a narrowband massive MIMO signal model, readers are referred to [49] for wideband channel models. We first consider an uplink scenario, where K users occupy the same frequency resource and communicate simultaneously with the BS equipped with M antennas. We then denote $x_{k,t}$ as the transmission signal of the k-th user at time t and collect all K signals in a vector $\mathbf{x}_t \in \mathbb{C}^K$. The propagation channel between users and the BS is denoted as $\mathbf{H}_t \in \mathbb{C}^{M\times K}$. It is usually assumed that $\mathbf{H}_t$ remains static under a symbol time, therefore $\mathbf{H}_t$ can be replaced with $\mathbf{H} \in \mathbb{C}^{M\times K}$. Under this assumption, the received signal $\mathbf{y}_t \in \mathbb{C}^M$ can be written as

$$\mathbf{y}_t = \sqrt{\alpha_{up}}\mathbf{H}\mathbf{x}_t + \mathbf{n}_t, \tag{2.15}$$

where α_{up} denotes the normalization factor, which can be interpreted as the average power of uplink transmission signals of all K users at time t if we assume $\mathbb{E}_k\{|x_{k,t}|^2\} = 1$. In addition, $\mathbf{H}$ is often normalized so

that $\mathbb{E}\{|\mathbf{H}_{i,j}|^2\} = 1$. The system noise on the BS side is represented by $\mathbf{n}_t \in \mathbb{C}^M$, which is usually modeled as identical independent distributed complex Gaussian variables of zero mean and $var\{\mathbf{n}_{t,m}\} = \sigma^2$ for all indices j and all t. The BS may further process the signal by applying a beamforming matrix $\mathbf{W} = \left[\mathbf{w}_1^T, ..., \mathbf{w}_M^T\right] \in \mathbb{C}^{M \times M}$ to $\mathbf{y}_t$, which yields

$$\mathbf{z}_t = \mathbf{W}\mathbf{y}_t, \tag{2.16}$$

where $\mathbf{w}_i \in \mathbb{C}^M$ is seen as the i-th beamforming vector. $\mathbf{W}$ can be chosen depending on the specific applications. Two popular examples are the zero forcing matrix: $\mathbf{W} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H$, and the matched filter matrix: $\mathbf{W} = \mathbf{H}^H$.

For downlink transmission, reciprocity is usually assumed, which means that the downlink channel is the transpose of the uplink. Therefore, similarly to (2.15), the signal model for downlink transmission can be written as

$$\tilde{\mathbf{y}}_t = \sqrt{\alpha_{dl}}\mathbf{H}^T\widetilde{\mathbf{W}}\tilde{\mathbf{x}}_t + \tilde{\mathbf{n}}_t, \tag{2.17}$$

where α_{dl} represents the normalization factor for the downlink channel, $\tilde{\mathbf{x}}_t \in \mathbb{C}^M$ denotes the signals transmitted by the BS. The additive noise and the received signal for each UE are represented as $\tilde{\mathbf{n}}_t \in \mathbb{C}^K$ and $\tilde{\mathbf{y}}_t \in \mathbb{C}^K$, respectively. $\widetilde{\mathbf{W}} \in \mathbb{C}^{M \times M}$ represents the precoding matrix. As shown in [50], two common precoding matrices are zero forcing, and maximum ratio transmission.

One fundamental metric for the MIMO channel is the sum-rate capacity, which sets the bound for the maximum transmission rate for the whole system. Taking uplink transmission as an example, the sum-rate capacity can be written as [50]

$$C = \log_2 \det(\mathbf{I} + \frac{\alpha_{up}}{\sigma^2}\mathbf{H}^H\mathbf{H}) = \sum_{i=1}^{K}\log_2(1 + \frac{\alpha_{up}}{\sigma^2}\lambda_i), \tag{2.18}$$

where λ_i is the i-th eigenvalue of matrix $\mathbf{H}^H\mathbf{H}$. Furthermore, a significant capacity gain can be achieved if M increases excessively. Applying the weak law of large numbers, when M is sufficiently large and channels of different users are uncorrelated [50],

$$\frac{1}{M}\mathbf{h}_k^H\mathbf{h}_l \to 0, l \neq k; \quad \frac{1}{M}\mathbf{h}_k^H\mathbf{h}_k \to 1. \tag{2.19}$$

Thus, for massive MIMO systems with uncorrelated channels among different users, the capacity in (2.18) can be further approximated as $C \approx K\log_2(1 + M\frac{\alpha_{up}}{\sigma^2})$.

2.2.2 DIRECTIONAL BEAMFORMING

Another advantage of using a massive MIMO array is that it can create narrower beams, compared with MIMO system with few antennas. Thus, it can better reduce interference. To simplify the explanation of this technology, we consider the uplink of a single-user system. We assume that the propagation environment consists of a LoS path and P other reflective paths. The UE transmits a signal with wavelength λ, and the BS is equipped with a uniform linear array and the distance between the phase center of two adjacent antennas is d. Using these assumptions, the received signal $\mathbf{y}_t$ can be written as

$$\mathbf{y}_t = \gamma_{LoS}\,\mathbf{a}(\theta_{LoS}) + \sqrt{\alpha_{up}} \sum_{p=1}^{P} \gamma_p \mathbf{a}(\theta_p) + \mathbf{n}_t, \tag{2.20}$$

where γ_{LoS} is a complex scalar indicating the amplitude and phase shift of the line of sight (LoS) component, while the complex scalar γ_p represents the amplitude and phase of the p-th reflective path component. θ_p denotes the angle of arrival and $\mathbf{a}(\theta_p) \in \mathbb{C}^M$ represents the steering vector. If the separation distance between the UE and the BS is much larger than the Rayleigh distance, the steering vector $\mathbf{a}(\theta_p)$ can be approximated as $\mathbf{a}(\theta_p) = \left[1, \exp(j\frac{2\pi\, d \sin(\theta_p)}{\lambda}), \exp(j\frac{2\pi\, 2d \sin(\theta_p)}{\lambda}), ..., \exp(j\frac{2\pi\,(M-1)\, d \sin(\theta_p)}{\lambda})\right]$, see [50].

Observe that applying different beamforming vectors in (2.16) can strengthen or weaken the signal coming from certain directions. Taking the vector $\mathbf{w} = \mathbf{a}^H(\theta_p)$ as an example, it amplifies the signal with the incoming angle θ_p while suppressing the signal from angles diverging from θ_p. Consequently, at the BS, various beamforming vectors may be employed to create distinct beams, each optimized for specific signal directions. Similar principles apply on the transmitter side as well. Readers can refer to [50] for further details.

3

ML-based channel estimation

This chapter is dedicated to investigating the application of ML algorithms to channel estimation. Accurate wireless channel estimation plays a crucial role in the wireless system since it is a fundamental requirement for communication, sensing, and localization applications. The chapter is divided into two parts. The first part provides an introduction to massive MIMO RF chain calibration algorithms. The second part focuses on ML-based channel estimation.

3.1 RF CHAIN CALIBRATION

In this section, we first briefly introduce the research topic of RF chain calibration. Then, we present our signal model and the corresponding Cramér–Rao lower bound (CRLB), which can be used to set up a bound to evaluate estimators. Finally, we illustrate the closed-form moment-based estimation and the maximum likelihood estimation (MLE), for the amplitude and phase estimation of the RF chains, respectively.

3.1.1 BACKGROUND

In the literature, massive MIMO calibration approaches fall into two main categories: *reciprocity* calibration [51–55] and *absolute* calibration [56–58]. Due to the influences of transceiver RF chains, we may face challenges to leverage channel reciprocity properties in practice, which sets up a barrier for implementing systems operating in the time-division duplex mode. In particular, the aggregated responses received on the uplink encompass contributions from the UE transmitter, the channel, and the BS receiver. In contrast, the responses received on the downlink involve the UE receiver, the channel, and

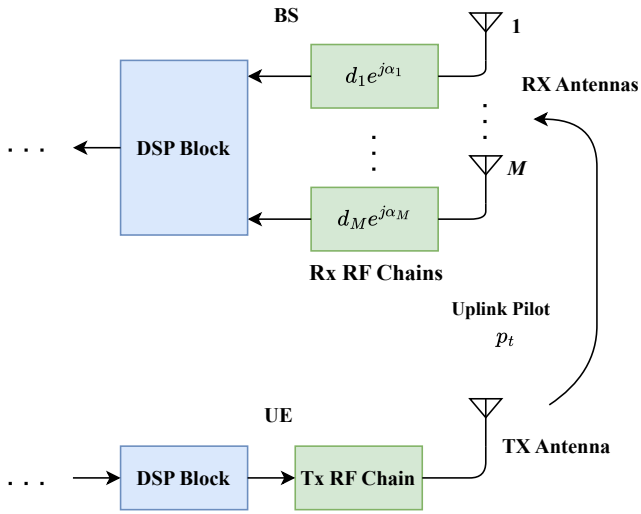


Figure 3.1: An uplink massive MIMO system with an M-antenna array and a single user.

the BS transmitter. One of the remedies for this issue is to apply reciprocity calibration algorithms to ensure that the channel is reciprocal, i.e., the same in uplink and downlink. A reciprocity calibration algorithm was first introduced in [51], and [52] presented a network protocol for UE synchronization and reciprocity-based calibration. Practical approaches for reciprocity calibration are proposed in [53–55]. Absolute calibration is necessary for AOA estimation and positioning. The idea is to estimate the receiver RF chain amplitude and phase coefficients for the entire BS array, as explained in [56–58]. The task becomes increasingly challenging as the number of antennas increases.

3.1.2 SYSTEM MODEL AND PROBLEM FORMULATION

We investigate a narrow band uplink channel model of a single-user massive MIMO system equipped with M antennas at the BS side. As illustrated in Fig. 3.1, the user transmits the pilot signal at time t to estimate the amplitudes and phases of M BS RF chains. We assume that the propagation channel is *Ricean* fading, i.e., there exists a LoS path, plus a few diffuse multipath components. The LoS path can be modeled by a steering vector $\mathbf{a}(\phi_t) \in \mathbb{C}^{M\times 1}$ with an arrival angle ϕ_t, while the sum of all diffuse multipaths can be modeled by complex Gaussian distributions. Based on the assumptions, the received signal vector $\mathbf{y}_t \in \mathbb{C}^M$ can be expressed as

$$\mathbf{y}_t = \underbrace{\gamma_t \mathbf{D}_t \mathbf{a}(\phi_t)\, p_t}_{\mathbf{s}_t} + \underbrace{\mathbf{D}_t \mathbf{h}_t\, p_t + \mathbf{n}_t}_{\boldsymbol{\omega}_t}, \tag{3.1}$$

where the scalars γ_t refers to the power of LoS component, while p_t represents the uplink pilot at time t, respectively. The power of p_t is normalized so that $|p_t|^2 = 1, t = 1, 2, \ldots, T$. We define a diagonal matrix $\mathbf{D}_t \in \mathbb{C}^{M\times M} = \text{diag}(d_1 e^{j\alpha_1}, d_2 e^{j\alpha_2}, ..., d_M e^{j\alpha_M})$, which collects all amplitudes $(d_1, ..., d_M)$ and phases $(\alpha_1, ..., \alpha_M)$ for the M RF chains. In addition, the sum of all diffuse paths at time t is modeled as a random vector $\mathbf{h}_t \in \mathbb{C}^{M\times 1}$ and $\mathbf{h}_t \sim \mathcal{CN}(0, \sigma^2)$. In other words, the mean of the diffuse components is zero and the variance (power) is σ^2 across all $t = 1, 2, \ldots, T$. The added white Gaussian noise (AWGN) at the receiver side is represented by another random vector $\mathbf{n}_t \in \mathbb{C}^{M\times 1}$, and $\mathbf{n}_t \sim \mathcal{CN}(0, N_0/2)$. We also assume that $\mathbf{n}_t$ and $\mathbf{h}_t$ are statistically independent. Taking into account the information above, we can calculate the autocorrelation $\mathbb{E}\{\boldsymbol{\omega}_t \boldsymbol{\omega}_t^H\}$ as [2]

$$\begin{aligned}\mathbb{E}\left\{\boldsymbol{\omega}_t \boldsymbol{\omega}_t^H\right\} &= \mathbb{E}\left\{(\mathbf{D}_t \mathbf{h}_t\, p_t + \mathbf{n}_t)(\mathbf{D}_t \mathbf{h}_t\, p_t + \mathbf{n}_t)^H\right\} \\ &= N_0 \mathbf{I}_M + \mathbf{D}_t\, |p_t|^2 \sigma^2 \mathbf{D}_t^H,\end{aligned} \tag{3.2}$$

where $\mathbf{I}_M \in \mathbb{C}^{M\times M}$ represents the identity matrix. Furthermore, the cross-correlation of $\boldsymbol{\omega_t}$ at different time intervals, namely $t = t_1$ and $t = t_2$, is calculated as

$$\mathbb{E}\{\boldsymbol{\omega}_{t=t_1} \boldsymbol{\omega}_{t=t_2}^H\} = \sigma^2 \mathbf{D}_{t_1} \mathbf{D}_{t_2}^H. \tag{3.3}$$

Note that the RF chain coefficients $\mathbf{D}_t$ typically remain static under a period longer than the channel coherence time. The incoming angles ϕ_t are supposed to change. However, it is assumed that the position of the UE is known by the BS during the calibration process so that the BS knows the incoming angles ϕ_t and the corresponding steering vectors $\mathbf{a}(\phi_t)$. For simplicity, from now on we omit the subscript t of ϕ_t and $\mathbf{D}_t$. Therefore, the probability distribution of the received vector $\mathbf{y}_t$ is expressed as [2]

$$\mathbf{y}_t \sim \mathcal{CN}\left(\gamma \mathbf{D}\, \mathbf{a}(\phi)\, p, N_0 \mathbf{I}_M + \mathbf{D}\, |p|^2 \sigma^2 \mathbf{D}^H\right). \tag{3.4}$$

If we observe over a time period T, the overall received signal $\mathbf{y}$ can be achieved by stacking all T vectors $\mathbf{y}_t, t = 1, 2, \ldots, T$, which yields

$$\mathbf{y} \sim \mathcal{CN}\big(\underbrace{\mathbf{1} \otimes \gamma\, p\, \mathbf{D}\mathbf{a}(\phi)}_{\boldsymbol{\mu}(\boldsymbol{\xi})},\ \underbrace{\mathbf{I}_T \otimes \mathbf{I}_M N_0 + \widetilde{\mathbf{I}}_T \otimes \sigma^2 \mathbf{D}\mathbf{D}^H}_{\mathbf{C}(\boldsymbol{\xi})}\big), \tag{3.5}$$

where $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^{T\times 1}$ and $\widetilde{\mathbf{I}}_T = \mathbf{1}.(\mathbf{1}^T) \in \mathbb{R}^{T\times T}$ is a matrix with all elements equal to 1. Furthermore, $\otimes$ represents the Kronecker product. We assume that $\mathbf{y}_t$, $\mathbf{a}(\phi)$ and p in model (3.5) are known by the BS. We

collect all the remaining parameters, which are unknown to BS, in a vector $\boldsymbol{\xi} = [d_1, d_2, \ldots, d_M, \alpha_1, ..., \alpha_M, \sigma^2, \gamma]^T$. As seen in (3.5), both the mean and the variance of the received signal $\mathbf{y}$ are related to $\boldsymbol{\xi}$, which are denoted by $\boldsymbol{\mu}(\boldsymbol{\xi})$ and $\mathbf{C}(\boldsymbol{\xi})$, respectively. It should be noted that the LoS coefficient γ should technically be a complex scalar. However, it is important to estimate the relative phase differences among RF chains, instead of the absolute phase, for calibration tasks. Considering this; the phase of γ can be absorbed by diagonal elements of $\mathbf{D}$ and thus γ can be treated as a real scalar for simplicity.

3.1.3 THE CRLB ANALYSIS

As a lower limit for variances of unbiased estimations, CRLB plays a crucial role in evaluating the performances of estimators. Suppose that all unknown parameters are collected in a vector $\boldsymbol{\theta} \in \mathbb{R}^{M'}$ and all observation data are collected in a vector $\mathbf{x} \in \mathbb{R}^N$. If the regularity condition $\mathbb{E}\left[\frac{\partial \ln p(\mathbf{x},\theta)}{\partial \theta}\right] = 0$ is satisfied, the variance of the i-th element of the vector $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{M'}$ is bounded by [59]

$$var(\hat{\boldsymbol{\theta}}_i) \geq [\mathbf{I}^{-1}(\boldsymbol{\theta})]_{ii}, \tag{3.6}$$

where $\mathbf{I}(\boldsymbol{\theta}) \in \mathbb{R}^{M' \times M'}$ represents the Fisher information matrix (FIM), with each element $\mathbf{I}(\boldsymbol{\theta})_{i,j}$ as [59]

$$\mathbf{I}(\boldsymbol{\theta})_{i,j} = -\mathbb{E}\left[\frac{\partial^2 \ln p(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_i \partial \theta_j}\right]. \tag{3.7}$$

In particular, we consider that (3.5) follows the Gaussian distribution and that both the mean and variance contain information of $\boldsymbol{\xi}$, in accordance with [59], the FIM of $\boldsymbol{\xi}$ is expressed as

$$\begin{aligned}\mathbf{I}(\boldsymbol{\xi})_{i,j} = \mathrm{Tr}\Big[\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_i}\mathbf{C}^{-1}(\boldsymbol{\xi})\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_j}\mathbf{C}^{-1}(\boldsymbol{\xi})\Big] \\ + 2\,\mathrm{Re}\Big[\frac{\partial \boldsymbol{\mu}^H(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_i}\mathbf{C}^{-1}(\boldsymbol{\xi})\frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_j}\Big].\end{aligned} \tag{3.8}$$

The CRLB also indicates the feasibility of the parameter estimation. As illustrated in (3.6) and (3.7), if a diagonal element $\mathbf{I}(\boldsymbol{\theta})_{i,i}$ of the Fisher information matrix is equal to 0, it implies that the variance of the corresponding estimated parameter becomes infinite. This phenomenon shows the impossibility of accurately estimating that particular parameter.

The sufficient and necessary condition for an estimator to achieve the CRLB is that there exists a function $g(\mathbf{x})$, which satisfies the following condition [59]:

$$\ln p(\mathbf{x}, \theta) = \mathbf{I}(\boldsymbol{\theta})\big[g(\mathbf{x}) - \boldsymbol{\theta}\big]. \tag{3.9}$$

If an unbiased estimator $\hat{\boldsymbol{\theta}} = g(\mathbf{x})$ satisfies (3.9) and reaches CRLB, it is considered as an *efficient* estimator, and the corresponding parameter estimation variances are bounded by the diagonal elements of $\mathbf{I}^{-1}(\boldsymbol{\theta})$. However, condition (3.9) is only achievable in very few special cases, and the rarity of such cases necessitates the exploration of other practical estimators in real-world applications.

3.1.4 TWO CLASSICAL ESTIMATORS

In many real-world scenarios, achieving an estimator that exactly matches (3.9) can be challenging. Considering this, it is necessary to explore other estimators that are feasible in practical cases. We therefore introduce the two most popular classical estimation algorithms that do not rely on posterior knowledge, namely, the MLE and the moment estimation.

The basic principle of MLE is to transfer an estimation task into an optimization task that maximizes the likelihood function. Suppose that all unknown parameters are collected in a vector $\boldsymbol{\theta} \in \mathbb{R}^{M'}$, the estimation $\hat{\boldsymbol{\theta}}$ is denoted as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{x}, \boldsymbol{\theta}). \tag{3.10}$$

Furthermore, if $p(\mathbf{x}, \boldsymbol{\theta})$ is differentiable, the MLE can be calculated by finding $\boldsymbol{\theta}$ to solve the following equation:

$$\frac{\partial \ln p(\mathbf{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = 0. \tag{3.11}$$

One fundamental characteristic of the MLE estimator is the asymptotic property. Suppose that if $\mathbf{x}$ satisfies the regularity condition, then $\hat{\boldsymbol{\theta}}$ asymptotically approaches the Gaussian distribution with mean $\boldsymbol{\theta}$ and covariance matrix $\mathbf{I}^{-1}(\boldsymbol{\theta})$. The variance of $\hat{\boldsymbol{\theta}}$ approaches the diagonal elements of $\mathbf{I}^{-1}(\boldsymbol{\theta})$, if large independent observation samples are achievable.

In certain applications, it can be challenging to obtain a closed-form expression for MLE estimators. One remedy to this problem is to apply numerical methods to find a solution to (3.10). Alternatively, one may explore other estimators, such as the commonly used moment estimator. It is relatively simpler to achieve a closed-form solution for a moment estimator, although the solution typically does not converge to the optimum. Specifically, given a vector $\boldsymbol{\theta} = [\theta_1, ..., \theta_{M'}]$ with unknown variables M', the first step to derive a moment estimator is to calculate the first M' order *sample* moments from the observation samples $\mathbf{x}$, and collect the calculation result in a vector $\boldsymbol{\eta} \in \mathbb{R}^{M'} = [\eta_1, ..., \eta_{M'}]$. Those sample moments are used to statistically estimate the first M' order *theoretical moments* that are associated with $\boldsymbol{\theta}$.

Specifically,

$$\begin{aligned} \eta_1 &= h_1(\theta_1, ..., \theta_{M'}) \\ &... \\ \eta_{M'} &= h_{M'}(\theta_1, ..., \theta_{M'}), \end{aligned} \tag{3.12}$$

where $h_i(\theta_1, ..., \theta_{M'})$ represents the i-th order theoretical moments. By solving equation (3.12), the moment estimator of $\boldsymbol{\theta}$ can be found.

3.2 ML-BASED CHANNEL ESTIMATION

In this section, we first provide a basic introduction to the topic of wireless channel estimation. In the next step, we start by introducing the relevant multi-carrier system model, followed by describing a popular channel estimation method, namely the Bayesian method.

3.2.1 BACKGROUND

The channel estimation quality has large influence on the performance of a wireless system, including the OFDM system. The current solution in a modern wireless system is to insert pilot signals that are known by both the transmitters and receivers into OFDM radio frames. To enhance the accuracy of channnel estimation under this framework, an intuitive method is to use more pilots or to increase transmission power. However, these methods may sacrifice the overall spectral efficiency of the system [60]. Therefore, for channel estimation, leveraging the correlation properties rather than modifying the pilot signals offers a more favorable solution.

According to the literature, relevant algorithms can be predominantly classified into two main groups: classical signal processing-based approaches and ML-based approaches. A representative of the former method is the linear minimum mean square error estimation (LMMSE), whose effectiveness depends on the availability of prior knowledge on the first- and second-order channel statistics [59]. In contrast, data-driven methods, which can effectively learn the correlation properties among subcarriers, are expected to produce promising outcomes [17, 40]. However, to our best knowledge, most of the methods listed above still face challenges in targeting robust performance in low-SNR scenarios.

3.2.2 SYSTEM MODEL AND PROBLEM FORMULATION

We hereby consider a simple single-input-single-output OFDM system that uses F subcarriers and T symbols and occupies in total FT resource elements. We consider a frequency-selective fading channel with transfer function

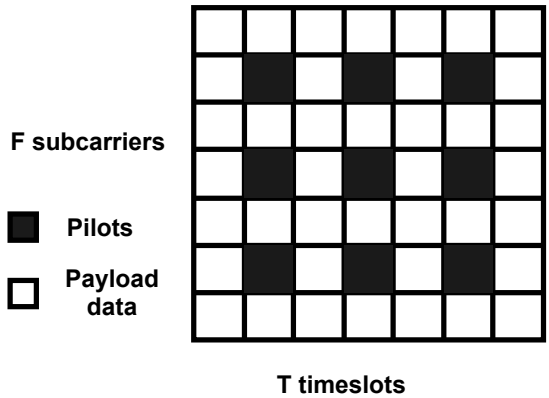


Figure 3.2: A typical OFDM time-frequency resource grid.

$H(f,t) \in \mathbb{C}$. If a signal $X(f,t) \in \mathbb{C}$ is transmitted in the t-th time slot and f-th frequency subcarrier, the received signal $Y(f,t) \in \mathbb{C}$ is expressed as

$$Y(f,t) = H(f,t)X(f,t) + N(f,t), \tag{3.13}$$

where $N(f,t) \in \mathbb{C}$ denotes the additive noise at the receiver side. As introduced before, one prerequisite for many applications in communication systems (e.g. radio-based localization, downlink precoding, etc.) is to acquire accurate channel knowledge, i.e, $H(f,t)$. To perform this task, P OFDM resource elements are allocated for pilot signals, indicated by black squares in Fig. 3.2. These pilot signals are concatenated into a vector $\hat{\mathbf{h}}_x \in \mathbb{C}^{P\times 1}$. In particular, $\hat{\mathbf{h}}_x \in \mathbb{C}^{P\times 1}$ represents a noisy version of the true pilot CSI $\mathbf{h}_x \in \mathbb{C}^{P\times 1}$, since the received pilot signals are also polluted by noise. Our task is to estimate the channel responses of all elements of the resource $\mathbf{H}_x \in \mathbb{C}^{F\times T}$, using the information available from $\hat{\mathbf{h}}_x \in \mathbb{C}^{P\times 1}$.

3.2.3 INTRODUCTION TO THE BAYESIAN METHOD

To solve the channel estimation problem, one standard approach is to leverage the LMMSE, which belongs to the Bayesian family. We first illustrate the concepts of Bayesian estimation, followed by presenting the mathematical form of LMMSE.

Unlike the classical estimation approach, the philosophy of the Bayesian method assumes that the estimation parameters are random variables instead of deterministic constants. Thus, one can effectively leverage the statistical prior knowledge of the estimation parameter to enhance the estimation performance. To illustrate this concept, we start by introducing the principle

of single-variable Bayesian estimation, since the basic concept can be readily extended to multiple-variable estimations.

We define the scalar η as the estimation error associated with the observation $\mathbf{x}$, where $\eta = \theta - \hat{\theta}$. In addition, a deterministic function $\mathcal{C}(\eta)$ is introduced as the cost function, which can be chosen based on the specific requirements of different applications. Two commonly used cost functions are as follows, namely, (a) MSE: $\mathcal{C}(\eta) = \eta^2$, (b) "Hit-or-Miss" function: $\mathcal{C}(\eta) = \text{sgn}(|\eta| - \delta)$, where sgn denotes the sign function while δ the threshold [59]. To assess the performance of the estimation, the Bayesian risk $\mathcal{R} = \mathbb{E}\{\mathcal{C}(\eta)\}$ is used as the evaluation criterion. This criterion is associated with both the observation signal $\mathbf{x}$ and θ. Using the Bayes law, $\mathcal{R}$ is expressed as

$$\begin{aligned} \mathcal{R} &= \int\int \mathcal{C}(\theta - \hat{\theta})\, p(\mathbf{x}, \theta)\, d\mathbf{x}\, d\theta \\ &= \int \left[\int \mathcal{C}(\theta - \hat{\theta})\, p(\theta|\mathbf{x})\, d\theta \right] p(\mathbf{x}) d\mathbf{x}. \end{aligned} \tag{3.14}$$

Our objective is to derive an expression for $\hat{\theta}$ that minimizes the Bayesian risk $\mathcal{R}$. This involves computing the derivative of $\mathcal{R}$ with respect to $\hat{\theta}$. For brevity, we omit the derivation details, which are available in [59], and present only the final mathematical forms of the estimators corresponding to the two cost functions mentioned above, (a) and (b). Specifically, if the MSE criterion is chosen, the optimal estimator is given by $\hat{\theta} = \mathbb{E}(\theta|\mathbf{x})$, equivalent to the mean of the posterior probability density function $p(\theta|\mathbf{x})$. On the other hand, when the "hit-or-miss" criterion is selected with a sufficiently small δ, the resulting estimator is the *maximum a posteriori* estimator: $\hat{\theta} = \max_\theta p(\theta|\mathbf{x})$.

However, it is sometimes challenging to achieve a closed-form estimator to minimize the Bayesian Mean Squared Error (BMSE) illustrated in (3.14). One solution is to resort to the framework of a linear estimator, where an estimation $\hat{\theta}$ is expressed as a linear combination of $\mathbf{x}$: $\hat{\theta} = \sum_k a_k\, x[k] + a_{k+1}$. The closed form of θ can be found by minimizing $\mathbb{E}\{(\theta - \hat{\theta})^2\}$. Specifically, θ and the corresponding BMSE are expressed as

$$\begin{aligned} \hat{\theta} &= \mathbb{E}(\theta) + \mathbf{C}_{\theta\mathbf{x}} \mathbf{C}_{\mathbf{xx}}^{-1} (\mathbf{x} - \mathbb{E}(\mathbf{x})), \\ BMSE &= \mathbf{C}_{\theta\theta} - \mathbf{C}_{\theta\mathbf{x}} \mathbf{C}_{\mathbf{xx}}^{-1} \mathbf{C}_{\mathbf{x}\theta}. \end{aligned} \tag{3.15}$$

Here, $\mathbf{C}_{\mathbf{xx}}$ and $\mathbf{C}_{\theta\theta}$ denote the autocorrelation matrix of $\mathbf{x}$ and θ, respectively, while $\mathbf{C}_{\theta\mathbf{x}}$ represents the cross-correlation matrix between θ and $\mathbf{x}$. The expression in (3.15) can easily be extended to vector estimation cases by replacing θ with a vector $\boldsymbol{\theta}$. It should be noted that the LMMSE method requires prior knowledge of the first- and second-order statistics of θ instead of the full probability density function of θ. In cases where $\mathbf{x}$ and θ are

jointly Gaussian, (3.15) fulfills the criterion of minimizing the Bayesian MSE, and therefore the LMMSE and BMSE estimators share the same form in this particular case. However, the assumption of joint Gaussian distribution is less likely to hold in practice, which may limit the performance of the LMMSE estimator.

4

ML-based wireless sensing and human activity classification

This chapter explores ML-based wireless sensing approaches. We start with an overview of the fundamentals of wireless sensing, followed by a detailed presentation of a common system model used for distributed MIMO radar. Next, we introduce a key algorithm for feature extraction and dimensionality reduction, known as the tensor decomposition algorithm. The extracted features are then used as input to the subsequent ML block.

4.1 INTRODUCTION TO WIRELESS SENSING

Wireless sensing involves detecting events in the propagation environment by analyzing the patterns of the received signals [41]. A well-known example is radar, which dates back to the early twentieth century when the German scientist Christian Hülsmeyer successfully used high-frequency radio waves to detect ships [61]. Since then, radar has played a crucial role in various aspects of daily life, including navigation and weather forecasting. Due to the significance of sensing, the integration of sensing capabilities into future cellular communication systems has been identified as a key feature for next-generation wireless technologies beyond 5G [62].

Wireless sensing can be classified into two main types: device-based sensing (active sensing) and device-free sensing (passive sensing) [62]. In device-based sensing, the object of interest is equipped with radio devices that transmit or receive radio signals. A key application of this type of sensing is radio-based localization, which will be discussed in the next chapter. In contrast, device-free sensing does not require the target object to be equipped with a radio device. Instead, the object serves as a reflector or scatterer, influencing the radio signals transmitted by other devices. Although the

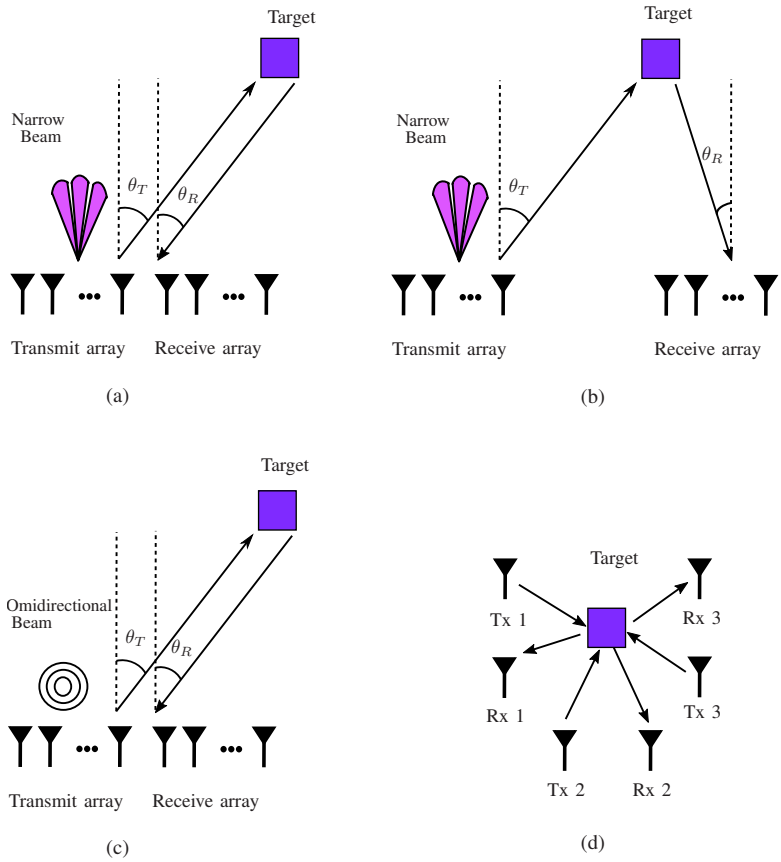


Figure 4.1: Four typical types of device-free sensing: (a) mono-static phased array radar (b) Bi-static phased array radar (c) Colocated MIMO radar (d) Distributed MIMO radar.

object lacks a radio device, it still affects the radio channel. In this chapter, we focus on device-free sensing.

Fig. 4.1 illustrates four typical use cases of device-free sensing [62]. These use cases are categorized on the basis of the type of beams used at the transmitter side and whether the transmitting and receiving arrays are located at the same place. Specifically, in a phased array, all antenna elements are synchronized and transmit the same base pulse, but the transmission signals of each antenna element can be phase-shifted or delayed to steer a narrow beam towards a desired direction. If the transmit array and the receive array are located at the same place, the device is named a mono-static phased array radar. In this case, the AoA is equal to the angle of departure. Conversely, if

the transmit and receive arrays are not at the same place, it is referred to as a bi-static phased array radar. In contrast, each element in a MIMO array may transmit different signals in a MIMO radar system, resulting in an overall beam that can be much wider or even omnidirectional. The transmit and receive antennas can be colocated or completely distributed, corresponding to colocated and distributed MIMO radar, respectively.

4.2 SYSTEM MODEL AND PROBLEM FORMULATION

We now introduce a typical system model for distributed MIMO radar sensing. We consider the uplink of a multi-user massive MIMO system, where the BS utilizes an OFDM scheme with F subcarriers. To acquire the propagation channel, multiple UEs simultaneously transmit pilot signals and these signals are received by the M-antenna BS antenna array, where each antenna is connected to an individual RF chain. We suppose that the uplink signal is continually recorded for a certain period, which results in T channel snapshots. Based on the information above, the received channel matrix $\mathbf{Y}_{f,t} \in \mathbb{C}^{M\times T}$ for each subcarrier at time index t is expressed as:

$$\mathbf{Y}_{f,t} = \mathbf{H}_{f,t} \odot \mathbf{\Gamma}_{f,t} + \mathbf{N}_{f,t}. \tag{4.1}$$

where $f \in [1, F]$ and $t \in [1, T]$ denote the subcarrier and the time index, respectively. The operation of the Hadamard product is denoted by $\odot$. The channel response of the f-th subcarrier is represented by $\mathbf{H}_{f,t} \in \mathbb{C}^{M\times T}$. Similarly, $\mathbf{\Gamma}_{f,t} \in \mathbb{C}^{M\times T}$ represents the responses of all the RF chains, while $\mathbf{N}_{f,t} \in \mathbb{C}^{M\times T}$ characterizes the additive noise. By summarizing channel responses for all F subcarriers, a third-order tensor $\mathcal{G} \in \mathbb{C}^{T\times F\times M}$ can be formulated, and this tensor contains CSI in time, frequency, and antenna domains. Suppose that in total P types of events occur between the UE and BS, our task is to classify these events according to the created tensor $\mathcal{G}$. Note that it is a non-trivial task to correctly establish a model for $\mathbf{H}_{f,t}$, because the BS lacks, in most cases, information regarding locations and the moving speed of UEs and scatters. Hence, employing an ML approach is particularly well-suited for handling this task.

4.3 ML-DRIVEN EVENT DETECTION

When we apply ML algorithms, it is important to extract adequate features used for training. Given the considerable size of our tensor $\mathcal{G}$ and its inherent rich information in the time, frequency, and antenna domains, it is essential to find an efficient approach to process $\mathcal{G}$ and to extract interesting features.

To this end, the tensor decomposition algorithm is a good solution [63]. This concept is similar to the widely used PCA approaches. Specifically, a third-order real tensor $\mathcal{H} \in \mathbb{R}^{T \times F \times M}$ can be expressed as

$$\mathcal{H} \approx \widetilde{\mathcal{H}} = \sum_{l=1}^{r_{\max}} \lambda_l \, \mathbf{x}_l \circ \mathbf{y}_l \circ \mathbf{z}_l, \tag{4.2}$$

In (4.2), the symbol $\circ$ represents the outer product of two vectors. The factor vectors are denoted as $\mathbf{x}_l \in \mathbb{R}^T$, $\mathbf{y}_l \in \mathbb{R}^F$, and $\mathbf{z}_l \in \mathbb{R}^M$, respectively. We normalize those vectors to unit length, and therefore the associated weights are absorbed into λ_l. All eigenvalues λ_l can be grouped in a vector $\boldsymbol{\lambda} \in \mathbb{R}^{r_{max}}$ that contains representative features. More details of the tensor decomposition algorithms can be found in [64]. The vector $\boldsymbol{\lambda}$ can then be fed into an NN for event detection, as shown in Fig. 2.1. As discussed in Chapter 2, the output layer of the neural network should compute the probability of each event when it addresses the classification task. The cross-entropy loss, defined in (2.8), can then be used as the cost function to measure the discrepancy between the ground truth and the predicted probabilities.

We introduce three popular metrics for evaluating classification performance: accuracy, precision, and recall. For simplicity, we use a binary classification task as an example, though the same concepts can be extended to multi-class cases. We assume that the testing dataset consists of $\mathcal{N}_{Ts}$ data samples, and $\mathcal{N}_{TP}$, $\mathcal{N}_{TN}$, $\mathcal{N}_{FP}$, $\mathcal{N}_{FN}$ represent the numbers of true positive, true negative, false positive, false negative decisions, respectively. Based on this, the metric accuracy p_{ac} is defined as

$$p_{AC} = \frac{\mathcal{N}_{TP} + \mathcal{N}_{TN}}{\mathcal{N}_{Ts}}, \tag{4.3}$$

which indicates the overall proportion of correct predictions made by the model. While accuracy provides an indication of general performance, it can be misleading, especially when the testing dataset is imbalanced with respect to positive and negative samples. In such cases, it is necessary to use another two metrics p_{PR} and p_{RE}, which are defined as

$$\begin{aligned} p_{PR} &= \frac{\mathcal{N}_{TP}}{\mathcal{N}_{TP} + \mathcal{N}_{FP}}, \\ p_{RE} &= \frac{\mathcal{N}_{TP}}{\mathcal{N}_{TP} + \mathcal{N}_{FN}}. \end{aligned} \tag{4.4}$$

We can read from (4.4) that precision measures the proportion of correct positive predictions out of all predicted positives, while recall reflects the model's ability to identify all positive cases correctly.

5

Radio-based localization and ML

This chapter explores radio-based localization systems, including both classical and ML-based localization approaches. We first introduce three traditional algorithms: ToA, AoA, and Time Difference of Arrival (TDoA) [65], which serve as foundational methods in wireless localization. Next, we demonstrate how ML can be applied to solve localization tasks by using a massive MIMO system as a representative example. Finally, we discuss various uncertainty estimation techniques, underscoring their importance in enhancing the precision and robustness of wireless localization systems.

5.1 CLASSICAL RADIO LOCALIZATION ALGORITHMS

The basic concepts of ToA, AoA, and TDoA are presented in Fig. 5.2. The first two algorithms belong to the range-based method, while the last belongs to the angle-based method. To implement those three methods, it is usually required to have several anchor nodes (e.g. BSs) with known positions. These anchor nodes can either receive the uplink signals from the UE or transmit the downlink signals to the UE [65]. We use the uplink as an example, while similar concepts can be applied to downlink wireless localization.

- ToA method: The mechanism of the ToA method is given in Fig. 5.2 (a). As shown, three synchronized BSs receive uplink positioning reference signals (PRS) sent by UE simultaneously and measure their signal arrival times to calculate the corresponding distances l_1, l_2, l_3. After this, three circles are established, where the center of the i-th circle is the axis of the i-th BS and the corresponding radius is l_i. After this, the position of the UE can be calculated by computing the intersection points of those three circles. Such method does not require that the

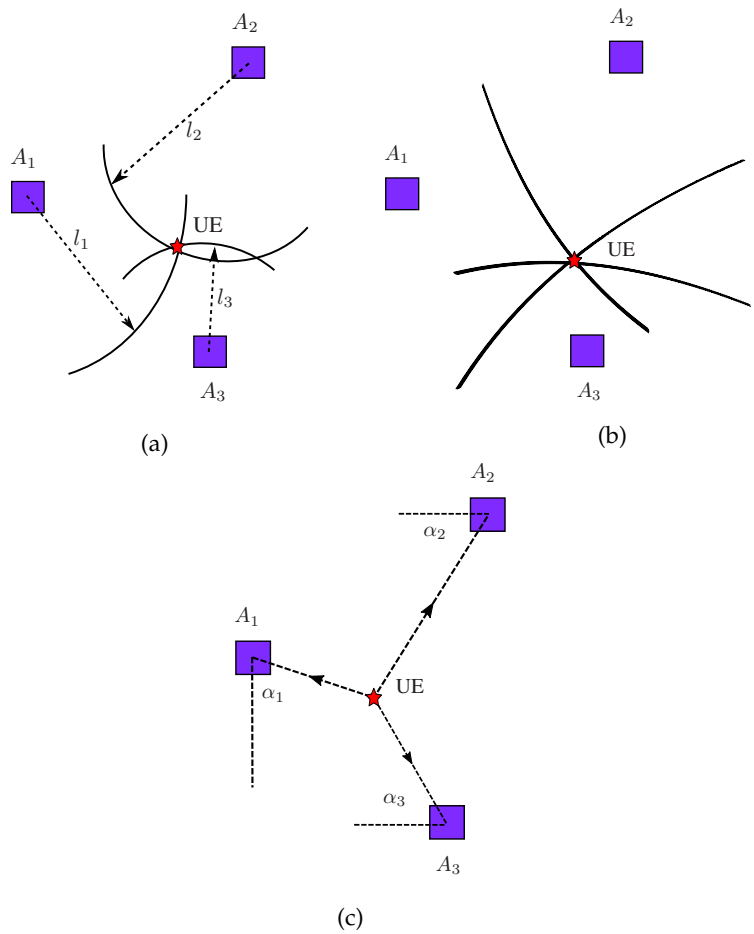


Figure 5.2: Three classical radio-based localization algorithms (a) ToA, (b) TDoA, (c) AoA.

BS is equipped with an antenna array. However, it is required that all BSs need to be synchronized, and the synchronization error has strong effect on the position accuracy [65].

- TDoA method: As illustrated in Fig. 5.2 (b), TDoA method measures the time arrival differences between every two BSs. After achieving TDoA parameters, the BS can create several hyperbolas and compute the UE position by calculating the axis of the intersection points of those hyperbolas. This method is widely applied in many commercial systems, such as the narrowband internet of things (NB-IoT), since it eases the requirement of synchronization between UE and the BS [1].

- AoA method: The AoA localization method determines the UE position by calculating the angle at which uplink signals arrive at the BS, see Fig. 5.2 (c). To estimate this arrival angle, the BS is usually required to be equipped with an antenna array. Common algorithms for AoA estimation include multiple signal classification (MUSIC) and space-alternating generalized expectation-maximization (SAGE). By using these estimated angles to draw lines representing the directions of the incoming signals, the location of the UE can be identified at the conjunction point where the lines from multiple BSs meet.

We then emphasize three key properties of the PRS, namely, autocorrelation, cross-correlation, and peak-to-average power ratio (PAPR) [66].

- Autocorrelation: This property measures how a signal correlates with a delayed version of itself. For a discrete signal $x(n)$, the auto-correlation function $R_{xx}(\tau)$ is expressed as

$$R_{xx}(\tau) = \mathbb{E}\{x(n)\,x^*(n+\tau)\}, \tag{5.1}$$

  where τ represents the delay parameter. This property is crucial for PRS since a good autocorrelation property facilitates effective signal identification and synchronization, leading to precise arrival time and position estimations.
- Cross-correlation: This property assesses the similarity between two different signals $x_1(n)$ and $x_2(n)$. The correlation function $R_{x_1,x_2}(\tau)$ is written as

$$R_{x_1,x_2}(\tau) = \mathbb{E}\{x_1(n)\,x_2^*(n+\tau)\}. \tag{5.2}$$

  It is desirable to have low cross-correlation between different signals to minimize interference from other UE.
- PAPR: This metric measures the ratio of the peak power of a signal to its average power. For a signal sequence $x_3(n)$, its PAPR is expressed as

$$PAPR = \frac{\max\{|x_3(n)|^2\}}{\mathbb{E}\{|x_3(n)|^2\}}. \tag{5.3}$$

  A low PAPR in PRS generally suggests a more uniform power distribution, which enhances the efficiency of the power amplifier. This is particularly crucial for cost-effective NB-IoT hardware.

5.2 ML-BASED LOCALIZATION IN MASSIVE MIMO SYSTEMS

This section discusses the application of ML algorithms to massive MIMO systems for localization tasks. We begin by presenting two common system

models in the antenna and beam space domains, respectively. Next, we introduce various fingerprints that can be used to train NNs for localization. Following this, we present an approach to determine training density using the Nyquist theorem. Finally, we discuss different uncertainty estimation algorithms that are crucial for reliable localization.

5.2.1 SYSTEM MODEL

Here, we introduce two common systems models in ML-based localizations. The first model presents the received uplink UE signal in the antenna space domain, while the second model operates in the beam domain.

1) **Antenna space domain**
We consider the uplink of a single-user M-antenna massive MIMO system using OFDM schemes with F subcarriers. Each antenna on the BS side is linked to an RF front-end and subsequent digital processing chain, which enables simultaneous processing of signals received from all antennas. When the UE moves to position $\mathbf{p}_i \in \mathbb{R}^2$, the received channel transfer function matrix $\mathbf{Y}_{\mathbf{p}_i} \in \mathbb{C}^{M\times F} = [\mathbf{y}_{\mathbf{p}_i,1}, ..., \mathbf{y}_{\mathbf{p}_i,F}]$ is expressed as

$$\mathbf{Y}_{\mathbf{p}_i} = \mathbf{H}_{\mathbf{p}_i} \odot \mathbf{\Gamma} + \mathbf{N}, \tag{5.4}$$

where the uplink channel is denoted by $\mathbf{H}_{\mathbf{p}_i} \in \mathbb{C}^{M\times F}$, while the complex coefficients of all M RF chains are represented by $\mathbf{\Gamma} \in \mathbb{C}^{M\times F}$. The additive noise matrix for all M RF chains is denoted by $\mathbf{N} \in \mathbb{C}^{M\times F}$. In addition, the symbol $\odot$ denotes the Hadamard product. As the UE moves, we record $\mathcal{T}$ snapshots and collect $\mathcal{T}$ matrices $\mathbf{Y}_{\mathbf{p}_i}$. Our objective is to calculate $\mathbf{p}_i$ based on the received $\mathbf{Y}_{\mathbf{p}_i}$.

2) **Beam space domain**
We consider the uplink of a commercial single-user 5G new radio (NR) massive MIMO system, where the UE transmits time series of sounding reference signals (SRS). These SRSs are received by the BS, which has M_{BS} antennas. Half of the antennas are vertically polarized, formulating N_{V} beams, and the other half is horizontally polarized, formulating N_{H} beams. The system employs the OFDM scheme with F subcarriers. Suppose there exist $\mathcal{P}$ multipath components in the channel, where $\tau_{p,t}$ represents the time delay of the p-th path at the time index t and $\alpha_{p,m,t}$ denotes the complex coefficient of the multipath component. The azimuth and elevation angles to the arriving p-th path are given by ϕ_p and θ_p, respectively. For the m-th UE, the propagation channel model for the i-th beam at time index t and the f-th

subcarrier can be formulated as

$$\begin{aligned} h_{\mathrm{V},i,m,t}(f) &= \sum_{p=1}^{\mathcal{P}} \beta_{\mathrm{V},i}(\phi_p, \theta_p, f)\, \alpha_{p,m,t}\, \exp\{-j2\pi f\, \tau_{p,t}\} \\ h_{\mathrm{H},i,m,t}(f) &= \sum_{p=1}^{\mathcal{P}} \beta_{\mathrm{H},i}(\phi_p, \theta_p, f)\, \alpha_{p,m,t}\, \exp\{-j2\pi f\, \tau_{p,t}\}, \end{aligned} \tag{5.5}$$

where $\beta_{\mathrm{V},i}(\phi_p, \theta_p, f)$ and $\beta_{\mathrm{H},i}(\phi_p, \theta_p, f)$ denote the beam responses formulated by vertical and horizontal polarized antennas, respectively.

5.2.2 FINGERPRINT GENERATION

It is important to select appropriate channel fingerprints for ML-based localization tasks. These fingerprints can be extracted from the channel transfer function. Examples of fingerprints are received signal strength, ToA, spatial covariance matrix, and the impulse-response beam matrices, etc. Here, we briefly introduce the spatial covariance matrix and the impulse-response beam matrices, since both of them can be used for MIMO localization tasks [6, 29].

1) **The spatial covariance matrix**
The covariance matrix characterizes the second-order statistical properties of the propagation channel, including both the cross-correlation between signals received by different antennas and the autocorrelation of signals of individual antennas in an antenna array. Specifically, the main diagonal entries of this matrix reflect the signal power received by each antenna, while the off-diagonal entries represent the cross-correlation between antennas [5]. Unlike other fingerprints, such as AoA, it is not necessary to have a fully-calibrated array to generate this fingerprint. The covariance matrix is defined as $\mathbf{C}_i = \mathbb{E}\{\mathbf{y}_{p_i}\mathbf{y}_{p_i}^H\} \in \mathbb{C}^{M\times M}$, where the expectation is over uncorrelated interference, noise and small-scale fadings. However, since it is impossible to collect an unlimited number of channel samples, we can only estimate the covariance matrix $\mathbf{C}_i \in \mathbb{C}^{M\times M}$ rather than achieve its exact form. We assume that a total of $\mathcal{N}_{p_i}$ channel samples are collected in the neighborhood region of $\mathbf{p}_i$. Based on this assumption, $\mathbf{C}_i$ can be estimated by the sample covariance matrix $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} \in \mathbb{C}^{M\times M}$, which is expressed as

$$\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} = \frac{1}{\mathcal{N}_{p_i}} \sum_{j=1}^{\mathcal{N}_{p_i}} \mathbf{Y}_{p_j}\mathbf{Y}_{p_j}^H. \tag{5.6}$$

2) **Impulse-response beam matrix**
This fingerprint visualizes the channel state information in the angular-delay

domain, which is shown to be beneficial to ML-based localization tasks [25]. At first, we apply a specific window (e.g. the Hann window) to the captured channel transfer functions in (5.5), to obtain $\tilde{h}_{V,i,m,t}(f)$ and $\tilde{h}_{H,i,m,t}(f)$. Such an operation can reduce the side-lobe effect. Suppose that the number of subcarriers is F, the F-length Hann window is expressed as [67]

$$w[f] = \sin^2\left(\frac{\pi f}{F}\right), \quad f = 0, \ldots, F-1. \tag{5.7}$$

After the windowing operation, the inverse-Fourier transform is applied to the windowed-channel transfer function $\tilde{h}_{V,i,m,t}(f)$ and $\tilde{h}_{H,i,m,t}(f)$ to achieve impulse responses $\tilde{h}_{V,i,m,t}(\tau)$ and $\tilde{h}_{H,i,m,t}(\tau)$. We can then define the impulse-response beam matrices $\mathbf{H}_{m,t} \in \mathbb{C}^{(N_V+N_H)\times F}$ by collecting impulse responses for all beams.

5.2.3 NYQUIST ANALYSIS

Determining the necessary training density is important when designing an ML-based localization pipeline. This problem can be solved by finding the maximum separation distances between two adjacent training samples. To this end, one can apply the Nyquist sampling theorem in the spatial domain, which states that spatial aliasing occurs if the sampling rate is insufficient. We consider uniform sampling along one dimension for simplicity (see Fig. 5.3) and all available samples are evenly distributed with a separation distance δ_d, while similar methods can be extended to 2-D cases. We denote a vector

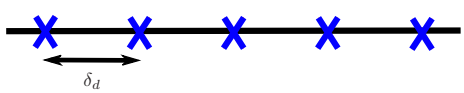


Figure 5.3: Uniform sampling along one dimension.

$\mathbf{v} \in \mathbb{R}^{\tilde{M}}$ as the fingerprint used for localization and $\mathbf{v}$ varies when the UE moves to Q different positions. We denote the channel fingerprint at the j -th position as $\mathbf{v}_j$ and define a matrix $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_Q] \in \mathbb{R}^{\tilde{M}\times Q}$ to collect all fingerprint vectors. In the next step, the discrete one-dimensional Fourier transform is applied to each row of $\mathbf{V}$, to achieve the spatial spectrum of the fingerprint vector. The *fftshift* operation is then applied along the horizontal axis of $\mathbf{V}$, so that the achieved matrix $\tilde{\mathbf{V}} \in \mathbb{R}^{\tilde{M}\times Q}$ has a direct current component in the middle column. Then a spectrum window L is defined that selects L consecutive columns of $\tilde{\mathbf{V}}$ representing the lower frequency components. These L columns are used to form a new matrix $\tilde{\mathbf{V}}^L \in \mathbb{R}^{\tilde{M}\times L}$. After selecting L, we compute the corresponding separation distance between

two adjacent training samples Δ_d as: $\Delta_d = \frac{L}{Q}\delta_d$. We define an aliasing indicator η as the ratio between the Euclidean norm of $\tilde{\mathbf{V}}^L$ and $\tilde{\mathbf{V}}$, that is, $\eta = \frac{||\tilde{\mathbf{V}}^L||^2}{||\tilde{\mathbf{V}}||^2}$. If η approaches 1, it indicates that the system suffers less from spatial aliasing.

5.3 UNCERTAINTY ESTIMATION

Despite the impressive results achieved by deep NNs in various areas, their disadvantages, such as the lack of interpretability, can hinder their deployment in real-world applications. Especially when DL methods are used for life-critical localization tasks such as autonomous driving, it is crucial not only to estimate the vehicle's position, but also to know the reliability of this estimated position. Significant research has been conducted to evaluate and improve the quality of uncertainty estimation for deep neural networks [68].

5.3.1 TWO TYPES OF UNCERTAINTY

We introduce here two types of uncertainty that we often encounter when designing an ML algorithm: Aleatoric and epistemic uncertainty [68]. Aleatoric uncertainty, also known as statistical uncertainty, is derived from inherent randomness that cannot be reduced by designing a better ML algorithm or collecting more data. An example of this is the outcome when throwing a dice. In contrast, epistemic uncertainty, also known as model uncertainty, arises from a lack of knowledge about the accurate model parameters of the system being modeled and the underlying data distribution of the input data. For example, an autonomous driving model may be less confident when encountering new scenarios that did not present in the training dataset. Unlike aleatoric uncertainty, epistemic uncertainty can be reduced by improving the algorithm or by collecting more high-quality training data.

5.3.2 UNCERTAINTY ESTIMATION IN REGRESSION MODELS

Many regression models based on NNs include uncertainty estimates in their frameworks [68–70]. In this section, we introduce two representatives: the Gaussian negative log-likelihood (NLL) [69], and regression-by-classification (RbC) methods [69]. NLL assumes that the output of the NN follows a Gaussian distribution, while RbC can model outputs that are not necessarily Gaussian. To implement these methods, one only needs to modify the output layer and the cost functions of the NN in Fig. 2.1, rather than changing the entire NN structure. This makes it straightforward to integrate these approaches into existing NNs. In the following text, we denote Ω_1 as the

entire training data set with $\mathcal{N}_{tr}$ training samples. For simplicity, we assume that the output is a scalar, with y_i as the groundtruth of the i-th training sample. The concept can be easily extended to multidimensional cases.

Gaussian NLL method
For simplicity, we assume that the output is a scalar, with y_i representing the ground truth of the i-th sample. The NLL method assumes that the output of the NN follows a Gaussian distribution [69], determined by both the estimated value $\hat{y}_i$ and the error variance $\hat{\sigma}_i^2$ of the network. By considering all $\mathcal{N}_{tr}$ training samples, we can use the following negative log-likelihood Ψ_{NLL} as the cost function, which is expressed as

$$\Psi_{NLL} = \frac{1}{\mathcal{N}_{tr}} \sum_{i \in \Omega_{tr}} \left(\frac{\ln \hat{\sigma}_i^2}{2} + \frac{(y_i - \hat{y}_i)^2}{2\hat{\sigma}_i^2}\right). \tag{5.8}$$

Note that in (5.8), both the accuracy of the estimate (indicated by $(y_i - \hat{y}_i)^2$) and the confidence level (indicated by $\hat{\sigma}_i^2$) influence Ψ_{NLL}. A smaller Ψ_{NLL} indicates better uncertainty estimation. A much larger $\hat{\sigma}^2$ (underconfident) increases the first term, while a smaller $\hat{\sigma_i}^2$ (overconfident) increases the second term.

Regression-by-classification
The Gaussian NLL method may face challenges if the probability density function of the error deviates largely from a Gaussian distribution. To address this issue, a family of RbC algorithms can be employed [70]. The objective of the RbC algorithm is to predict the probability density function of the parameter(s) of interest. For simplicity, we assume the NN outputs a one-dimensional discretized probability distribution.

The first step in applying RbC is to define the range of the parameter of interest, with lower and upper bounds as y_{lb} and y_{ub}, respectively. This range is divided into L discrete bins. We further denote $\bar{l}_{y,k}$ as its midpoint of the k-th discrete bin. Suppose that we have $\mathcal{N}_{tr}$ training samples, for the i-th training sample, our network needs to compute the probability of each bin $q_{k,i}$ using a *softmax* activation function, see (2.12), at the final layer of the network. This ensures that the output of the network forms a valid probability distribution. To account for potential quantization errors, an optional bias term $b_{k,i}$ can be added for each bin. We define $\mathbf{b}_i \in \mathbb{R}^L$ to collect all the bias terms. To train the network, the following η-norm cost function can be considered:

$$\Psi_{\text{Rbc}} = \frac{1}{N_{tr}} \sum_{i \in \Omega'_{tr}} \Big(|| \sum_{k=1}^{L} q_{k,i} \bar{l}_{y,k} - y_i + b_{k,i} ||^{\eta} + \gamma_1 ||\mathbf{b}_i|| \Big),$$

where two common choices of η are $\eta = 1$ (*Taxicab* norm) or $\eta = 2$ (*Euclidean* norm) [22]. In contrast to the NLL method, RbC does not assume a Gaussian distribution for the output probability, which extends the applicability of this algorithm.

5.3.3 CRITERIA TO EVALUATE ESTIMATED UNCERTAINTY

Evaluating the quality of uncertainty estimation is crucial, though it is not as straightforward as assessing regression or classification results. We hereby introduce two metrics to evaluate uncertainty estimation algorithms, namely NLL and area under the sparsification error curve (AUSE) [68]. The NLL is a widely used metric because it quantifies how well the predicted probability distribution aligns with the true distribution of the data (not necessarily Gaussian). On the other hand, AUSE evaluates whether the predicted error aligns with the level of uncertainty. The ideal case is when the predicted error is small, the uncertainty is also small, and vice versa.

NLL
We denote $S_{\text{ts}} = \{(\mathbf{x}_i, y_i) | i = 1, ..., N_{ts}\}$ as the testing dataset, and $\tilde{p}(y|\mathbf{x}, \theta)$ the probability density function learned by the NN with the hyperparameter θ. This metric examines whether the data distribution of the testing dataset matches the learned density function by calculating the following negative log-likelihood:

$$NLL(S_{\text{ts}}) = -\sum_{i=1}^{N_{\text{ts}}} \ln \tilde{p}(y_i|\mathbf{x}_i, \theta). \tag{5.9}$$

A smaller NLL means that the learned model better predicts the data distribution.

AUSE
To calculate AUSE, the predicted values and their uncertainty measures for all testing samples are first obtained from the model. One representative uncertainty measure is the entropy of the predicted probability density function. Next, we calculate the predicted MSE and sort the prediction errors and uncertainty measures in ascending order. We then sequentially remove the least confident predictions, starting with a percentage φ that increases from 0% to 100%. After each removal, the prediction MSE is recalculated and plotted against the removal percentage φ, creating the sparsification error curve $s(\varphi)$. Next, a similar process is applied to the uncertainty measures to obtain the oracle curve $g(\varphi)$. We then normalize $g(\varphi)$ by multiplying by a scalar to get $g'(\varphi)$, so that the maximum values of $s(\varphi)$ and $g'(\varphi)$ are the

same. The AUSE is the area between $s(\varphi)$ and $g'(\varphi)$, which is

$$\text{AUSE} = \int_0^1 |s(\varphi) - g'(\varphi)| \, d\varphi. \tag{5.10}$$

A lower AUSE value indicates better uncertainty estimates, which indicates that the predicted error and uncertainty coincide well with each other. An example can be seen in the attached paper VI.

6

Conclusions and Future work

This chapter concludes the thesis and suggests two potential directions for future research.

6.1 KEY CONCLUSIONS AND TAKEAWAYS

This thesis presents various applications of machine learning (ML) in wireless communication systems, such as channel estimation, radio-based sensing, and localization tasks. The results show that ML is a powerful tool to enhance the performance of wireless systems, such as improving the accuracy of the localization, which can be a good complement to existing traditional methods. Furthermore, it is also important to point out that applying some necessary pre-processing steps to generate representative learning features (such as fingerprints) for learning algorithms is still important. Thus, end-to-end machine learning is not the best solution to everything when it comes to a wireless system. Instead, one should integrate domain knowledge and feature engineering to better tailor the learning algorithms to the specific needs of wireless communication systems. We list our main conclusions as follows.

We demonstrate the strengths of ML-based algorithms in terms of performing channel estimation tasks in OFDM systems. We investigated a channel estimation pipeline based on a pure FCNN and evaluated its performance using simulations. The results show that by adequately using a simple FCNN and only a few pilots, the quality of channel estimation can be significantly improved and more robust in low signal-to-noise-ratio regions, compared to traditional methods, such as the LMMSE estimator. This indicates that ML algorithms can be integrated into the current OFDM channel estimation

framework to further increase the performance of the system and save transmission energy.

ML technology is a suitable tool for performing device-free wireless sensing tasks, and the sensing performance can be largely improved if the system is equipped with an excessive number of antennas. In this thesis, we explore a pipeline for human activity recognition using a massive MIMO array. To accomplish this goal, it is necessary to apply relevant algorithms to extract key features in delay, Doppler, and antenna domains. An example of the algorithm is the tensor decomposition algorithm. Training on the extracted features with a simple FCNN network can help our pipeline to achieve very good classification accuracy. This demonstrates the potential of ML in sensing tasks.

When applying ML-based algorithms to perform localization tasks, the choice of training algorithm and fingerprint generation are both important. Examples of the fingerprints are the impulse-response beam matrix, covariance matrix, truncated channel impulse responses, etc. After selecting different fingerprints, the necessary training density can be determined with the aid of the Nyquist sampling theorem, which may vary based on the fingerprint selected. Advanced algorithms can be applied to increase localization performance, such as the attention mechanism or the ensemble learning approaches. The attention mechanism allows the framework to focus on the most relevant features in the input data, improving its ability to prioritize critical information, while ensemble learning algorithms combine different channel fingerprints to make more robust predictions.

Accurate uncertainty estimation in localization algorithms is crucial, particularly in life-critical applications such as autonomous drive. This thesis explores two algorithms for estimating uncertainty: the Gaussian NLL and RbC methods. While Gaussian NLL assumes that localization errors follow a normal distribution, RbC does not impose this requirement. We evaluate these uncertainty estimation algorithms using real BS data from a 5G commercial measurement campaign. Results show that RbC and NLL perform similarly under low uncertainty conditions, which typically occurs in high SNR conditions or with large training datasets. However, under high uncertainty conditions, where localization errors deviate significantly from a normal distribution, RbC achieves better localization accuracy than NLL.

6.2 FUTURE PERSPECTIVES

We discuss two key directions for future research in AI-based vehicular navigation technology: enhancing robustness and increasing explainability. These goals align with the requirements outlined in the latest European Union

Artificial Intelligence Regulations [71] *. Our discussion is focused specifically on the technical aspects.

- **Robustness**. AI-based navigation systems must improve their robustness to ensure consistent performance across diverse scenarios. In other words, it is unacceptable for these systems to perform well in one situation while performing poorly in another, especially in life-critical contexts. An intuitive solution is to enlarge the training dataset, enabling the system to encounter a broader range of scenarios. However, since creating a training dataset that covers all possible scenarios is challenging, applying transfer learning algorithms becomes essential. From my perspective, the goal of transfer learning is not only to identify the statistical differences between datasets but also to learn and adapt the underlying knowledge. This approach allows the AI model to leverage insights from one domain and apply them effectively to new, unseen scenarios, which is especially valuable in wireless navigation fields. While some preliminary research has been conducted [72], there remain significant potential for further advancements.
- **Exaplainable AI**. It is crucial to increase the explainability of AI-powered navigation systems, particularly in safety-critical fields. In my opinion, while neural network-based methods often achieve excellent classification or regression accuracy, the lack of transparency may limit their application in such areas. Therefore, relying solely on neural networks may not be ideal due to their inherent complexity in interpreting its decision. One promising approach is the use of knowledge distillation [73], where a strong neural network-based model can be used to supervise the training of other models with better explainability, such as decision trees. However, additional research is essential to ensure that alternative models achieve high performance comparable to neural networks while also being robust and reliable for high-risk applications.

*The European Union Artificial Intelligence Act came into effect on August 1, 2024.

Bibliography

[1] G. Tian, H. OdetAlla, B. E. Priyanto, S. Hu, and F. Tufvesson, "Modified Gold Sequence for Positioning Enhancement in NB-IoT," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[2] G. Tian, H. Tataria, and F. Tufvesson, "Amplitude and Phase Estimation for Absolute Calibration of Massive MIMO Front-Ends," in *2020 IEEE International Conference on Communications (ICC)*, 2020.

[3] G. Tian, X. Cai, T. Zhou, W. Wang, and F. Tufvesson, "Deep-Learning Based Channel Estimation for OFDM Wireless Communications," in *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, 2022.

[4] B. R. Manoj, G. Tian, S. Willhammar, F. Tufvesson, and E. G. Larsson, "Sensing and Classification Using Massive MIMO: A Tensor Decomposition-Based Approach," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2649–2653, 2021.

[5] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and F. Tufvesson, "Deep-Learning-Based High-Precision Localization With Massive MIMO," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 19–33, 2024.

[6] G. Tian, D. Pjanić, X. Cai, B. Bernhardsson, and F. Tufvesson, "Attention-aided Outdoor Localization in Commercial 5G NR Systems," 2024. [Online]. Available: https://arxiv.org/abs/2405.09715

[7] M. Sellathurai and S. Haykin, "Statistical Learning and Layered Space-time Architecture for Point-to-point Wireless Communications," in *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers*, vol. 2, 1998, pp. 1084–1088.

[8] H. Kaiming, X. Zhang, R. Shaoqing, and S. Jian, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-End Speech Recognition: A Survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2024.

[10] 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.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All you Need," in *2017 Advances in Neural Information Processing Systems (NEURIPS)*, 2017.

[12] Z. Zhu and H. Zhao, "A Survey of Deep RL and IL for Autonomous Driving Policy Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 043–14 065, 2022.

[13] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep Learning-Based Vehicle Behavior Prediction for Autonomous Driving Applications: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 33–47, 2022.

[14] N. Khan, S. Coleri, A. Abdallah, A. Celik, and A. M. Eltawil, "Explainable and Robust Artificial Intelligence for Trustworthy Resource Management in 6G Networks," *IEEE Communications Magazine*, vol. 62, no. 4, pp. 50–56, 2024.

[15] X. Tong, Z. Zhang, and Z. Yang, "Multi-View Sensing for Wireless Communications: Architectures, Designs, and Opportunities," *IEEE Communications Magazine*, vol. 61, no. 5, pp. 40–46, 2023.

[16] H. Lee, H. K. Kim, S. H. Oh, and S. H. Lee, "Machine Learning-Aided Cooperative Localization under A Dense Urban Environment: Demonstrates Universal Feasibility," *IEEE Vehicular Technology Magazine*, vol. 19, no. 3, pp. 78–89, 2024.

[17] H. Ye, G. Y. Li, and B.-H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[18] A. V. Oppenheim, "Algorithm Kings: The Birth of Digital Signal Processing," *IEEE Solid-State Circuits Magazine*, vol. 4, no. 2, pp. 34–37, 2012.

[19] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965.

[20] G. Frantz, "Signal Core: A Short History of the Digital Signal Processor," *IEEE Solid-State Circuits Magazine*, vol. 4, no. 2, pp. 16–20, 2012.

[21] A. L. Fradkov, "Early History of Machine Learning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 1385–1390, 2020, 21st IFAC World Congress.

[22] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[23] S. Kay, *Fundamentals of Statistical Signal Processing: Detection theory*. Prentice-Hall PTR, 1998.

[24] K. Xu, M. Qin, F. Sun, Y. Wang, Y. Chen, and F. Ren, "Learning in the Frequency Domain," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] R. Whiton, J. Chen, and F. Tufvesson, "Wiometrics: Comparative Performance of Artificial Neural Networks for Wireless Navigation," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 9, pp. 13 883–13 897, 2024.

[26] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty Years of Machine Learning: The Road to Pareto-Optimal Wireless Networks," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 1472–1514, 2020.

[27] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, "Visualizing High-Dimensional Data: Advances in the Past Decade," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 3, pp. 1249–1268, 2017.

[28] M. Espadoto, R. M. Martins, A. Kerren, N. S. T. Hirata, and A. C. Telea, "Toward a Quantitative Survey of Dimension Reduction Techniques," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2153–2173, 2021.

[29] C. Studer, S. Medjkouh, E. Gonultaş, T. Goldstein, and O. Tirkkonen, "Channel Charting: Locating Users within the Radio Environment Using Channel State Information," *IEEE Access*, vol. 6, pp. 47 682–47 698, 2018.

[30] P. Ferrand, M. Guillaud, C. Studer, and O. Tirkkonen, "Wireless Channel Charting: Theory, Practice, and Applications," *IEEE Communications Magazine*, vol. 61, no. 6, pp. 124–130, 2023.

[31] L. Busoniu, R. Babuska, and B. De Schutter, "A Comprehensive Survey of Multiagent Reinforcement Learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[32] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep Reinforcement Learning: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, 2024.

[33] D. Silver, A. Huang *et al.*, "Mastering the Game of Go with Deep Neural Networks and Tree Search." *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[34] O. Vinyals, I. Babuschkin *et al.*, "Grandmaster Level in StarCraft II using Multi-agent Reinforcement Learning." *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[35] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, "Reinforcement Learning for Mobile Robotics Exploration: A Survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3796–3810, 2023.

[36] Y. S. Nasir and D. Guo, "Multi-Agent Deep Reinforcement Learning for Dynamic Power Allocation in Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[37] Y. Li, Y. Xu, Y. Xu, X. Liu, X. Wang, W. Li, and A. Anpalagan, "Dynamic Spectrum Anti-Jamming in Broadband Communications: A Hierarchical Deep Reinforcement Learning Approach," *IEEE Wireless Communications Letters*, vol. 9, no. 10, pp. 1616–1619, 2020.

[38] C.-X. Wang, X. You *et al.*, "On the Road to 6G: Visions, Requirements, Key Technologies, and Testbeds," *IEEE Communications Surveys Tutorials*, vol. 25, no. 2, pp. 905–974, 2023.

[39] H. Huang, S. Guo, G. Gui, Z. Yang, J. Zhang, H. Sari, and F. Adachi, "Deep Learning for Physical-Layer 5G Wireless Techniques: Opportunities, Challenges and Solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.

[40] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers," *IEEE Communications Letters*, vol. 22, no. 12, pp. 2627–2630, 2018.

[41] D. Wu, D. Zhang, C. Xu, H. Wang, and X. Li, "Device-Free WiFi Human Sensing: From Pattern-Based to Model-Based Approaches," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 91–97, 2017.

[42] X. Wang, X. Wang, and S. Mao, "RF Sensing in the Internet of Things: A General Deep Learning Framework," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 62–67, 2018.

[43] 3GPP, "Release 14 Description; Summary of Rel-14 Work Items," 3rd Generation Partnership Project, Technical Specification (TS) 21.914, 2018, version 14.0.0.

[44] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep Convolutional Neural Networks for Massive MIMO Fingerprint-based Positioning," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017.

[45] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-based Localization for Massive MIMO-OFDM System with Deep Convolutional Neural Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 846–10 857, 2019.

[46] S. Haykin, *Neural Networks: A comprehensive foundation*. Pearson, 1998.

[47] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-Art Speech Recognition with Sequence-to-Sequence Models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[48] Z. H. Zhou, *Ensemble Methods Foundations and Algorithms*. Chapman Hall/CRC, 2012.

[49] A. F. Molisch, *Wireless Communications: From Fundamentals to Beyond 5G*, 3rd ed. Wiley-IEEE Press, 2022.

[50] E. Björnson, J. Hoydis, and L. Sanguinetti, *Massive MIMO Networks: Spectral, Energy, and Hardware Efficiency*. now, 2017.

[51] M. Guillaud, D. Slock, and R. Knopp, "A Practical Method for Wireless Channel Reciprocity Exploitation through Relative Calibration," in *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications*, vol. 1, 2005.

[52] R. Rogalin, O. Y. Bursalioglu, H. Papadopoulos, G. Caire, A. F. Molisch, A. Michaloliakos, V. Balan, and K. Psounis, "Scalable Synchronization and Reciprocity Calibration for Distributed Multiuser MIMO," *IEEE Transactions on Wireless Communications*, vol. 13, no. 4, pp. 1815–1831, 2014.

[53] J. Vieira, F. Rusek, and F. Tufvesson, "Reciprocity Calibration Methods for Massive MIMO Based on Antenna Coupling," in *IEEE Global Communications Conference*, 2014.

[54] J. Vieira, F. Rusek, O. Edfors, S. Malkowsky, L. Liu, and F. Tufvesson, "Reciprocity Calibration for Massive MIMO: Proposal, Modeling, and Validation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3042–3056, 2017.

[55] X. Jiang, A. Decurninge, K. Gopala, F. Kaltenberger, M. Guillaud, D. Slock, and L. Deneire, "A Framework for Over-the-Air Reciprocity Calibration for TDD Massive MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 5975–5990, 2018.

[56] B. C. Ng and C. M. S. See, "Sensor-array Calibration Using a Maximum-likelihood Approach," *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 6, pp. 827–835, 1996.

[57] H. Liu, L. Zhao, Y. Li, X. Jing, and T.-K. Truong, "A Sparse-Based Approach for DOA Estimation and Array Calibration in Uniform Linear Array," *IEEE Sensors Journal*, vol. 16, no. 15, pp. 6018–6027, 2016.

[58] X. Luo, F. Yang, and H. Zhu, "Massive MIMO Self-Calibration: Optimal Interconnection for Full Calibration," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 10 357–10 371, 2019.

[59] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. USA: Prentice-Hall, Inc., 1993. ISBN 0133457117

[60] T. L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3590–3600, 2010.

[61] W. Koch, "The Radar Festival on the Rhine: A Highlight in the History of the IEEE AESS," *IEEE Aerospace and Electronic Systems Magazine*, vol. 38, no. 1, pp. 70–75, 2023.

[62] A. Liu, Z. Huang, M. Li, Y. Wan, W. Li, T. X. Han, C. Liu, R. Du, D. K. P. Tan, J. Lu, Y. Shen, F. Colone, and K. Chetty, "A Survey on Fundamental Limits of Integrated Sensing and Communication," *IEEE Communications Surveys Tutorials*, vol. 24, no. 2, pp. 994–1034, 2022.

[63] G. Bergqvist and E. G. Larsson, "The Higher-Order Singular Value Decomposition: Theory and an Application [Lecture Notes]," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 151–154, 2010.

[64] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sep. 2009.

[65] X. Li, "Parametric Radio Channel Estimation and Robust Localization," Doctoral Thesis (compilation), Lund University, May 2022.

[66] S. Hu, X. Li, and F. Rusek, "On Time-of-Arrival Estimation in NB-IoT Systems," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019.

[67] J. G. Proakis and D. K. Manolakis, *Digital Signal Processing (4th Edition)*. Prentice Hall, 2006.

[68] S. Kristoffersson Lind, Z. Xiong, P.-E. Forssén, and V. Krüger, "Uncertainty Quantification Metrics for Deep Regression," *Pattern Recognition Letters*, vol. 186, pp. 91–97, 2024.

[69] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles," in *Conference on Neural Information Processing Systems*, 2017.

[70] S. Farooq Bhat, I. Alhashim, and P. Wonka, "AdaBins: Depth Estimation Using Adaptive Bins," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[71] THE EUROPEAN PARLIAMENT and THE COUNCIL OF THE EUROPEAN UNION. (2024.07.12) Artificial Intelligence Act. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689&qid=1727554530273

[72] A. Salihu, M. Rupp, and S. Schwarz, "Self-Supervised and Invariant Representations for Wireless Localization," *IEEE Transactions on Wireless Communications*, vol. 23, no. 8, pp. 8281–8296, 2024.

[73] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

PAPERS

Paper I

Paper I

Reproduced, with permission, from: IEEE

G. Tian, H. OdetAlla, B. E. Priyanto, S. Hu, and F. Tufvesson, "Modified Gold Sequence for Positioning Enhancement in NB-IoT," *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, 2019, pp. 1-6, doi: 10.1109/WCNC.2019.8886076.

Modified Gold Sequence for Positioning Enhancement in NB-IoT

Guoda Tian[†], Hatem OdetAlla[†], Basuki E. Priyanto[‡],
Sha Hu[†], and Fredrik Tufvesson[†]
[†]Department of Electrical and Information Technology, Lund University, Lund, Sweden
[‡]Sony Mobile Communications AB, Lund, Sweden
[†]{firstname.lastname}@eit.lth.se, [‡]basuki.priyanto@sony.com

Abstract

Positioning is an essential feature in Narrow-Band Internet-of-Things (NB-IoT) systems. Observed Time Difference of Arrival is one of the supported positioning techniques for NB-IoT. It utilizes the downlink NB positioning reference signal (NPRS) generated based on a length-31 Gold sequence. Although a Gold sequence has good auto-correlation and cross-correlation properties, the correlation properties of NPRS in NB-IoT are still sub-optimal. The reason is mainly due to two facts: the number of NPRS symbols in each subframe is limited, and the featured sampling-rate is low. In this paper, we propose to modify the NPRS generation by exploiting the cross-correlation function of the NPRS. That is, for each orthogonal frequency division multiplexing (OFDM) symbol we generate the first NPRS symbol as specified in the current standard, i.e., a Gold sequence; while the second OFDM symbol is set to the additive inverse of the first one. Our simulation results show that the proposed NPRS sequence results in improving the correlation properties, particularly with respect to the cross-correlation property. Furthermore, 15%-30% positioning-accuracy improvements can be attained with the proposed method, compared to the legacy one under both Additive White Gaussian Noise and Extended-Pedestrian-A channels. The proposed NPRS sequence can also be applied to other similar systems, such as long-term-evolution (LTE).

I. INTRODUCTION

NB-IoT (Narrow-Band Internet of Things) has been developed by 3rd Generation Partnership Project (3GPP) to accommodate the expected massive connections of low-power IoT devices to the cellular network [1]. The advantages of this NB-IoT are its low complexity, coverage enhancement, and compatibility with existing cellular technologies. To reduce the complexity, NB-IoT applies simple baseband processing and operates at a reduced sampling rate [2], which significantly simplifies hardware requirements compared to other existing cellular communication techniques such as Long-Term Evolution (LTE). In addition, NB-IoT devices occupy one channel with narrower bandwidth (180 kHz) and the network can deploy anchor and non-anchor channels to accommodate a large number of devices connecting to the network using a limited bandwidth. Furthermore, NB-IoT has been designed to fit in various deployment modes (i.e. in-band, guard-band, and stand alone) so that it can smoothly co-exist with LTE/LTE-A systems.

Similar to many other technologies such as LTE, NB-IoT also provides positioning services of the user equipment (UE). 3GPP release 14 has a positioning target of 50 m (for 67 % of total measurements) and assumes a Positioning Reference Signal (PRS) to be transmitted in multiple subframes. The demand for positioning services has increased drastically with the rapid deployment of wireless networks. Therefore, positioning technologies have become a vital field for the research community. To name a few, Lin et al. gave an overview on IoT positioning technologies and compared the positioning performance between NB-IoT and LTE systems [3]. The work in [10] investigated the enhancements of positioning results by designing an adequate frequency-hopping pattern at the transmitter side, while [4] provided an overview of all NB-IoT features, including positioning, developed by 3GPP from release 13-15.

In this paper, our discussion and proposed method are based on a well-known downlink (DL) based positioning technique. A general overview of this approach and related receiver algorithms can be found in e.g., [5]–[7]. The work in [6] applied successive interference cancellation, and residual frequency-offset estimation algorithms at the receiver for the purpose of positioning accuracy enhancement. [7] provided an indoor positioning solution based on a channel state information fingerprinting approach. In addition, a new frequency hopping pattern was proposed in [8], which significantly increased the accuracy of the positioning results there.

To summarize, most of the research activities focus on improving the receiver algorithms and the frequency hopping patterns. However, the positioning accuracy can also be improved by carefully designing a sequence with good correlation properties. In this paper, we propose a novel approach by slightly modifying the original Gold sequence specified in the 3GPP standard before mapping it onto the time frequency resource grid, aiming at minimizing the cross-correlation. To be specific, we generate the first Narrow-Band Positioning Reference Signal (NPRS) symbol on each time frequency resource block according to the 3GPP specification [2], and constrain the second NPRS symbol to be the additive inverse of the first one. With such a modified NPRS sequence, the cross-correlation function (CCF) can be significantly suppressed and therefore the positioning performance can be improved. The proposed modification of the NPRS sequence does not involve a complete change of NPRS sequence generation. However, it can still achieve promising enhancements in terms of positioning accuracy. As shown by the simulation results, the NPRS sequences designed perform better than the existing NPRS sequence as specified in the standard in [2]. Bear in mind that our target is to further improve the positioning performance by modifying the PRS sequence, therefore the optimization of the receiver design is beyond the scope of this paper.

Our contributions of this paper are summarized as follows:

- We propose a novel approach to enhance the CCF of the NPRS sequence and its mathematical derivation under the framework of NB-IoT. The proposed method leads to a better CCF between different PRSs that belong to different cell-IDs.
- The suggested approach requires a straight-forward modification of the current standard. Thus the implementation impacts to the transmitter (i.e. base-station) and receiver (i.e. terminal) are very small.
- The approach has been validated through simulations applied on both Additive White

Gaussian Noise (AWGN) and Extended-Pedestrian-A (EPA) channel models in an NB-IoT system. In addition, the proposed method can also be applied to other systems (e.g. LTE).

The remainder of the paper is organized as follows. An overview of positioning in the 3GPP NB-IoT standards is given in Section II. Section III presents our approach of modifying PRSs, followed by a mathematical derivation. Section IV covers the receiver design, focusing on the correlator that operates in the frequency domain. Numerical results are presented in section V, and section VI concludes the paper.

II. OTDOA Positioning in NB-IoT

The fundamental operation of the OTDOA positioning is illustrated in Fig. 1. In principle, an estimate of the UE position is obtained under the aid of PRS transmissions from several base-stations, also known as evolved Node-Bs (eNBs). Typically, one eNB acts as the serving eNB and others as the neighbour eNBs. Each PRS can be received by the UE at different times due to the different propagation distances between the UE and each eNB. As shown in Fig 1, three eNBs simultaneously transmit PRSs while these signals arrive at the UE at different times, namely, τ_1 , τ_2 , and τ_3. When these signals are received by the UE, measurements of the time of arrival (TOA) of all PRSs take place, known as the Reference Signal Time Different (RSTD) measurement. Then, RSTD measurements are reported to a location server, typically via a serving eNB. Afterwards, the location server analyzes the RSTD measurements and creates the corresponding hyperbolas used for positioning estimation. Ideally, the coordinate of a unique intersection point of these hyperbolas can be calculated as the estimated UE position, since the geographical coordinates of the eNBs are known. In practice, however, the estimated position may deviate from the real position. It is essential for UE to receive PRSs from at least three eNBs in order to attain a unique intersection point.

In accordance with the current 3GPP specification [1], NPRSs are uniquely created based on the Physical Cell ID (PCI), slot number, orthogonal frequency division multiplexing (OFDM) symbol number and the type of cyclic prefix (CP). Furthermore, each NPRS sequence consists of two elements, each being a complex number. The mathematical expression of the NPRS sequence $z_{n_s,l}(q)$ is given by

$$z_{n_s,l}(q) = \frac{1}{\sqrt{2}}((1-2c(2q)) + j(1-2c(2q+1))) \tag{1}$$

where n_s and l represent the slot and symbol number, respectively; q indicates indices of elements of each NPRS sequence; As illustrated by [2], $c(q)$ is a length-31 Gold sequence initialized by the seed

$$\begin{aligned} c_{init} = 2^{28}\left\lfloor \frac{N_{ID}}{512} \right\rfloor + 2^{10}(7(n_s+1)+l+1)\times \\ (2(N_{ID} \bmod 512)+1) + 2(N_{ID} \bmod 512) + N_{CP}, \end{aligned} \tag{2}$$

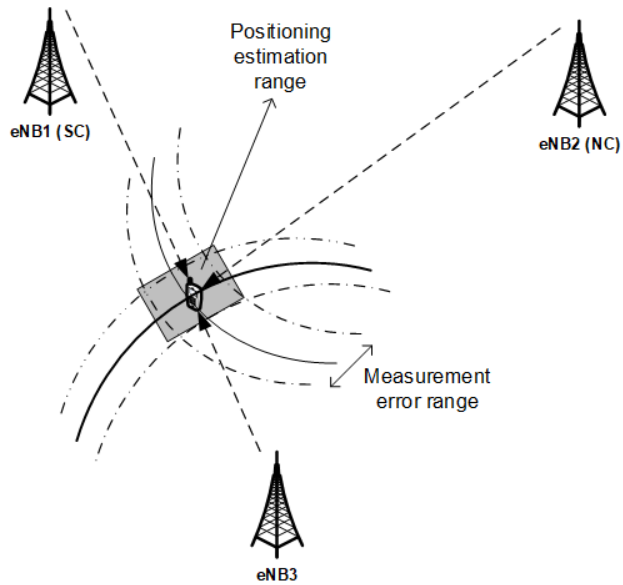


Fig. 1. NB-IoT Deployment Scenarios

where N_{ID} denotes PCI number of the corresponding eNB; N_{cp} depends on the type of cyclic prefix applied by the system (i.e normal or extended CP). If normal CP is applied by the system, one radio frame consists of 2 slots and 14 OFDM symbols.

After generating the NPRS signal, the next step is to map the symbols onto the time-frequency resource grid according to the deployment modes of the system. NB-IoT can be deployed in three different modes: in-band, guard-band and standalone modes. Generally, the positioning sequence $z_{n_s,\ell}(q)$ is mapped to complex Quadrature Phase Shift Keying (QPSK) symbols $\alpha^p_{k,\ell} = z_{n_s,\ell}(q)$, where p is defined as the index of antenna port and $k = 6n_s + (6 - \ell + v_{shift})$ indicates the subcarrier number inside one time frequency resource block. Typically, each eNB uses only one resource block and two subcarriers at each OFDM symbol in NB-IoT. Provided that the system operates on the in-band mode, ℓ and v_{shift} are configured as

$$\ell = \begin{cases} 3, 5, 6 & n_s \bmod 2 = 0 \\ 1, 2, 3, 5, 6 & n_s \bmod 2 = 1 \end{cases}$$
$$v_{shift} = N_{ID} \bmod 6. \tag{3}$$

In accordance with (3), 8 out of 14 symbols are reserved for NB-IoT positioning within one time-frequency resource block, while the other 6 symbols (shaded blue in Fig. 2) are allocated to be occupied for the other signaling purposes (e.g. NB Physical DL Control Channel).

Finally, an Inverse Fast Fourier transform (IFFT) is applied, followed by CP insertion and upconversion to radio frequency.

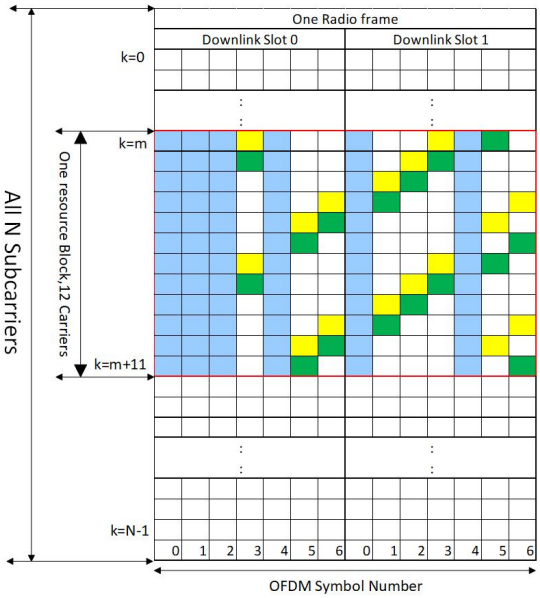


Fig. 2. The allocation of the OFDM symbol within one radio frame

III. The Proposed NPRS Generation

The Gold sequence has been chosen for the NPRS generation as described in [2] due to its good correlation properties. However, there is still room for improvements with respect to the cross-correlation properties by making adequate modifications to the existing NPRS generation method in [2]. We start by considering the NB-IoT system, which only uses 12 subcarriers as shown in Fig. 2.

As described in the previous section, the NPRS sequence $z_{n_s,l}(q)$ contains two elements only if one resource block is used, namely $z_{n_s,l}(0)$ and $z_{n_s,l}(1)$, both of which are generated by a length-31 Gold sequence in [2]. Here, we suggest that by generating the first element according to [2] and the second element as the additive inverse of the first one, the cross-correlation property of the modified sequence is improved. The derivations and explanations are follows.

Suppose that $x_1(n)$ and $x_2(n)$ represent the signals transmitted by two eNBs (eNB1 and eNB2), respectively, in the time domain and $X_1(k)$ and $X_2(k)$ are the corresponding signals in the frequency domain. Under the framework of NB-IoT, both $X_1(k)$ and $X_2(k)$ contain only two non-zero elements, as determined by the positioning sequence $z_{n_s,l}(q)$, at each OFDM symbol. Assuming that the system operates in in-band mode, 8 out of 14 OFDM symbols are utilized for positioning, as illustrated in Fig. 2. For simplicity, only the cross-correlation between $x_1(n)$ and $x_2(n)$ at OFDM symbol number 3 is derived as an example below; however, the result can be easily extended to other OFDM symbols. For OFDM symbol number 3, the frequency domain

components $X_1(k)$ and $X_2(k)$ vectors can be represented as:

$$x_1(k) = \begin{cases} a_1, & k = m, \\ a_2, & k = m+6, \\ 0, & \text{otherwise.} \end{cases} \quad x_2(k) = \begin{cases} b_1, & k = m+1, \\ b_2, & k = m+7, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where m denotes the index of the first subcarrier number occupied by eNB 1.

After conducting the IFFT (of size N), the time domain expressions of $x_1(n)$ and $x_2(n)$ are given by:

$$\begin{aligned} x_1(n) &= \mathcal{F}^{-1}\{X_1(k)\} = a_1 e^{j\frac{2\pi nm}{N}} + a_2 e^{j\frac{2\pi n(m+6)}{N}}, \\ x_2(n) &= \mathcal{F}^{-1}\{X_2(k)\} = b_1 e^{j\frac{2\pi n(m+1)}{N}} + b_2 e^{j\frac{2\pi n(m+7)}{N}}, \end{aligned} \tag{5}$$

where n is in the range from 0 to $N-1$. The CCF $A(\tau)$ between $x_1(n)$ and $x_2(n)$ is given by:

$$\begin{aligned} A(\tau) &= \sum_{n=-\infty}^{\infty} x_1(n) x_2^*(n+\tau) \\ &= \sum_{n=-\infty}^{\infty} (a_1 e^{j\frac{2\pi nm}{N}} + a_2 e^{j\frac{2\pi n(m+6)}{N}}) \\ &\quad \times (b_1^* e^{-j\frac{2\pi(n+\tau)(m+1)}{N}} + b_2^* e^{-j\frac{2\pi(n+\tau)(m+7)}{N}}), \end{aligned} \tag{6}$$

where τ represents the correlation lag. Expanding (6) and keeping in mind that the IFFT/FFT size is *N*, it holds that,

$$\begin{aligned} A(\tau) = e^{-j\frac{2\pi(m+1)\tau}{N}} &\left[\sum_{n=0}^{N-(\tau+1)} a_1 b_1^* e^{-j\frac{2\pi n}{N}} + a_2 b_1^* e^{j\frac{10\pi n}{N}} \right] \\ + e^{-j\frac{2\pi *(m+7)\tau}{N}} &\left[\sum_{n=0}^{N-(\tau+1)} a_1 b_2^* e^{-j\frac{14\pi n}{N}} + a_2 b_2^* e^{-j\frac{2\pi n}{N}} \right]. \end{aligned} \tag{7}$$

Since

$$\sum_{n=0}^{N-1} e^{j\frac{2\pi pn}{N}} = 0 \tag{8}$$

for $p \neq 0$, we can simplify the correlation function $A(\tau)$ as

$$\begin{aligned} A(\tau) = e^{\frac{-j2\pi(m+1)\tau}{N}} &\left[\sum_{n=-\tau}^{-1} \left(a_1 b_1^* e^{\frac{-j2\pi n}{N}} + a_2 b_1^* e^{\frac{j10\pi n}{N}} \right) \right. \\ &\left. + e^{\frac{-j12\pi\tau}{N}} \left(\sum_{n=-\tau}^{-1} \left(a_1 b_2^* e^{\frac{-j14\pi n}{N}} + a_2 b_2^* e^{\frac{-j2\pi n}{N}} \right) \right) \right]. \end{aligned} \tag{9}$$

Assuming that τ is relatively small compared to N, the exponential terms $e^{\frac{-j2\pi n}{N}}$, $e^{\frac{j10\pi n}{N}}$, $e^{\frac{-j14\pi n}{N}}$

and $e^{\frac{-j12\pi\tau}{N}}$ are close to 1. Therefore, (10) can be simplified as

$$\begin{aligned}\left|A(\tau)\right| &\approx \left|e^{\frac{-j2\pi(m+1)\tau}{N}}\right|\left|\tau(a_1b_1^* + a_2b_1^* + a_1b_2^* + a_2b_2^*)\right| \\ &= \tau\left|(a_1+a_2)b_1^* + (a_1+a_2)b_2^*\right|.\end{aligned} \tag{10}$$

In order to minimize the amplitude of $A(\tau)$, we could set $a_1 = -a_2$. With that, the correlation is close to zero regardless of the values of b_1 and b_2, which leads to the following theorem.

Theorem 1. *To minimize cross-correlations between NPRSs transmitted by different eNBs under the framework of NB-IoT with respect to the small lags τ and large FFT sizes N, the NPRS sequence $z_{n_s,l}(q)$ should be generated according to:*

$$\begin{aligned}z_{n_s,l}(0) &= \frac{1}{\sqrt{2}}(1-2c(0)) + j\frac{1}{\sqrt{2}}(1-2c(1)), \\ z_{n_s,l}(1) &= -z_{n_s,l}(0).\end{aligned} \tag{11}$$

where we set the first element of the sequence exactly the same as defined in the standard, while flipping the second element to be the opposite of the first.

Proof. Suppose that τ is small and $N \to \infty$, from (10) it holds that $|A(\tau)| \to 0$ if $z_{n,\ell}(q)$ is chosen according to (11). □

Therefore, in accordance with Theorem 1, to minimize the cross correlation for a small lag τ, for each pairs of frequency components of the same OFDM symbol and Cell ID, the first component needs to be set as the inverse of the second. Although Theorem 1 only shows the correlation property between two cells (cell-ID 1 and cell-ID 2), the cases with more cells can be accommodated by positioning systems utilizing the same principle. By selecting the NPRS according to Theorem 1, any arbitrary two cells can achieve favorable correlation properties.

A final remark is that it is reasonable to assume a small lag τ. Supposed that the largest radius of a typical NB-IoT cellular is R and the speed of light is c, the maximum possible propagation time in Line-of-Sight (LOS) is $\tau_{max} = \frac{R}{c}$. Since NB-IoT devices usually operate at a low sampling rate R_p, it is sufficient to consider samples up to $R_p\tau_{max}$. For example, considering an NB-IoT cellular network with radius 2 km and the sampling rate $R_p = 1.92$ MHz, the maximum possible delay in number of samples is around 13, which is much smaller than the IFFT size *N = 128*.

IV. Receiver Structure

In this section, we briefly present the receiver structure for NB-IoT positioning. As shown in Fig. 3, the receiver framework consists of signal combiner, correlator and peak detector, which shares much similarities with other OTDOA positioning systems. The correlator correlates the received reference signal (demodulated signal) and the locally generated reference signal. Keep in mind that the receiver must know of reference signal generated by the transmitter.

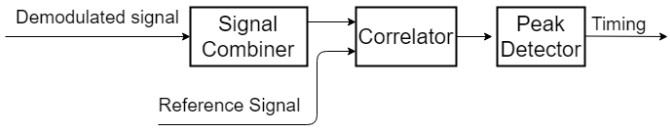


Fig. 3. NB-IoT Receiver Structure

Considering a scenario with M eNBs as transmitters and a single UE collecting the positioning signals from N slots, the propagation channel is described by

$$\boldsymbol{H} = \begin{bmatrix} h_{1,1} & \dots & h_{M,1} \\ \vdots & \ddots & \vdots \\ h_{1,N} & \dots & h_{M,N} \end{bmatrix}$$

and the received signal y is given by:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{12}$$

where $\boldsymbol{x}$ denotes as the transmit signal vector and $\boldsymbol{n}$ the noise vector.

A. Signal Combiner

Multiple repetitions of a subframe are usually transmitted by the transmitter for the sake of obtaining time-domain diversity and enhancing the positioning quality under in low signal-to-noise-ratio (SNR) conditions. The SNR is defined as the ratio between power of the received serving eNB signal and the noise. In order to effectively exploit the information provided by multiple signals, it is necessary to implement a signal combiner at the receiver side.

Normally, non-coherent combining can improve the SNR of the received signal while coherent combining can effectively exploit channel knowledge in addition to the noise reduction. For coherent combiners, we let $\boldsymbol{G}$ denote as the combining matrix while $\boldsymbol{z}$ represents the output of the combiner, apparently:

$$\boldsymbol{z} = \boldsymbol{G}\boldsymbol{y}. \tag{13}$$

Generally there are three main groups of coherent combining methods, namely, Zero Forcing (ZF), Minimum Mean Square Error (MMSE) or Matched Filter (MF) in equation (14).

$$\begin{aligned} \boldsymbol{G}_{ZF} &= (\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H \\ \boldsymbol{G}_{MMSE} &= \boldsymbol{H}^H(\boldsymbol{H}\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{H}^H + \boldsymbol{N}_{\boldsymbol{o}})^{-1} \\ \boldsymbol{G}_{MF} &= \boldsymbol{H}^H \end{aligned} \tag{14}$$

In equation (14), $\boldsymbol{R}_{\boldsymbol{xx}}$ represents the correlation matrix of the transmitted signal and $\boldsymbol{N}_{\boldsymbol{o}}$ the noise power density.

B. Correlator Design

The correlator is one of the key blocks in the positioning receiver chain for TOA estimation since it calculates the correlation between the received sequence and the reference pilots, as

shown in Fig. 4. The output signal of the correlator is supposed to feed the peak detector which in turn detects the delay samples of the initial path as the estimated TOA. In this paper, we focus on the frequency-domain correlator structure.

For the sake of accelerating the processing speed, a good solution to calculate the TOA is to process the signal in frequency domain by the aid of an FFT operation, as seen in Fig. 5 [11]. Leveraging FFT, time domain convolution can therefore be replaced by one-tap multiplication instead of convolution operation. Note that the circular cross-correlation rather than linear correlation is computed in accordance with the FFT properties. Regarding to receiver parameter setting, it is of necessity to enlarge the FFT window size to 256 (the typical window size for NB-IoT is 128), since the unknown delay of propagation and the length of cyclic prefix should be taken into consideration.

C. Peak Detector

A Peak detector works for detecting the initial peak of the correlation function as the estimated delay of the signal propagation. For conducting the maximum likelihood estimation, we can calculate the delay sample as, as illustrated by [9]

$$\hat{n}_p = \arg\max_n |R(n)| \tag{15}$$

Where $R(n)$ represents the cross-correlation function presented in Fig. 4, and $\hat{n}_p$ denotes the estimated delay sample.

V. Simulation Results

In this section, we present simulation results to illustrate the performance of the proposed NPRS sequence compared to the existing configuration specified in [2]. UEs are dropped uniformly within one hexagonal cell and the channel model is selected according to [12]. To assess the performance of the proposed NPRS sequence, the cumulative distribution function (CDF) of the positioning errors is selected as a criterion for evaluation. The parameter settings applied in the simulations are listed in Table I.

Various numbers of cell IDs combinations (both consecutive and random Cell-IDs) have been selected and tested. The simulation results show that the proposed method outperforms the legacy

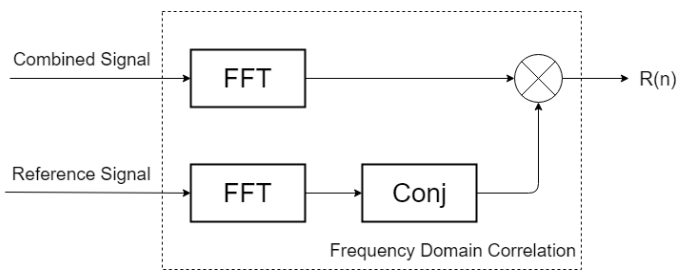


Fig. 4. Frequency Domain Correlator

solution for approximately 85% of the combinations while the remaining 15% performs equal or slightly worse than the standard, in terms of the position error corresponding to a probability of 0.5 on the CDF curve.

Fig. 5 presents a comparison of correlation properties of the proposed design against the standard. Two QPSK vectors $\boldsymbol{x_1} = [x_{11}, x_{12}]^T$, $\boldsymbol{x_2} = [x_{21}, x_{22}]^T$ are selected from the legacy and the modified sequence. Specifically, vectors $\hat{x}_1 = [1-j,\ 1-j]$ and $\hat{x}_2 = [1+j,\ -1-j]$ represent two symbols generated according to the standard while the vectors $\tilde{x}_1 = [1-j,\ -1+j]$ and $\tilde{x}_2 = [-1-j,\ 1+j]$ generated from the modified algorithm presented in this paper. In Fig. 5, the x axis represents samples of correlation lags while y axis the normalized correlation values. The blue line in Fig. 5 shows the main side lobe of ACF of the signal before modification while the red line and black line show the CCFs corresponding to the legacy and the presented work, respectively. It is clear that the proposed scheme has very good CCF, especially at small lags, which aids the peak detector to correctly detect the delay owing to less interference caused by the CCF onto the ACF peak.

Fig. 6 shows a comparison between the CDFs of the position error of the legacy sequence (shown as the reference in Fig.6, the same in Fig. 7 and Fig. 8) and the proposed sequence under two SNR conditions (10 dB and -10 dB). Due to the good AWGN channel properties (frequency flatness), only NPRS with one subframe is transmitted to evaluate the positioning performance. As illustrated, around 27% and 13% improvements (the positioning error corresponding to probability 0.5) can be achieved in both high and low SNR conditions, respectively.

Fig. 7 illustrates the simulation results of the EPA propagation channel. Because of the more complex propagation scenario compared to AWGN, NPRS with 10 subframes are transmitted and matched filter combining is applied in order to improve the positioning performance. The simulation results show that the enhancement reduces to 15%, due to multipath propagation and Doppler frequency shift.

Fig. 8 shows a simulation result with wider bandwidth (2 PRBs) and 1 subframe under the

TABLE I
SIMULATION PARAMETERS SETTINGS

Parameters	Values
Number of eNBs	6
Number of UEs	1000
Intersite-distance	1732 m
Operation Mode	In-band
SNR	-10 and 10 dB
Consecutive NPRS Subframes	1 and 10
NPRS muting	NA
Doppler spread	25 Hz for EPA, 0 Hz for AWGN
UE sampling frequency	1.92 MHz
Standard Deviation for Shadowing	8 dB
Inter-cell shadowing correlation	0.5
Pass Loss model (r_p in km)	$(\alpha_p)_{dB} = 120.9 + 37.6\log_{10}\mathrm{r}_p$

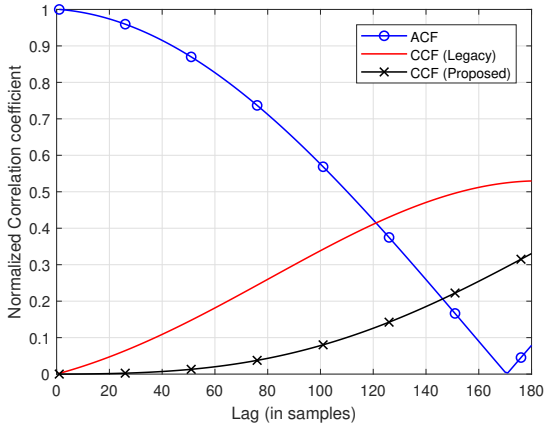


Fig. 5. Correlation properties of the proposed sequence vesus legacy sequence in NB-IoT.

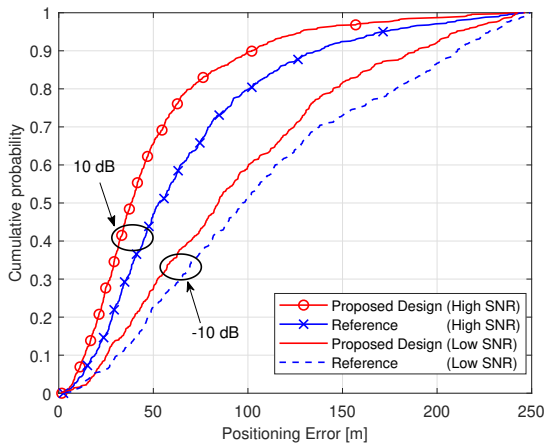


Fig. 6. CDF of the positioning error under the AWGN channel with 1 PRB in the frequency domain and 1 subframe.

AWGN channel. It is obviously a non-NB-IoT system since it occupies more than 1 PRB. Our intention is to investigate the impact of wider bandwidth on the performances of the proposed method. In general, positioning results achieve steadily enhancements owing to the sharper ACF main-lobe resulted from wider system bandwidth. Here, we can still obtain the benefits of the proposed method.

VI. Conclusion

In this paper, we have proposed a novel approach by modifying the Gold sequence of NPRS generation for achieving better CCF properties. The advantages of the proposed NPRS sequence

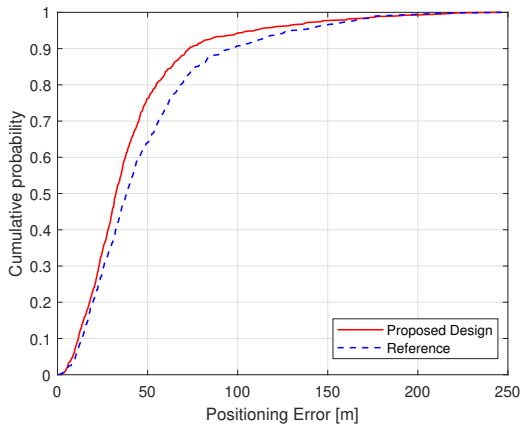


Fig. 7. CDF of the positiong error under EPA channel with 1 PRB in the frequency domain and 10 subframes.

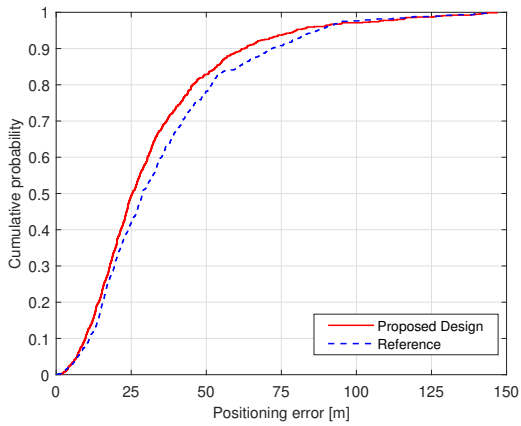


Fig. 8. CDF of the positiong error for AWGN channel with 2 PRBs in the frequency domain and 1 subframe.

have been verified by simulation results, which showed 15%-30% positioning-accuracy improvements both under AWGN and EPA channels. Moreover, the influence on both standardization and UE implementation impacts is very small, since it only requires a straight-forward modification of the sequence generation of the NPRS signal. At the receiver side, a coherent combining algorithm has been utilized to compensate for the effect of the propagation channel. The proposed frequency domain correlator significantly reduces the calculation complexity compared to the time-domain approach. Finally, the proposed NPRS can also be aplied to other systems such as LTE in order to enhance the positioning accuracy performance there.

ACKNOWLEDGEMENT

The authors would like to thank Prof. Fredrik Rusek for the technical discussions and assistance with the receiver design, Dr. Peter Karlsson and Sony Mobile Communications AB for their technical assistance and financial support.

REFERENCES

[1] Huawei, "Narrowband IoT side range of opportunities," Tech. Rep., Nov. 2016

[2] 3GPP Technical Specification (TS) 36.211, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation," version 15.2.0, 2018.

[3] X. Lin et al., "Positioning for the Internet of Things: A 3GPP Perspective," IEEE Communi. Mag., vol. 55, no. 12, pp. 179-185, Dec. 2017.

[4] R. Ratasuk, N. Mangalvedhe, Z. Xiong, M. Robert, and D. Bhatoolaul, "Enhancements of narrowband IoT in 3GPP Rel-14 and Rel-15," IEEE Conf. Stand. Commun. and Netw. (CSCN), Helsinki, 2017, pp. 60-65.

[5] K. Radnosrati, G. Hendeby, C. Fritsche, F. Gunnarsson, and F. Gustafsson, "Performance of OTDOA positioning in narrowband IoT systems," IEEE Ann. Int. Symp. Personal, Indoor, and Mobile Radio Commun. (PIMRC), Montreal, QC, 2017, pp. 1-7.

[6] S. Hu, A. Berg, X. Li, and F. Rusek, "Improving the Performance of OTDOA Based Positioning in NB-IoT Systems," IEEE Global Commun. Conf. (GLOBECOM), Singapore, 2017, pp. 1-7.

[7] Q. Song, S. Guo, X. Liu, and Y. Yang, "CSI Amplitude Fingerprinting-Based NB-IoT Indoor Localization," *IEEE Internet of Things J.,* vol. 5, no. 3, pp. 1494-1504, Jun. 2018.

[8] W. S. Jeon, S. B. Seo, and D. G. Jeong, "Effective Frequency Hopping Pattern for ToA Estimation in NB-IoT Random Access," to appear in *IEEE Trans. Veh. Technol.,* Jul. 2018.

[9] S. M. Kay, *Fundamentals of Statistical Signal Processing: Volume I: Estimation Theory,* Prentice Hall, 2011.

[10] A. del Peral-Rosado, J. A. López-Salcedo, and G. Seco-Granados, "Impact of frequency-hopping NB-IoT positioning in 4G and future 5G networks," IEEE Int. Conf. Commun. Workshops (ICC Workshops), Paris, 2017, pp. 815-820.

[11] J. A. Del Peral-Rosado, "Evaluation of the LTE positioning capabilities in realistic navigation channels," Ph.D. dissertation, Dept. Telecomunicació Enginyeria Sistemes, Univ. Autonoma Barcelona, Barcelona, Spain, 2014.

[12] 3GPP Technical Specification (TS) 36.104, "Evolved Universal Terrestrial Radio Access (E-UTRA); Base Station (BS) radio transmission and reception," version 15.3.0, 2018.

Paper II

Paper II

Reproduced, with permission, from: IEEE

G. Tian, H. Tataria, and F. Tufvesson, "Amplitude and Phase Estimation for Absolute Calibration of Massive MIMO Front-Ends," *2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1-7, doi: 10.1109/ICC40277.2020.9148962.

Amplitude and Phase Estimation for Absolute Calibration of Massive MIMO Front-Ends

Guoda Tian, Harsh Tataria, and Fredrik Tufvesson Department of Electrical and Information Technology, Lund University, Lund, Sweden
e-mail: {guoda.tian, harsh.tataria, fredrik.tufvesson}@eit.lth.se

Abstract

Massive multiple-input multiple-output (MIMO) promises significantly higher performance relative to conventional multiuser systems. However, the promised gains of massive MIMO systems rely heavily on the accuracy of the *absolute front-end calibration*, as well as quality of *channel estimates* at the base station (BS). In this paper, we analyze *user equipment-aided* calibration mechanism to estimate the *amplitude* scaling and *phase drift* at each radio-frequency chain connected to the BS array. Assuming a uniform linear array at the BS and Ricean fading, we obtain the estimation parameters with *moment-based* (amplitude, phase) and *maximum-likelihood* (phase-only) estimation techniques. In stark contrast to previous works, we mathematically articulate the *equivalence* of the two approaches for phase estimation. Furthermore, we rigorously derive a *Cramér-Rao lower bound* to characterize the accuracy of the two estimators. Via numerical simulations, we evaluate the estimator performance with varying dominant line-of-sight powers, dominant angles-of-arrival, and signal-to-noise ratios.

I. Introduction

Fifth-generation (5G) systems are being deployed into commercial networks [1]. The standardization efforts have resulted in a new radio access framework, known as Third Generation Partnership Project Release 15 (and beyond) [2]. A fundamental technology contributing to the spectral and energy efficiency targets of 5G systems is *massive multiple-input multiple-output (MIMO)*. By scaling up the number of antennas at cellular base stations (BSs), massive MIMO sharply increases the *beamforming gain* of the system, and enhances the ability to provide *uniformly* good service to each user equipment (UE) [3]–[5]. This has resulted in an order-of-magnitude *increase* in the average spectral efficiency of 5G systems relative to their fourth-generation counterpart [4].

Since its inception in 2010, a vast amount of literature has developed around characterizing different performance aspects of massive MIMO systems (see e.g., [7], [8], [18] for a summary). Nevertheless, the promised gains of massive MIMO greatly hinge on two key factors: (1) the *knowledge* of the channel state information at the BS and UE, (2) *calibration quality* (precise definition presented later in the text). According to the related literature, massive MIMO calibration approaches are generally classified into two categories: namely, *reciprocity* calibration [10]–[14] and *absolute* calibration [15]–[17], [21]–[24]. Reciprocity calibration is required in massive MIMO to ensure that the downlink channel is *reciprocal* to the uplink. The concept of the relative reciprocity calibration was first introduced in [10]. Extending this, a high-level network protocol of UE synchronization and reciprocity-based calibration was presented in [11].

Moreover, the authors of [12], [13] derived several practical approaches for reciprocity calibration and validated the results in real-time via the Lund University massive MIMO testbed. A taxonomy of the existing reciprocity calibration methods with an antenna grouping strategy is proposed in [14] to *shorten* the calibration time. In contrast to reciprocity calibration, absolute calibration, is required for angle-of-arrival (AOA) estimation and positioning. Absolute calibration exploits the amplitude and phase spectra *across* the BS array, as shown in [15], [16]. Approaches such as *intra-array* and *UE-aided* calibration are discussed in [17], [21], [22]. The authors of [23] combine array calibration with AOA estimation, while the authors of [24] propose mutual coupling-based methods for estimating the phase and amplitude relationships between each radio-frequency (RF) chain at the BS.

The intra-array based calibration can be implemented either with or without transmission lines between antenna elements. The later case outperforms the former in terms of *interconnect flexibility* at the cost of calibration accuracy, since its performance *degrades* with increasing electrical distance between successive antennas [17]. For UE-aided calibration, a better trade-off between the flexibility and accuracy is expected, and is therefore worth further investigation. To our best knowledge, prior works on UE-aided calibration only consider simple additive white Gaussian noise (AWGN) channels, which naturally do not reflect the physics of wave propagation. To this end, we analyze UE-aided absolute calibration over a Ricean fading channel, often used to model dominant line-of-sight (LOS) components in addition to diffuse multipath components [18], [19]. We provide a methodology to analyze two types of practical estimators (described later in the text) and derive the corresponding Cramér-Rao lower bound (CRLB) for evaluating the quality of amplitude and phase estimates.

Our main contributions are as follows: For an uplink single-user massive MIMO system, assuming a uniform linear array (ULA) at the BS and Ricean fading propagation, we establish two general, yet practical, analytical approaches to estimate the amplitude scaling and phase drift associated with each RF chain. The first approach is based on *moment-based* estimation of the aforementioned parameters, while the second is based on *maximum-likelihood* estimation (MLE), for obtaining phase estimates. We mathematically show that both estimators have an equivalent form when estimating the phase of the RF chains, and back up the mathematical findings with the required physical intuition. For evaluating the accuracy of both estimators, we derive the CRLB to characterize the fundamental lower limits on error of the estimated phase and amplitude scaling coefficients across the array. *To the best of our knowledge, this has been missing from the literature.* We evaluate the derived estimator performance on a ULA-based *numerical* framework. We show that under the presence of dominant LOS conditions, the *variance* of the phase estimates rapidly converges to the predicted CRLB for both estimator types. In addition, the amplitude and phase estimation accuracies of both approaches significantly improve with growing LOS powers and signal-to-noise ratios (SNRs).

II. System Model

We consider the *uplink* of a single-user massive MIMO system, which has M antenna elements configured in a ULA at the BS. We assume reciprocity-based operation in the time-division

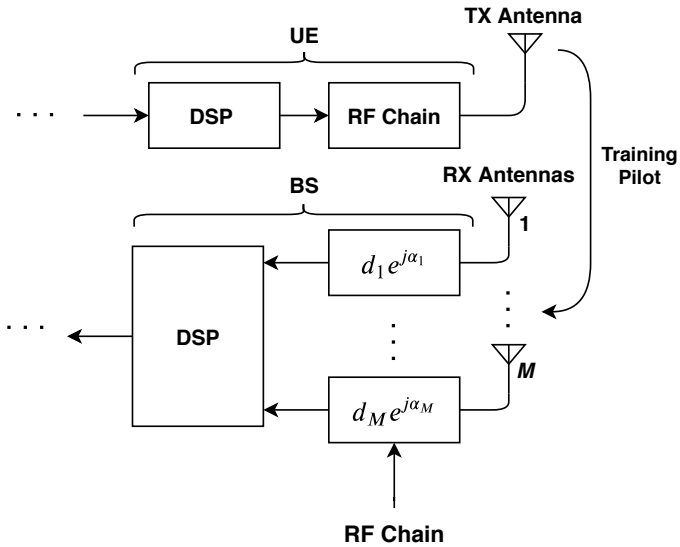


Fig. 1: A single-user uplink massive MIMO system with pilot transmission from the UE to the M BS antennas, which are interfaced with M RF chains.

duplex mode where the UE sends uplink pilot signals, which are used to *estimate* the calibration parameters at the M RF chains interfacing with the receive antennas. The overall system model is depicted in Fig. 1. We assume narrowband propagation between the UE and the BS, with uniform power allocation. More specifically, we employ the use of a general *Ricean* fading model, where the small-scale fading impulse response is an amalgamation of a dominant LOS component, in addition to the diffuse multipath components. The LOS component is governed by the far-field array steering vector in a given direction, and the diffuse components are modeled as complex Gaussian random variables (exact definition later in the text). The use of such model is rather popular in massive MIMO performance evaluation, particularly in urban scenarios where many diffuse paths are expected with some dominant LOS components [25]–[27]. Considering this, the received signal observation vector, $\mathbf{y}_t \in \mathbb{C}^{M\times 1}$ during time t can be written as

$$\mathbf{y}_t = \underbrace{\gamma_t \mathbf{D}_t \mathbf{a}(\phi_t) p_t}_{\mathbf{s}_t} + \underbrace{\mathbf{D}_t \mathbf{h}_t p_t + \mathbf{n}_t}_{\boldsymbol{\omega}_t}, \tag{1}$$

where γ_t and p_t are scalar quantities which denote the large-scale LOS power and the pilot transmitted by the UE at time t. The power contained in p_t is normalized to unity, such that $|p_t|^2 = 1$, over all values of $t = 1, 2, \ldots, T$. Since we consider a ULA, the array steering vector is a known function of the azimuth AOA, which is denoted as $\mathbf{a}(\phi_t) \in \mathbb{C}^{M\times 1}$ with an incoming angle ϕ_t. In addition, the vectors $\mathbf{h}_t \in \mathbb{C}^{M\times 1}$ and $\mathbf{n}_t \in \mathbb{C}^{M\times 1}$ denote the diffuse multipath components and the AWGN at time t, such that $\mathbf{h}_t \sim \mathcal{CN}(0, \sigma^2)$ and $\mathbf{n}_t \sim \mathcal{CN}(0, N_0/2)$. To

this end, the mean of the diffuse components is zero and the variance (power) is σ^2 across all $t = 1, 2, \ldots, T$. Likewise, the mean of the AWGN at the BS is zero and variance is $N_0/2$. Following this, the SNR at time t is given by $|p_t|^2/(N_0/2)$. The diagonal matrix, $\mathbf{D}_t \in \mathbb{C}^{M\times M}$, contains the M *amplitude scaling* and the *phase drift* entries for each RF chain. This matrix models the random phase and amplitude changes introduced by phase jitter at the local oscillators, and RF signal conditioning units such as low-noise amplifiers and active bandpass filters. We note that $\mathbf{D}_t = \text{diag}(d_1 e^{j\alpha_1}, d_2 e^{j\alpha_2}, ..., d_M e^{j\alpha_M})$. We further assume that $\mathbf{h}_t$ and $\mathbf{n}_t$ are statistically independent and $\mathbf{n}_t$ is uncorrelated over $t = 1, 2, \ldots, T$. With the above in mind, the auto-correlation at time t, $\mathbb{E}\{\boldsymbol{\omega}_t \boldsymbol{\omega}_t^H\}$, can be evaluated as

$$\begin{aligned}\mathbb{E}\left\{\boldsymbol{\omega}_t\boldsymbol{\omega}_t^H\right\} &= \mathbb{E}\left\{(\mathbf{D}_t\mathbf{h}_t p_t + \mathbf{n}_t)(\mathbf{D}_t\mathbf{h}_t p_t + \mathbf{n}_t)^H\right\} \\ &= N_0\mathbf{I}_M + \mathbf{D}_t|p_t|^2\sigma^2\mathbf{D}_t^H, \end{aligned} \tag{2}$$

where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. Moreover, by definition, the cross-correlation between two time intervals, namely $t = 1$ and $t = 2$, can be expressed as

$$\mathbb{E}\{\boldsymbol{\omega}_{t=1}\boldsymbol{\omega}_{t=2}^H\} = \sigma^2\mathbf{D}_t\mathbf{D}_t^H. \tag{3}$$

Between multiple time instances, the channel, $\mathbf{h}_t$, is assumed to be changing in accordance with its definition. This can be caused by small changes in the UE position, or mobility of objects in the propagation environment. For simplicity, from here onward, we drop the subscript t used in the right-hand side of (1), and assume that all further computations are performed at a given time instance t. Therefore, the received vector $\mathbf{y}_t$ follows a complex Gaussian distribution given by

$$\mathbf{y}_t \sim \mathcal{CN}\left(\gamma\mathbf{D}\mathbf{a}(\phi)p, N_0\mathbf{I}_M + \mathbf{D}|p|^2\sigma^2\mathbf{D}^H\right). \tag{4}$$

Given the model in (1)-(4), $\mathbf{y}_t$, p and $\mathbf{a}(\phi)$ are assumed to be known by the BS. Other parameters such as γ, σ^2, $\mathbf{D}$, $\mathbf{h}$, $\mathbf{n}$ are assumed to be *unknown*, which is the case in practice. Observing over T intervals, the *composite* received signal is given by stacking all $\mathbf{y}_t$ across $t = 1, 2, \ldots, T$ obtaining

$$\mathbf{y} \sim \mathcal{CN}\big(\underbrace{\mathbf{1}\otimes\gamma p\mathbf{D}\mathbf{a}(\phi)}_{\boldsymbol{\mu}(\boldsymbol{\xi})}, \underbrace{\mathbf{I}_T\otimes\mathbf{I}_M N_0 + \widetilde{\mathbf{I}}_T\otimes\sigma^2\mathbf{D}\mathbf{D}^H}_{\mathbf{C}(\boldsymbol{\xi})}\big), \tag{5}$$

where $\mathbf{1} \in \mathbb{R}^{T\times 1}$ is a column vector of unit entries, $[1, 1, \ldots, 1]^T$, while $\widetilde{\mathbf{I}}_T = \mathbf{1}.(\mathbf{1}^T) \in \mathbb{R}^{T\times T}$ is a matrix containing unit entries. In addition, $\otimes$ is the Kronecker product operation, $\boldsymbol{\mu}(\boldsymbol{\xi})$ denotes the mean vector and $\mathbf{C}(\boldsymbol{\xi})$ denotes the variance over all T time intervals. Note that the vector argument $\boldsymbol{\xi}$ contains the unknown quantities in d, α, σ^2 and γ, respectively. That is, $\boldsymbol{\xi} = [d_1, d_2, \ldots, d_M, \alpha_1, ..., \alpha_M, \sigma^2, \gamma]^T$. With this setup, the subsequent section of the paper discusses the phase and amplitude estimation techniques with the aim to calibrate M RF chains at M antennas of the BS.

III. Estimator Design

In order to perform absolute calibration, one needs to estimate the subspace spanned by the vector $\mathbf{v} = \{d_1 e^{j\alpha_1}, d_2 e^{j\alpha_2}, ..., d_M e^{j\alpha_M}\}$. According to this requirement, we analyze *two*

estimators, namely the moment-based estimator and the MLE estimator for estimating the phase drift vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_M]^T$ and a moment-based estimator for estimating the vector space $\mathbb{R}^{M\times 1}$ spanned by the magnitude vector $\mathbf{d} = [d_1, d_2, ..., d_M]^T$ of the RF chains. Later on, we prove that the analytical expression of the moment-based phase drifting estimator coincides with the MLE-based estimator.

A. Phase Estimation

1) Moment-Based Estimator: From (5), we can see that the information regarding the RF phase drifts is only embedded in the *first-order* statistics of the composite received signal. Enlightened by this, we analyze the moment-based estimator which computes the *expectation* of the composite vector $\mathbf{y}$ before estimating the phase drifts. The phase vector $\boldsymbol{\alpha} = \left[\alpha_1, \alpha_2, \ldots, \alpha_M\right]^T$ is estimated by

$$\hat{\boldsymbol{\alpha}} = \arg\left\{\sum_{t=1}^{T} \mathbf{y}_t\right\} - \arg\left\{\mathbf{a}(\phi)\right\} - \arg\left\{\mathbf{1}p\right\}. \tag{6}$$

Sketch of Proof: Via some straightforward algebra, one can show that $\mathbb{E}\{\mathbf{y}_t\} = \gamma \mathbf{D}\mathbf{a}(\phi)p$. With a *large* total observation time in T, one can expect that the empirical probability distribution of $\mathbf{y}_t$ converges almost surely to its true probability distribution. Using this fact allows us to accurately approximate the moments (first-order only, since second-order contains no utilizable information) of the empirical distribution with the true distribution, such that $\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t \approx \mathbb{E}\{\mathbf{y}_t\}$. Taking $\arg\{\frac{1}{T}\sum_{t=1}^{T}\mathbf{y}_t\} = \arg\{\mathbf{a}(\phi)\} + \arg\{\mathbf{1}p\} + \boldsymbol{\alpha}$, and solving for $\boldsymbol{\alpha}$ yields the desired phase estimate.

2) MLE: If $\mathbf{y}$ has the probability distribution function $\bar{p}(\boldsymbol{\xi}, \boldsymbol{y})$, then the MLE formulates an optimization problem on the maximization of the log-likelihood function. That is

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= \arg\max\left\{\bar{p}(\boldsymbol{\xi}, \mathbf{y})\right\} \\ &\overset{(a)}{=} \min_{\alpha}\left\{\ln\det\left(\boldsymbol{\Lambda}\right) + \boldsymbol{\beta}^H\mathbf{C}^{-1}\left(\boldsymbol{\xi}\right)\boldsymbol{\beta}\right\},\end{aligned} \tag{7}$$

where $\boldsymbol{\beta} = \mathbf{y} - (\mathbf{1}\otimes\gamma\mathbf{D}\mathbf{a}(\phi))$, $\mathbf{C}\left(\boldsymbol{\xi}\right) = \tilde{\mathbf{Q}}^H\boldsymbol{\Lambda}\tilde{\mathbf{Q}}$, with $\tilde{\mathbf{Q}} = \mathbf{Q}\otimes\mathbf{I}_M$ and $\mathbf{Q}\in\mathbb{C}^{T\times T}$ is defined as the normalized discrete Fourier transform (DFT) matrix. Moreover, $\boldsymbol{\Lambda}$ is defined as

$$\boldsymbol{\Lambda} = \begin{bmatrix}\sigma^2 T\mathbf{D}\mathbf{D}^H & & \\ & \ddots & \\ & & \mathbf{0}\end{bmatrix} + N_0\mathbf{I}_{MT},$$

and is a $MT\times MT$ matrix. In (7), (a) is a result of equivalently minimizing the argument of the exponential function in $\bar{p}(\boldsymbol{\xi}, \boldsymbol{y})$. Substituting $\mathbf{1}\otimes\mathbf{D}\mathbf{a}(\phi) = \sqrt{T}\tilde{\mathbf{Q}}^H\left[\mathbf{I}\ldots\mathbf{0}\right]^T\mathbf{D}\mathbf{a}(\phi)$ in (7) and simplifying yields

$$\begin{aligned}\hat{\boldsymbol{\alpha}} = \min_{\alpha}\{&\ln\det\boldsymbol{\Lambda} + \mathbf{y}^H\mathbf{C}^{-1}(\boldsymbol{\xi})\mathbf{y} \\ &- 2\gamma\sqrt{T}\Re\Big[\mathbf{a}^H(\phi)\mathbf{D}^H\big[(\sigma^2 T\mathbf{D}\mathbf{D}^H + N_0\mathbf{I}_M)^{-1}\ldots\mathbf{0}\big]\tilde{\mathbf{Q}}\mathbf{y}\Big] \\ &+ \gamma^2 T\mathbf{a}^H(\phi)\mathbf{D}^H(\sigma^2 T\mathbf{D}\mathbf{D}^{\mathbf{H}} + N_0\mathbf{I}_M)^{-1}\mathbf{D}\mathbf{a}(\phi)\}. \end{aligned} \tag{8}$$

Note that $\Re[\cdot]$ denotes the real component of a complex quantity. From (8), it is clear that the phase information is only contained in the term $\Re[\mathbf{a}^H(\phi)\,\mathbf{D}^H[(\sigma^2 T\mathbf{D}\mathbf{D}^H + N_0\mathbf{I}_M)^{-1}\ldots\mathbf{0}]\tilde{\mathbf{Q}}\mathbf{y}]$. Thus, can derive the MLE of $\boldsymbol{\alpha}$ as

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= \max_{\alpha}\Big\{\Re\Big[\mathbf{a}^H(\phi)\,\mathbf{D}^H\big[(\sigma^2 T\mathbf{D}\mathbf{D}^H + N_0\mathbf{I}_M)^{-1}\ldots\mathbf{0}\big]\tilde{\mathbf{Q}}\mathbf{y}\Big]\Big\} \\ &= \arg\left\{\sum_{t=1}^{T}\mathbf{y}_t\right\} - \arg\left\{\mathbf{a}(\phi)\right\} - \arg\left\{\mathbf{1}p\right\}. \end{aligned} \tag{9}$$

The above result is mathematically equivalent to the one derived from the moment-based estimator in (6). The intuition behind this equivalence can be explained as follows: The received vector $\mathbf{y}$ follows complex Gaussian distribution, and hence the first and second-order statistics of $\mathbf{y}$ contain the vast majority of its underlaying information. To this end, the *optimal* solution can be found by exploiting the first and second-order statistics [28]. For the moment-based estimator, since the second-order statistics do not contribute to the phase estimates, the first-order statistics can be used to derive an optimal estimator, which is identical to the MLE.

B. Amplitude Estimation

We now analyze the moment-based estimator for deriving the amplitude scaling coefficients of the M RF chains. We refrain from utilizing the MLE for amplitude estimation as the presence of higher-order terms makes maximization of the log-likelihood function a mathematically complex task. We estimate the vector space spanned by $\mathbf{d}$. Unlike for phase estimation, since *both* the first and second-order statistics of $\mathbf{y}$ contains useful information, it is necessary to estimate the covariance matrix $\mathbf{C}(\boldsymbol{\xi})$, which we denote as $\hat{\mathbf{C}}(\boldsymbol{\xi})$. We observe that the upper and lower triangular block diagonal sub matrices of $\hat{\mathbf{C}}(\boldsymbol{\xi})$ contain the relevant terms for $\sigma^2\mathbf{d}\odot\mathbf{d}$, which can be extracted for estimation. Note that $\odot$ denotes the Hadamard product. We therefore provide a *closed-from* solution for the moment-based amplitude estimator as

$$\hat{\mathbf{d}} = \sqrt{\sum_{t=1}^{T}\mathbf{y}_t\odot\mathbf{y}_t^* + \text{vecdiag}\Big[\sum_{t=1}^{T}\sum_{\substack{t'=1\\t'\neq t}}^{T}\tilde{\mathbf{C}}\left(\boldsymbol{\xi}\right)|_{(t,t')}\Big]}, \tag{10}$$

where $*$ represents the complex conjugate operation and "vecdiag" is an operation which extracts and stacks the diagonal elements of a matrix into a vector. Also, $\tilde{\mathbf{C}}(\boldsymbol{\xi})|_{(t,t')} \in \mathbb{C}^{M\times M}$ represents the $(t,t')-$th sub-matrix of $\hat{\mathbf{C}}(\boldsymbol{\xi})$.

IV. CRLB Analysis

We derive the Fisher Information Matrix (FIM), followed by the analytical squared estimation error bound for evaluating the accuracy of the estimators in the previous section.

A. FIM

The derivation of FIM starts from equation (5), according to [29], the FIM of the unknown vector $\boldsymbol{\xi}$ is given by

$$\begin{aligned}\mathbf{I}(\boldsymbol{\xi})_{i,j} = \mathrm{Tr}\Big[\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_i}\mathbf{C}^{-1}(\boldsymbol{\xi})\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_j}\mathbf{C}^{-1}(\boldsymbol{\xi})\Big] \\ + 2\Re\Big[\frac{\partial \boldsymbol{\mu}^H(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_i}\mathbf{C}^{-1}(\boldsymbol{\xi})\frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}_j}\Big].\end{aligned} \tag{11}$$

We exercise a slight abuse of notation here when we denote the FIM as $\mathbf{I}(\boldsymbol{\xi})$*, since a* $M \times M$ *identity matrix is denoted by* $\mathbf{I}_M$. We note that $\mathbf{I}(\boldsymbol{\xi}) \in \mathbb{C}^{(2M+2)\times(2M+2)}$. Furthermore, $\mathrm{Tr}[\cdot]$ denotes the matrix trace operator. According to (11), the FIM $\mathbf{I}(\boldsymbol{\xi})$ is an addition of two matrices, namely, $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}$ and $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}$, where $[\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}]_{i,j}$ is defined as the $(i,j)-$ th element of $\mathrm{Tr}[(\partial \mathbf{C}(\boldsymbol{\xi}))/(\partial \boldsymbol{\xi}_i)\mathbf{C}^{-1}(\boldsymbol{\xi})(\partial \mathbf{C}(\boldsymbol{\xi}))/(\partial \boldsymbol{\xi}_j)\,\mathbf{C}^{-1}(\boldsymbol{\xi})]$. Likewise, $[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{i,j}$ is defined as the $(i,j)-$th element of $2\Re[(\partial \boldsymbol{\mu}^H(\boldsymbol{\xi})/\partial \boldsymbol{\xi}_i)\mathbf{C}^{-1}(\boldsymbol{\xi})(\partial \boldsymbol{\mu}(\boldsymbol{\xi})/\partial \boldsymbol{\xi}_j)]$. We first evaluate $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}$, which begins with calculating the *derivative* of the $\mathbf{C}(\boldsymbol{\xi})$ with respect to elements in $\boldsymbol{\xi}$. That is,

$$\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \gamma} = 0 \quad \text{and} \quad \frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \sigma^2} = \widetilde{\mathbf{I}}_T \otimes \mathbf{D}\mathbf{D}^H. \tag{12}$$

For every RF chain, $m = 1, 2, \ldots, M$,

$$\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \alpha_m} = 0 \quad \text{and} \quad \frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial d_m} = \widetilde{\mathbf{I}}_T \otimes \sigma^2 \widetilde{\mathbf{D}}_m. \tag{13}$$

Here $\widetilde{\mathbf{D}}_m = \mathrm{diag}\{0, \ldots, 2d_m, \ldots, 0\}$ denotes a diagonal matrix. Closely observing (12) and (13), one can see that $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}$ contains all *zero* elements except for the sub-matrix blocks of $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(\sigma^2,\sigma^2)}$, $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(d,d)}$, and $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(\sigma^2,d)}$, respectively. To derive these three quantities, it is necessary to perform *eigenvalue decompositions* of $\partial \mathbf{C}(\boldsymbol{\xi})/\partial \sigma^2$ and $\partial \mathbf{C}(\boldsymbol{\xi})/\partial d_m$ via $\tilde{\mathbf{Q}}$, which leads to the following representation:

$$\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \sigma^2} = \tilde{\mathbf{Q}}^H \begin{bmatrix} T\mathbf{D}\mathbf{D}^H & & \\ & \mathbf{0} & \\ & & \ddots \\ & & & \mathbf{0} \end{bmatrix} \tilde{\mathbf{Q}}, \tag{14}$$

and

$$\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial d_m} = \tilde{\mathbf{Q}}^H \begin{bmatrix} \sigma^2 \widetilde{\mathbf{D}}_m T & & \\ & \mathbf{0} & \\ & & \ddots \\ & & & \mathbf{0} \end{bmatrix} \tilde{\mathbf{Q}}. \tag{15}$$

Leveraging the *unitary* property of $\tilde{\mathbf{Q}}$ and the *cyclic* property of the $\mathrm{Tr}[\cdot]$ operation,

$$\begin{aligned}[\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}]_{\sigma^2,\sigma^2} &= \mathrm{Tr}\left[\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \sigma^2}\mathbf{C}^{-1}(\boldsymbol{\xi})\frac{\partial \mathbf{C}(\boldsymbol{\xi})}{\partial \sigma^2}\mathbf{C}^{-1}(\boldsymbol{\xi})\right] \\ &= \sum_{m=1}^{M} \frac{T^2 d_m^4}{(\sigma^2 T d_m^2 + N_0)^2}.\end{aligned} \tag{16}$$

Following a similar methodology, one can compute

$$[\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}]_{d_m,d_k} = \frac{4\sigma^4 T^2 d_m^2}{(\sigma^2 T d_k^2 + N_0)^2}\delta_{mk}, \tag{17}$$

and

$$[\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}]_{\sigma^2,d_m} = \frac{2\sigma^2 T^2 d_m^3}{(\sigma^2 T d_m^2 + N_0)^2}, \tag{18}$$

where $\delta_{mk} = 1$ only if $m = k$. Due to space constraints, we avoid presenting the full calculation of (17) and (18), respectively. Following this, we derive $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}$. We begin by taking the derivative of $\boldsymbol{\mu}(\boldsymbol{\xi})$ with respect to elements in $\boldsymbol{\xi}$. Doing this yields the following results

$$\begin{aligned}
&\frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial \gamma} = \mathbf{1} \otimes \mathbf{D}\,\mathbf{a}(\phi), \quad \frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial d_m} = \mathbf{1} \otimes e^{j\alpha_m}\mathbf{E}_{mm}\gamma\,\mathbf{a}(\phi), \\
&\frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial \sigma^2} = 0, \text{ and } \frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial \alpha_m} = \mathbf{1} \otimes j d_m e^{j\alpha_m}\mathbf{E}_{mm}\gamma\,\mathbf{a}(\phi).
\end{aligned} \tag{19}$$

Note that $\mathbf{E}_{mm}$ is the elementary matrix which has unit value only at the intersection of the $m-$th row and $m-$th column, and zeros elsewhere. In accordance with (19), it is trivial that $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\sigma^2,d)}$, $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\sigma^2,\alpha)}$, $[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{\sigma^2,\sigma^2}$, and $[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{\sigma^2,\gamma}$ are all 0, since the first two quantities are zero vectors, while the second two quantities are zero scalars. To derive the remaining sub-matrices of $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}$, we express a unit vector as $\mathbf{1} = \sqrt{T}\mathbf{Q}^H\boldsymbol{\eta}$, where $\boldsymbol{\eta}$ denoted as a $T \times 1$ column vector $[1, 0, \ldots, 0]^T$. Based on the properties of the unitary matrix and the mixed-product property of the kronecker operation, we can express $[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{d_m,d_k}$ as

$$\begin{aligned}
[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{d_m,d_k} &= 2\Re\left[\frac{\partial \boldsymbol{\mu}^H(\boldsymbol{\xi})}{\partial d_m}\,\mathbf{C}^{-1}(\boldsymbol{\xi})\,\frac{\partial \boldsymbol{\mu}(\boldsymbol{\xi})}{\partial d_k}\right] \\
&\overset{(a)}{=} 2\Re\left[\boldsymbol{\kappa}^H(\mathbf{Q}\otimes\mathbf{I}_M)^{-1}\boldsymbol{\Lambda}^{-1}(\mathbf{Q}\otimes\mathbf{I}_M)\boldsymbol{\kappa}\right] \\
&= \frac{2T}{N_0 + \sigma^2 T d^2 m}\gamma^2\delta_{mk},
\end{aligned} \tag{20}$$

where (a) contains $\boldsymbol{\kappa} = \mathbf{Q}^H\boldsymbol{\eta} \otimes \gamma e^{j\alpha_m}\mathbf{E}_{mm}\,\mathbf{a}(\phi)$. Following the same method, we can derive the rest of sub-matrices $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}$. Due to space limitation, we avoid presenting the exact calculations, however we quote the final results below:

$$[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{\gamma,\gamma} = \sum_{m=1}^{M}\frac{2d_m^2 T}{\sigma^2 T d_m^2 + N_0}, \tag{21}$$

$$[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{d_m,\gamma} = \frac{2T d_m \gamma}{\sigma^2 T d_m^2 + N_0}, \tag{22}$$

$$[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{\alpha_m,\alpha_k} = \frac{2T d_m^2\gamma^2}{N_0 + \sigma^2 T d_m^2}\delta_{mk}, \tag{23}$$

$$[\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{\gamma,\alpha_k} = 0 \text{ and } [\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}]_{d_m,\alpha_k} = 0. \tag{24}$$

$$\mathbf{I}(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(d,d)} + \mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(d,d)} & \mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(\sigma^2,d)} & \mathbf{0} & \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(d,\gamma)} \\ \mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}^{H}|_{(\sigma^2,d)} & \mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(\sigma^2,\sigma^2)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\alpha,\alpha)} & \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\alpha,\gamma)} \\ \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}^{H}|_{(d,\gamma)} & \mathbf{0} & \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}^{H}|_{(\alpha,\gamma)} & \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\gamma,\gamma)} \end{bmatrix}. \tag{25}$$

Adding $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}$ with $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}$, the *closed-form* FIM $\mathbf{I}(\boldsymbol{\xi})$ is given by (25), presented on top of the following page.

B. Inverse of FIM

To compute the CRLB, one is generally required to *invert* the FIM. We check the invertability of $\mathbf{I}(\boldsymbol{\xi})$ by computing its determinant numerically, and ensuring that the result is non-zero. In order to perform parameter estimation for absolute calibration, we are only interested in the following two terms of the FIM: $\mathbf{I}(\boldsymbol{\xi})_{d,d}^{-1}$ and $\mathbf{I}(\boldsymbol{\xi})_{\alpha,\alpha}^{-1}$. *This is since only these terms contain the necessary information for the amplitude scaling and phase shifts associated with each RF chain. The other terms do not need to be inverted, since they contain information relating to γ and σ^2 which denote the LOS power and power of the diffuse multipath components which do not need to be estimated.* Enlightened by this, we provide the following analysis which begins by splitting $\mathbf{I}(\boldsymbol{\xi})$ into four parts for mathematical convenience. Specifically,

$$\mathbf{I}(\boldsymbol{\xi}) = \begin{bmatrix} \mathbf{X} & \boldsymbol{\psi} \\ \boldsymbol{\psi}^H & w \end{bmatrix}, \tag{26}$$

where the scalar $w = \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\gamma,\gamma)}$, the vector $\boldsymbol{\psi}$ is given by $(\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}^{T}|_{(d,\gamma)} \;\; \mathbf{0}_{1\times M} \;\; \mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}^{T}|_{(\alpha,\gamma)})^T$ and $\mathbf{X}$ for the rest of $\mathbf{I}(\boldsymbol{\xi})$. Leveraging the relationship between the *adjugate matrix* and the *inversion matrix*, $\mathbf{I}^{-1}(\boldsymbol{\xi})$ can be expressed as

$$\mathbf{I}^{-1}(\boldsymbol{\xi}) = \frac{\mathbf{I}^{\dagger}(\boldsymbol{\xi})}{\det(\mathbf{I}(\boldsymbol{\xi}))}, \tag{27}$$

where $\mathbf{I}^{\dagger}(\boldsymbol{\xi})$ is the adjugate matrix, which can be obtained by extracting the resulting sub-matrix after striking out the $i-$th row and column of $\mathbf{I}(\boldsymbol{\xi})$. Since only the diagonal elements of $\mathbf{X}$ contain the phase shift and amplitude scaling estimation parameters of interest, the range of $i = 1, 2, \ldots, M, M+2, M+3, ..., 2M+1$. Then, by applying the definition of adjugate matrix and *Schur complement*, one can calculate the $i-$th diagonal element of $\mathbf{I}^{-1}(\boldsymbol{\xi})$ as [30]

$$\mathbf{I}^{-1}(\boldsymbol{\xi})_{ii} = \frac{\det(\widetilde{\mathbf{X}}_{ii})}{\det(\mathbf{X})} \left(\frac{w - \boldsymbol{\eta}_i^H \widetilde{\mathbf{X}}_{ii}^{-1} \boldsymbol{\eta}_i}{w - \boldsymbol{\eta}^H \mathbf{X}^{-1} \boldsymbol{\eta}} \right), \tag{28}$$

where $\widetilde{\mathbf{X}}_{ii}$ can be obtained $\mathbf{X}$ by striking the $i-$th row and column, while $\boldsymbol{\eta}_i$ can be extracted from the column vector $\boldsymbol{\eta}$ by striking the i-th row. We now present the full analytical form of $\mathbf{I}^{-1}(\boldsymbol{\xi})_{ii}$: For convenience, we let vectors $\boldsymbol{\phi} \in \mathbb{R}^M$, $\boldsymbol{\zeta} \in \mathbb{R}^M$ denote the diagonal elements of the matrices $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(d,d)} + \mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(d,d)}$ and $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(\alpha,\alpha)}$ respectively. Furthermore, we let $\boldsymbol{\varphi} \in \mathbb{R}^M$ and $\boldsymbol{\vartheta} \in \mathbb{R}^M$ represent vectors $\mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(\sigma^2,d)}$ and $\mathbf{I}(\boldsymbol{\xi})_{\boldsymbol{\mu}}|_{(d,\gamma)}$ respectively. In addition, we define scalars

$\rho = \mathbf{I}(\boldsymbol{\xi})_{\mathbf{C}}|_{(\sigma^2,\sigma^2)}$, $\chi_i = \det(\widetilde{\mathbf{X}}_{ii})/\det(\mathbf{X})$ and $\chi_i' = (w - \boldsymbol{\eta}_i^H \widetilde{\mathbf{X}}_{ii}^{-1}\boldsymbol{\eta}_i)/(w - \boldsymbol{\eta}^H \mathbf{X}^{-1}\boldsymbol{\eta})$. With the aid of *Gaussian elimination*, we can calculate χ_i as [30]

$$\chi_i = \begin{cases} \dfrac{\rho - \sum_{j\neq i}^{M} \frac{\boldsymbol{\varphi}_j^2}{\boldsymbol{\phi}_j}}{\boldsymbol{\phi}_i(\rho - \sum_{j=1}^{M} \frac{\boldsymbol{\varphi}_j^2}{\boldsymbol{\phi}_j})}; & i = 1, 2, ..., M \\ \dfrac{1}{\boldsymbol{\zeta}_i}; & i = M+2, M+3, ..., 2M+1. \end{cases} \tag{29}$$

Using block matrix inversion theorem [30],

$$\chi_i' = \begin{cases} \dfrac{w - \left(\sum_{j\neq i}^{M} \frac{\boldsymbol{\vartheta}_j^2}{\boldsymbol{\phi}_j}\right) - f_2^{-1}\left(\sum_{j\neq i}^{M} \frac{\boldsymbol{\vartheta}_j \boldsymbol{\varphi}_j}{\boldsymbol{\phi}_j}\right)^2}{w - \left(\sum_{j=1}^{M} \frac{\boldsymbol{\vartheta}_j^2}{\boldsymbol{\phi}_j}\right) - f_1^{-1}\left(\sum_{j=1}^{M} \frac{\boldsymbol{\vartheta}_j \boldsymbol{\varphi}_j}{\boldsymbol{\phi}_j}\right)^2} \\ \qquad\qquad ; i = 1, 2, ..., M \\ 1 \quad ; i = M+2, M+3, ..., 2M+1, \end{cases} \tag{30}$$

where scalrs f_1 and f_2 are defined as $f_1 = \rho - \sum_{j=1}^{M}(\boldsymbol{\varphi}_j^2/\boldsymbol{\phi}_j)$, $f_2 = \rho - \sum_{j\neq i}^{M}(\boldsymbol{\varphi}_j^2/\boldsymbol{\phi}_j)$, respectively.

Note that this is a very general solution to a complex problem which holds for any SNR value and any number of receive antennas at the BS. An interesting special case of (30) can be analyzed, which is as follows: Supposing that the system is operated in the high SNR regime, implying that N_0 is much *smaller* than $\sigma^2 T d_i^2$, when $1 \leq i \leq M$, χ_i' can be approximated as:

$$\chi_i' \approx \frac{\varepsilon\left[\sigma^2 M(2\varepsilon - \gamma^2) + 2\sigma^4 + \gamma^2\varepsilon + 2\gamma^2\sigma^2\right]}{\sigma^2(2\varepsilon - \gamma^2)(M\varepsilon + \sigma^2)}, \tag{31}$$

where ε is defined as $\varepsilon = 2T\gamma^2 + (4T^2 - 1)\sigma^2$. If M is much larger than σ^2, which is typically the case for massive MIMO systems, then χ_i' can be approximated as 1 since the numerator and denominator of (31) both scale linearly with M resulting in a cancellation. Relative to M, the other variables do not significantly influence the result of (31) and hence are less dominant. Based on (28)-(30), in high SNR conditions, for a massive MIMO system, the diagonal elements of $\mathbf{I}^{-1}(\boldsymbol{\xi})$ can be revealed in a rather elegant form. which demonstrate the CRLBs of the amplitude and phase estimations. These are

$$\mathbf{I}^{-1}(\boldsymbol{\xi})_{ii} \approx \begin{cases} \dfrac{\sigma^2 d_i^2}{2(\gamma^2 + 2\sigma^2)} & ; i = 1, 2, ..., M \\ \dfrac{\sigma^2}{2\gamma^2} & ; i = M+2, M+3, ..., 2M+1. \end{cases} \tag{32}$$

From (32)*, the CRLB of the phase estimation is directly proportional to σ^2 and inversely proportional to γ^2. As a special case, if the system is operating with pure non LOS propagation environment (i.e., $\gamma = 0$), the CRLB relating to phase drifts goes to infinity, and it is impossible to estimate the phase drifts in this situation. However, for UE-aided absolute calibration, it is common that the UE will be in close proximity to the BS and hence will almost surely have a dominant LOS component, along with other multipath components. In contrast, for the amplitude*

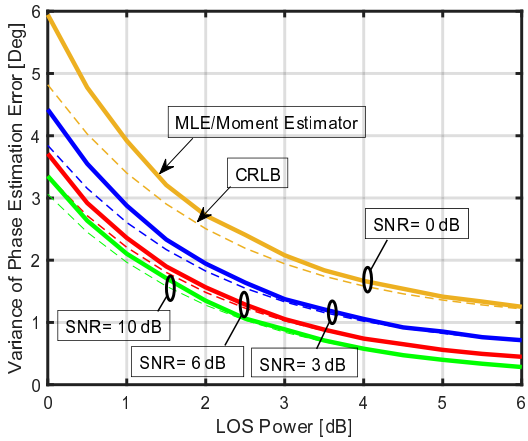


Fig. 2: MLE (and moment-based) estimator performance as a function of LOS powers for phase calibration with varying SNRs. The phase estimation CRLBs are shown for comparison purposes.

estimations, both γ and σ^2 can contribute to the inverse of the FIM. Therefore, it is possible to find a soluable estimator, even when there is no LOS component.

V. Numerical Results

The ultimate aim of our work is to implement the aforementioned calibration parameter estimation techniques into a real-time massive MIMO testbed. To this end, as a first step in this direction, we evaluate the estimation performance via Monte-Carlo simulations. Our simulation framework caters to a 100 element ULA connected to 100 individual RF chains. We assume that the physical distance between the electrical phase centers of successive antenna elements is $d = \lambda_f/2$, where λ_f is the wavelength corresponding to the operating carrier frequency. Considering this, the overall steering vector can be written as

$$\mathbf{a}\left(\phi\right) = \left[1, e^{-j2\pi d\cos(\phi)} \ldots, e^{-j2\pi d(M-1)\cos(\phi)}\right]. \tag{33}$$

Consistent with [14], the *ground truth* of the *magnitudes* of the RF chain coefficients are assumed to be unity, while the *phases* are assumed to be distributed uniformly between $[-\pi, \pi]$. This serves as a basis for comparison for the estimated amplitudes and phases. Furthermore, the SNR is given by

$$\begin{aligned}\text{SNR} &= \frac{\mathbb{E}\{(\gamma\mathbf{Da}(\phi)p + p\mathbf{Dh})(\gamma\mathbf{Da}(\phi)p + \mathbf{Dh}p)^H\}}{E\{\mathbf{n}\mathbf{n}^H\}} \\ &= \frac{\sum_{m=1}^{M} d_m^2(\sigma^2+\gamma^2)}{MN_0},\end{aligned} \tag{34}$$

where the respective quantities in (34) are defined in Sec. II. To manage the computational run-time of the numerical simulations, while observing data for a long enough time period, we

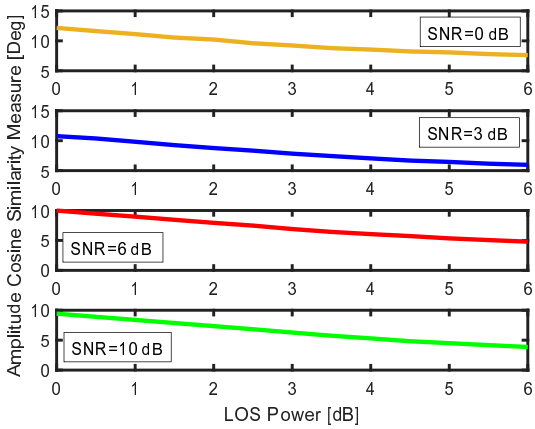


Fig. 3: Cosine similarity measure as a function of the LOS powers for amplitude estimation using the moment-based estimator. Variability in SNR is also presented.

set the total observation duration of the received vector as $T = 3$, for each 10 independent and identically distributed Monte-Carlo realizations are simulated. To evaluate the accuracy of phase and amplitude estimations, we hereby assume that the groundtruths of the phase drifts and amplitude scalings of each RF chain are generated by the simulation framework, and stored for the sake of comparison. In Fig. 2, we first present the performance of *both* types of *phase* estimators (mathematically proven to have same form), with the derived CRLB for phase estimation. We do this by reporting the variance of the phase MLE estimators over the 100 parallel RF chains against the derived CRLB under different SNRs and LOS powers (γ) factors. The CRLB of the phase estimation is inversely proportional to γ^2 (see (23)), thus the CRLB and the MLE estimator decay exponentially as γ increases. As shown, the variances of the MLE estimator approaches the CRLB as γ or the SNR increases. The results coincide with the MLE behaviour since the increase of either the SNR or LOS power reduces the phase variance, leading to a small estimation error and the MLE is therefore able to achieve its asymptotic probability distribution function [29]. Naturally, we would expect the moment-based estimator to have the same performance via (6) and (9).

Figure 3 presents the estimation result of the linear space spanned by the amplitude vector $\mathbf{d} = [d_1, d_2, ..., d_M]^T$ using the moment-based estimator. Using the *cosine similarity measure* [31], we define the criteria for measuring the angular difference between the estimated vector, $\hat{\mathbf{d}}$, and the groundtruth, $\mathbf{d}$, as

$$\text{cos. sim.} = \arccos\left(\frac{|\hat{\mathbf{d}}^H \mathbf{d}|}{\|\hat{\mathbf{d}}\| \, \|\mathbf{d}\|}\right), \tag{35}$$

where $\|\cdot\|$ denotes the vector norm. Based on (35), a perfectly estimated vector is aligned with the groundtruth vector, i.e., $\cos.\text{sim.} = 0$, while the worst estimate is a vector in perpendicular

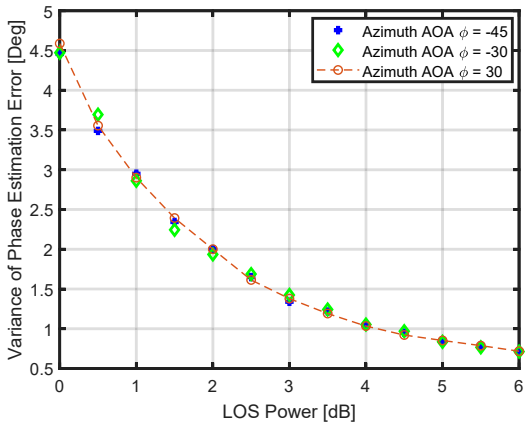


Fig. 4: MLE (and moment-based) estimator performance for different LOS powers and dominant LOS AOAs (in degrees) with SNR=3 dB.

direction to the groundtruth vector resulting in $\cos.\text{sim.} = 1$. The moment-based estimator in (10) exploits the first and second-order statistics by calculating the expectation of the received signal and reconstructing the covariance matrix. The increase of LOS power yields more superior reconstruction quality and therefore improves the estimation result. Also, as shown in Fig. 3, an increase in the SNR results in higher estimator accuracy as the estimated amplitude starts to converge towards the groundtruth. In addition, it is challenging to evaluate the CRLB of (35), since the inversion of the whole FIM is required, which is an extremely difficult task [29]. Therefore, we defer this to the upcoming journal version of the paper.

Figure 4 depicts the phase estimation results of the two estimators as a function of LOS powers, for different dominant LOS AOAs at SNR=3 dB. It can readily be observed that the resulting phase estimates for different AOAs are essentially the same. This is since each element of the steering vector has a constant amplitude of one for all of the incoming angles between $[-\pi/2, \pi/2]$. To this end, the change of dominant AOAs will have no influence on the estimator performance for a given LOS power and a given SNR. Although not shown here, the same trends hold for the moment-based amplitude estimator following the same phenomena.

VI. Conclusion

In this paper, we consider UE-aided amplitude and phase estimation for absolute calibration of massive MIMO front-ends. Assuming a Ricean fading channel model, for a single-user massive MIMO system, we analyze the performance of moment-based and MLE estimators for estimating the relative amplitude scalings and phase shifts associated with each RF chain. Our analysis assumes no knowledge of the LOS power, diffuse multipath component power, amplitudes and phase shifts. The derived estimators only need knowledge of the received array's steering vectors and transmitted pilots by the UE. We mathematically prove that for phase estimation, both MLE and moment-based estimators have the same form, and hence perform equally well. To

evaluate the performance of respective estimators, we investigate the CRLB via the analysis of the FIM, where we draw several important insights. We show that the presence of a dominant LOS component is mandatory for phase drift estimation, while not necessary for the amplitude estimation. Our numerical results indicate that the variance of the phase estimates converge to the corresponding CRLBs with increasing LOS powers and SNRs. Likewise, the amplitude estimation accuracy improves substantially with increasing SNR. Different dominant LOS AOA tends to make almost no difference to the phase and amplitude estimator performance, since the amplitude of each entry in the steering vector remains constant over all considered angles. In the future, our aim is to implement the estimators for absolute calibration on the Lund University massive MIMO testbed.

ACKNOWLEDGMENT

The authors gratefully acknowledge Prof. Bo Bernhardsson, A/Prof. Fredrik Rusek and Mr. Tianyi Bai. The support of Ericsson AB is also acknowledged for funding this work.

REFERENCES

[1] Ericsson, "Ericsson Mobility Report," June 2019, Accessible online at https://www.ericsson.com/49d1d9/assets/local/mobility-report.

[2] 3GPP TR 38.104, "Technical Specification on base station (BS) radio transmission and reception," v16.1.0, Sep. 2019.

[3] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *z*, vol. 9, no. 11, pp. 3590-3600, Nov. 2010.

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun.Mag.*, vol. 52, no. 2, pp. 186-195, Feb. 2014.

[5] H. Yang and T. L. Marzetta, "Performance of conjugate and zero-forcing beamforming in large-scale antenna systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 172-179, Feb. 2013.

[6] X. Gao, O. Edfors, F. Tufvesson, and E. G. Larsson, "Massive MIMO in real propagation environments: Do all antennas contribute equally?," *IEEE Trans. Commun.*, vol. 63, no. 11, pp. 3917-3928, Nov. 2015.

[7] M. Shafi, *et al.*, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201-1221, Jun. 2017.

[8] M. Shafi, *et al.*, "Microwave vs. millimeter-wave propagation channels: Key differences and impact on 5G cellular systems," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 14-20, Dec. 2018.

[9] A. F. Molisch, *et al.*, "Hybrid beamforming for massive MIMO: A survey," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 134-141, Sep. 2017.

[10] M. Guillaud, D. T. M. Slock and R. Knopp, "A practical method for wireless channel reciprocity exploitation through relative calibration," *Proc. 8th Int. Symp. Signal Processing and Its Applications (ISSPA)*, 2005., Sydney, Australia, 2005, pp. 403-406.

[11] R. Rogalin, *et al.*, "Scalable synchronization and reciprocity calibration for distributed multiuser MIMO," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 1815-1831, Apr. 2014.

[12] J. Vieira, F. Rusek, and F. Tufvesson, "Reciprocity calibration methods for massive MIMO based on antenna coupling," *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2014, pp. 3708-3712.

[13] J. Vieira, F. Rusek, O. Edfors, S. Malkowsky, L. Liu, and F. Tufvesson, "Reciprocity calibration for massive MIMO: Proposal, modeling, and validation," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 3042-3056, May 2017.

[14] X. Jiang *et al.*, "A framework for over-the-air reciprocity calibration for TDD massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5975-5990, Sept. 2018.

[15] B. C. Ng and C. M. S. See, "Sensor-array calibration using a maximum-likelihood approach," *IEEE Trans. Antennas Propag.*, vol. 44, no. 6, pp. 827-835, Jun. 1996.

[16] H. Liu, L. Zhao, Y. Li, X. Jing, and T. Truong, "A sparse-based approach for DOA estimation and array calibration in uniform linear array," *IEEE Sensors J.*, vol. 16, no. 15, pp. 6018-6027, Aug. 2016.

[17] X. Luo, F. Yang, and H. Zhu, "Massive MIMO self-calibration: Optimal interconnection for full calibration," *IEEE Trans. Veh. Technol.* (early access), Sep. 2019.

[18] A. F. Molisch, *Wireless Communications*, IEEE Wiley Press, 2011.

[19] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.

[20] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.

[21] A. Benzin and G. Caire, "Internal self-calibration methods for large scale array transceiver software-defined radios," *21th International ITG Workshop on Smart Antennas (WSA)*, Mar. 2017, pp. 49-56.

[22] I. Şeker, "Calibration methods for phased array radars," in *SPIE 8714, Radar Sensor Technology XVII, 87140W*, May. 2013.

[23] Z. Liu, "Conditional Cramer–Rao lower bounds for DOA estimation and array calibration," *IEEE Signal Process. Lett.*, vol. 21, no. 3, pp. 361-364, Mar. 2014.

[24] H. Wei, D. Wang, H. Zhu, J. Wang, S. Sun, and X. You, "Mutual coupling calibration for multiuser massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 606-619, Jan. 2016.

[25] H. Tataria, P. J. Smith, L. J. Greenstein, P. A. Dmochowski, and M. Matthaiou, "Impact of line-of-sight and unequal spatial correlation on uplink MU-MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 6, no. 5, pp. 634-637, Oct. 2017.

[26] H. Tataria, P. J. Smith, L. J. Greenstein, and P. A. Dmochowski, "Zero-forcing precoding performance in multiuser MIMO systems with heterogeneous Ricean fading," *IEEE Wireless Commun. Lett.*, vol. 6, no. 1, pp. 74-77, Nov. 2016.

[27] S. Wesemann, H. Schlesinger, A. Pascht, and O. Blume, “Measurement and characterization of the temporal behavior of fixed massive MIMO links,” in *21th International ITG Workshop on Smart Antennas (WSA)*, Mar. 2017, pp. 135-142.

[28] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.

[29] S. M. Kay, *Fundamentals of Statistical Processing Vol. I: Estimation Theory*, Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[30] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[31] M. Loog, ”On Distributional assumptions and whitened cosine similarities,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1114-1115, Jun. 2008.

Paper III

Paper III

Reproduced, with permission, from: IEEE

G. Tian, X. Cai, T. Zhou, Weinan. Wang, and F. Tufvesson, "Deep-Learning Based Channel Estimation for OFDM Wireless Communications," *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, Oulu, Finland, 2022, pp. 1-5, doi: 10.1109/SPAWC51304.2022.9834008.

Deep-Learning Based Channel Estimation for OFDM Wireless Communications

Guoda Tian[†], Xuesong Cai[†], Tian Zhou[‡], Weinan Wang[*], Fredrik Tufvesson[†]
[†]Department of Electrical and Information Technology, Lund University, Sweden
[‡]Shenzhen KC High-end Equipment Technology Research and Development Co., Ltd, China
[*]Shenzhen Metalligence Technology Co., Ltd, China
Emails: {guoda.tian, xuesong.cai, fredrik.tufvesson}@eit.lth.se, emconsulting@foxmail.com, weinanwang@metalligence.com.cn

Abstract

Multi-carrier technique is a backbone for modern commercial networks. However, the performances of multi-carrier systems in general depend greatly on the qualities of acquired channel state information (CSI). In this paper, we propose a novel deep-learning based processing pipeline to estimate CSI for payload time-frequency resource elements. The proposed pipeline contains two cascaded subblocks, namely, an initial denoise network (IDN), and a resolution enhancement network (REN). In brief, IDN applies a novel two-step denoising structure while REN consists of pure fully-connected layers. Compared to existing works, our proposed processing architecture is more robust under lower signal-to-noise ratio scenarios and delivers generally a significant gain.

Index Terms

Deep learning, channel estimation, OFDM, channel state information.

I. Introduction

Multi-carrier systems have been a fundamental radio access technology, e.g., as standardized in Third Generation Partnership Project (3GPP), for more than a decade [1]. One representative technology is the orthogonal frequency division multiplexing (OFDM), which was firstly invented by the Bell Labs back in the 1960s. Owing to its excellent properties on mitigating multi-path fading and increasing network spectral efficiency, OFDM has been supporting the modern fourth generation (4G) and 5G commercial networks for serveral decades and still has a strong potential for future 6G wireless communications [2], [3].

The performance of a multi-carrier system hinges greatly on the knowledge of channel state information (CSI) at each time-frequency resource grid. Usually, known signals, so-called pilot signals, are transmitted to estimate the CSI at certain time-frequency locations as an initial step, and afterwards CSI at other resource locations for payload communications can be obtained according to the acquired pilot CSI. To improve the accuracy of estimated payload CSI, one can intuitively boost transmission powers of pilot signals or insert more pilots onto the time-frequency resource grids. However, the power efficiency or spectrum efficiency of the system

would be sacrificed correspondingly. Moreover, adjacent cells may run a risk of being strongly interfered, resulting in pilot contamination [4]–[6]. Therefore, a favorable method is to exploit the correlation properties embedded in both frequency and time domains to improve the accuracy of estimated CSI without modifying the configuration of pilot signals.

Available algorithms in the literature mainly fall into two categories, namely, traditional signal processing based and machine learning based approaches. The principle of linear minimum mean-square error (LMMSE), as a classical signal processing based method, relies on prior knowledge of the first-order and second-order statistics of the channels [7], [8]. However, the performance of LMMSE based methods may have a bottleneck due to the underlying linear assumption. In contrast, data-driven methods that can explore non-linear properties are expected to achieve more promising results [9]–[13]. Specifically, [9] presented an initial attempt to apply deep learning tools to jointly enhance CSI estimation and symbol detection for OFDM systems. The authors in [10] proposed a novel neural network for decoding OFDM signals, which combined deep learning methods and knowledge of wireless communication systems. The works presented in [11], [12] borrowed techniques from image processing fields and applied the so-called "image super resolution" algorithms to enhance channel estimation results. In particular, the channel estimation framework "Channelnet" proposed by [11] achieves a significant gain compared with traditional LMMSE method. In addition, the authors in [13] leveraged convolutional neural networks (CNN) to strengthen the conventional LMMSE receiver for multiple user MIMO systems. However, despite the promising results achieved by previous work, it is still, to our best knowledge, challenging for existing channel estimation algorithms to deliver a solid performance under low SNR scenarios. In addition, current deep-learning based channel estimation approaches are mainly based on CNN family, which generally take much longer time to train and therefore to apply them in practical systems.

To address these issues, we propose a novel deep learning based pipeline, which contains two processing blocks with pure fully-connected layers, i.e., an initial denoise network (IDN) and a resolution enhancement network (REN). The IDN aims at improving the SNR of noisy pilots via a novel denoising structure. After this noise reduction operation, we feed the output of IDN into a fully-connected REN, which aims to recover all payload CSI based on the knowledge of pilot CSI. Results show that compared with both traditional method LMMSE as well as *ChannelNet*, the proposed pipeline can achieve a significant gain of around 10-15 dB both under low and high SNR scenarios.

The rest of the paper is structured as follows. We formulate the channel estimation problem in section II, and the proposed method is illustrated in section III. Section IV demonstrates the performance gain of the proposed method compared to existing works by using synthetic channel data. Finally, conclusions are given in section V.

II. Problem Formulation

As illustrated in Fig. 1, we hereby consider a single-input-single-output (SISO) OFDM system with F subcarriers and T symbols, i.e., FT resource elements (REs). The received signal $Y(f,t) \in \mathbb{C}$ at the f-th subcarrier and t-th symbol can be written as

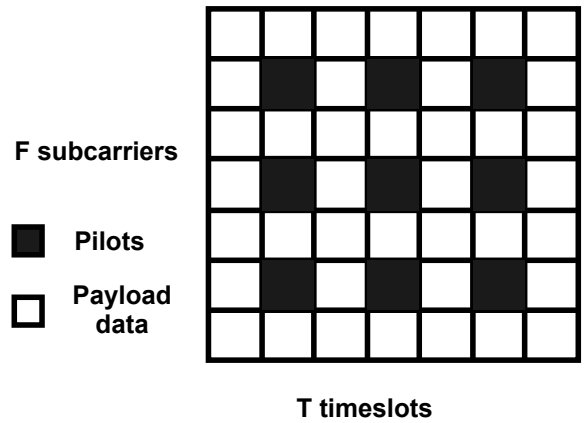


Fig. 1: Examples of pilot configurations in a typical OFDM system

$$Y(f,t) = H(f,t)X(f,t) + N(f,t), \tag{1}$$

where $H(f,t) \in \mathbb{C}$ and $X(f,t) \in \mathbb{C}$ represent the complex CSI and transmitted signal at this RE, respectively, and $N(f,t) \in \mathbb{C}$ denotes additive white gaussian noise (AWGN) with zero mean and variance $\frac{N_o}{2}$. Accurate knowledge of CSI at individual REs is essential for demodulating transmitted symbols. For this purpose, P known pilot signals, as indicated by the black squares in Fig. 1, are transmitted to sound the CSI of the P pilot REs. We concatenate them as a vector $\tilde{\mathbf{h}}_x \in \mathbb{C}^{P\times 1}$. Note that $\tilde{\mathbf{h}}_x \in \mathbb{C}^{P\times 1}$ is a noisy version of the true pilot CSI $\mathbf{h}_x \in \mathbb{C}^{P\times 1}$ due to the existence of noise in the received pilot signals. The problem we focus in this paper is then *how to estimate CSI of all (the other) REs, i.e.,* $\mathbf{H}_y \in \mathbb{C}^{F\times T}$*, according to* $\tilde{\boldsymbol{h}}_x$. Before elaborating our proposed method, we briefly review two representative methods already existing in the literature, namely LMMSE estimator [14] and ChannelNet [11].

A. LMMSE

The conventional LMMSE estimator belongs to Bayesian family and is constructed by a linear combination of the observed data. The combination matrix, which exploits the first and second-order statistics, is chosen to minimize the mean square error of the estimation result [14]. Specifically, the estimator of the CSI $\mathbf{h}_y \in \mathbb{C}^{TF\times 1}$ can be obtained as

$$\tilde{\mathbf{h}}_y = \mathrm{E}(\mathbf{h}_y) + \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\left[\tilde{\mathbf{h}}_x - \mathrm{E}(\tilde{\mathbf{h}}_x)\right], \tag{2}$$

where $\mathbf{h}_y$ is the vectorization of $\mathbf{H}_y$, i.e., $\mathbf{h}_y \triangleq \mathrm{vec}(\mathbf{H}_y)$, $\mathrm{E}(\cdot)$ represents the expectation operation, $\mathbf{C}_{yx}$ is the cross-correlation matrix of $\mathbf{h}_y$ and $\tilde{\mathbf{h}}_x$, and $\mathbf{C}_{xx}$ is the auto-correlation matrix of $\tilde{\mathbf{h}}_x$.

B. Channelnet approach

Channelnet was proposed in [11]. The whole network consists of two cascaded processing blocks, namely, the super resolution network (SRN) and the image restoration network (IRN).

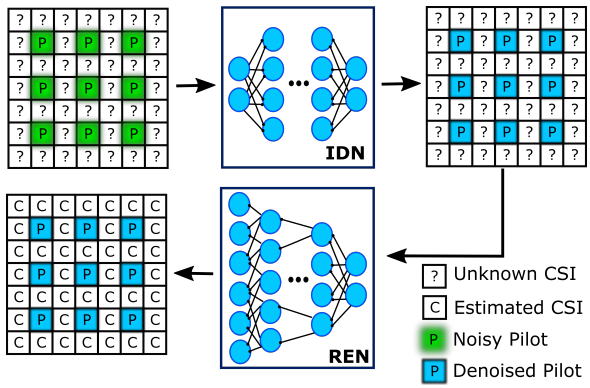


Fig. 2: Our proposed channel estimation pipeline.

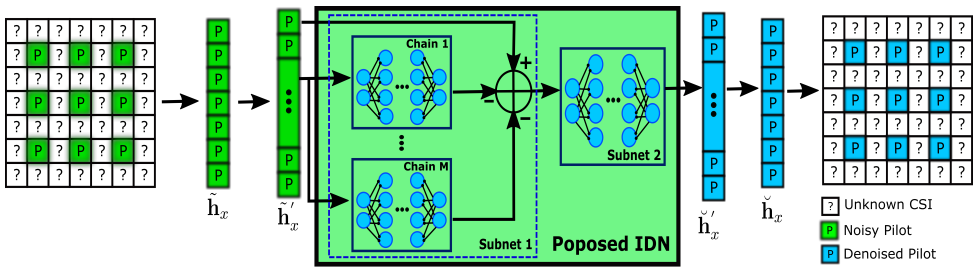


Fig. 3: The architecture of the initial denoise network.

Specifically, SRN takes the low resolution and noisy pilot CSI as input, resulting in a coarse estimation result. Then the coarse results are forwarded to the IRN to further enhance their SNRs. According to [11], CNN is selected as the core technique for both two blocks.

III. Structures of Proposed Networks

The pipeline of our proposed channel estimation algorithm is presented in Fig. 2. As illustrated, the proposed structure consists of two cascaded subblocks, i.e., the IDN and REN. Briefly, IDN takes corrupted pilot signal vector as input and aims at reconstructing the signal vector itself with much less noise. REN has a fully-connected neural network as the core, estimating CSI for all payload datas. In addition, the SNR label is not used during the training phase since we aim at training a generalized model for all possible SNR points.

A. Initial Denoise Network

We first deploy an IDN to improve the SNR of $\tilde{\mathbf{h}}_x$ using supervised learning, resulting in an improved pilot CSI $\breve{\mathbf{h}}_x \in \mathbb{C}^{P\times 1}$. Note that since the network aims to approach the noisy-free pilot CSI as much as possible, groundtruth $\mathbf{h}_x$'s are exploited to supervise the training process.

The architecture of our designed IDN is illustrated in Fig. 3. As shown, the network reshapes the vector $\tilde{\mathbf{h}}_x$ as a new vector $\tilde{\mathbf{h}}'_x \in \mathbb{R}^{2P\times 1} \triangleq [\mathrm{Re}(\tilde{\mathbf{h}}_x)^T, \mathrm{Im}(\tilde{\mathbf{h}}_x)^T]^T$ as input, while generates $\breve{\mathbf{h}}'_x \in \mathbb{R}^{2P\times 1} = [\mathrm{Re}(\breve{\mathbf{h}}_x)^T, \mathrm{Im}(\breve{\mathbf{h}}_x)^T]^T$ as a denoising result. The overall network structure is made up of two subnets. The first subnet consists of a number of M parallel branches of processing chains, each as a fully-connected neural network. By subtracting the input signal $\tilde{\mathbf{h}}'_x$ itself with the summation of all those processing, we collect the output signal of the first subnet. As a next step, we feed this output signal into the second subnet which consists of fully connected NN, generating vector $\breve{\mathbf{h}}'_x$ with enhanced SNR. We define $\mathbf{f}_{\text{IDN}}(.)$ as the overall function for the entire proposed denoise network, $\boldsymbol{\theta}_{\text{IDN}}$ as a vector collecting all hyper parameters and Ω as all training datasets. Based on this definition, the overall loss function C_1 can be written as

$$C_1 = \sum_{\tilde{\mathbf{h}}'_x \in \Omega} ||\mathbf{f}_{\text{IDR}}(\tilde{\mathbf{h}}'_x, \boldsymbol{\theta}_{\text{IDN}}) - \mathbf{h}'_x||_F^2, \tag{3}$$

where $||.||_F$ represents the Frobenius norm. During the training phase, the hyperparameter vector $\boldsymbol{\theta}_{\text{IDN}}$ is optimized in order to minimize cost function C_1.

B. Resolution Enhancement Network

The REN acts as an important block of our proposed processing pipeline, which provides channel coefficients estimation for the whole time-frequency resource grid via a supervised learning approach. The structure of the REN is illustrated in Fig. 4. Note that during the training process of the REN, $\mathbf{h}'_x$ acts as input, and $\mathbf{h}'_y \in \mathbb{R}^{2FT\times 1} = [\mathrm{Re}(\mathbf{h}_y)^T, \mathrm{Im}(\mathbf{h}_y)^T]^T$ as supervising data. In this way, we aim to capture the inherent non-linear correlation between $\mathbf{h}_x$ and $\mathbf{h}_y$. However, after the network is well-trained, one can only use the noisy pilot CSI $\breve{\mathbf{h}}'_x$ to estimate $\mathbf{h}_y$, i.e., $\breve{\mathbf{h}}'_{y,\text{REN}}$. Therefore, it is necessary and important to first improve the SNR of pilots CSI, which was exactly conducted in the first block IDN. We denote $\mathbf{f}_{\text{REN}}(.)$ as the overall function for the proposed resolution enhancement network, $\boldsymbol{\theta}_{\text{REN}}$ as the hyperparameter vector, and Ω_2 as all training sets. Similar as equation (3), the overall loss function C_2 for this network is presented by

$$C_2 = \sum_{\mathbf{h}'_x \in \Omega_2} ||\mathbf{f}_{\text{REN}}(\mathbf{h}'_x, \boldsymbol{\theta}_{\text{REN}}) - \mathbf{h}'_y||_F^2. \tag{4}$$

During the training phase, the hyperparameter vector $\boldsymbol{\theta}_{\text{REN}}$ is optimized in order to minimize cost function C_2.

IV. Results and Discussion

In this section, we evaluate our proposed networks using the open-source WAIC2nd Dataset [15]. An overview of the Dataset and detailed settings of the networks is included in the sequel.

A. An Overview of the Dataset

In the Dataset, downlink channels of a 5G base station were simulated. According to the simulation setup, the base station was equipped with 32 antennas, operating in a beam-locking mode. That is, a beam was formed by the 32 antennas to serve a user equipment that equipped

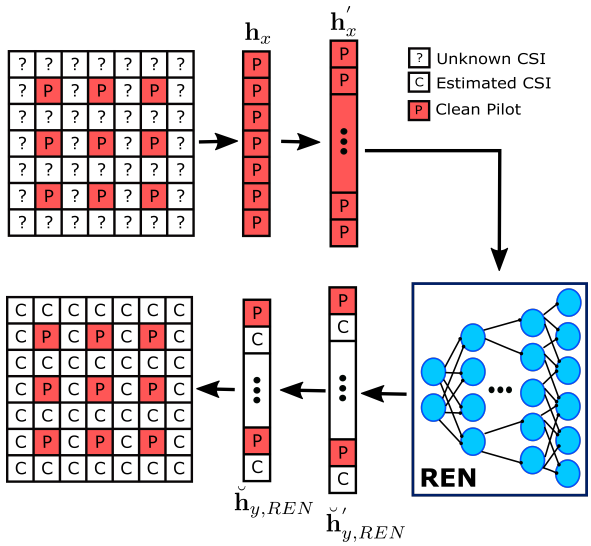


Fig. 4: The architecture of the resolution enhancement network.

with a single antenna. Thus, the downlink channels were essentially SISO. The considered time-frequency resources consist of 96 subcarriers and 14 OFDM symbols, which occupy 1.6 MHz bandwidth and each OFDM symbol lasts for 1 ms. Demodulation reference signals (DMRSs) were sparsely inserted as pilots into the time-frequency grids, occupying 48 subcarriers and 2 OFDM symbols. According to this, $\tilde{\mathbf{h}}_x \in \mathbb{C}^{96\times 1}$ and $\mathbf{H}_y \in \mathbb{C}^{96\times 14}$. In total $210,000$ $\tilde{\mathbf{h}}_x$ samples together with their corresponding groundtruths $\mathbf{H}_y$ are prepared. In addition, each $\tilde{\mathbf{h}}_x$ has an SNR label, ranging from 0 dB to 20 dB. Fig. 5 illustrates the CSI across the time-frequency grids of an example channel snapshot. The corresponding time-variant power delay profiles and delay-Doppler power spectrum are shown in Fig. 6. It can be observed from Fig. 6 that there are small numbers of multipath components (MPCs) existed in the channel. This is reasonable since beamforming can prune out-of-beam MPCs [16]. Nevertheless, as shown in Fig. 5, the channel still exhibited frequency and time selectivity. Readers are referred to [15] for more details.

B. Network Settings

1) IDN: As described in the previous section, the IDN takes the vector $\tilde{\boldsymbol{h}}'_x \in \mathbb{R}^{2P\times 1}$ as input, while the true vector $\mathbf{h}'_x$ is applied to supervise the training process. Specifically in this task, the input vector length is calculated as $2P = 2 \times 48 \times 2 = 192$. To reduce the network size and shorten the training time, we set $M = 5$ and the structure of each chain is selected to be identical. Specifically, each chain contains 4 cascaded fully-connected NN. The first NN has size 192×256, followed by 2 square NNs with identical structures as size 256×256, while the 4-th NN has size 256×192. In addition, decoders consist three cascaded fully-connected NN blocks, with sizes 192×256, 256×256 and 256×192 respectively. For all aforementioned

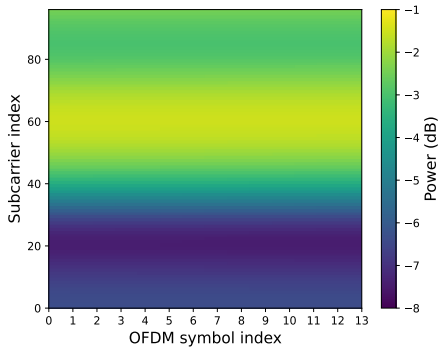
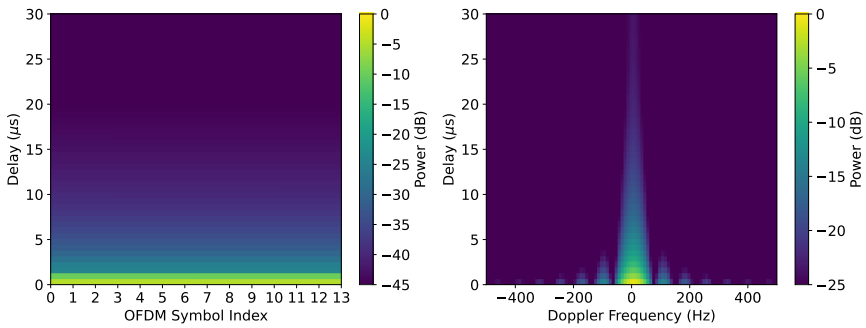


Fig. 5: CSI of an example channel snapshot.



(a) Time Variant power delay profiles

(b) Delay-Doppler power spectrum

Fig. 6: Time-variant power delay profiles and delay-Doppler spectrum of an example channel snapshot.

NNs, we apply the same parameter settings to simplify the network fine-tuning process during the training phase, which are listed in Table I.

TABLE I: IDN Parameter settings

Bias	Inactive
Loss Function	Mean Square Error
Activation Function	Leaky Relu (negative slope as -0.3)
Optimizer	Adam
Total Epoches	1000
Learning Rate	Initially 0.0004, reduce 50% every 200 epochs

2) REN: We input noise-free $\mathbf{h}'_x \in \mathbb{R}^{2P=2\times48\times2}$ to our proposed resolution enhancement network, and $\mathbf{h}'_y$ is used to supervise the training process. REN has 5 cascaded fully-connected layers. The size of first layer is 192×256, followed by 3 layers with the same network sizes

as 256×256. The last layer has size 256×2688. During the training phase, the learning rate is set to be 0.0004 at first, and reduced by half after 500 Epochs (1000 Epochs in total). Except for this particular parameter, we set other parameters exactly the same as that applied in IDN.

A popular open-source machine-learning framework is applied by us, namely *Pytorch*, to train our proposed NNs. After the training process, IDN, REN as two separate files are saved, with sizes $5160, 3667$ KB, respectively. We manage to limit the overall sizes of our proposed network within 10 MB, which enables opportunity to apply our proposed method to some embedded devices.

C. Final Results

To evaluate our proposed schemes, we randomly divide all $210,000$ samples ($\tilde{\mathbf{h}}_x$ and $\mathbf{H}_y$) into training and testing datasets. Specifically, 90% are used for training while the rest 10% ($21,000$ samples in total, 1000 samples per SNR point) for testing purposes. We select MSE score as a criteria to assess the performance of all NNs, which is defined as

$$Score = -10\log_{10}(\frac{1}{N}\sum_{i=1}^{N}\frac{||\mathbf{h}_i - \hat{\mathbf{h}}_i||_F^2}{||\mathbf{h}_i||_F^2}), \tag{5}$$

where N is the total number of testing samples, $\hat{\mathbf{h}}_i$ and $\mathbf{h}_i$ represent the estimated vector and ground-truth, respectively. Based on this criteria, we first illustrate the evaluation results of proposed denoise network with regard to different training epochs in Fig. 7. It can be observed that, the curve indicates a rising trend, although with some extend of fluctuations, with the increase of either input signal SNR or training Epochs. Nevertheless, *even under the region of low SNR*, the proposed denoise network still manages to reduce noise so that the outputs still approach closely to groundtruths. In addition, after 800 Epoches, only marginal improvement can be attained, since proposed network performance reaches a saturation point. At the next step, the REN network itself is evaluated by inputting all $21,000$ test samples at one shot, since the all input samples are noise-free. During the training phase, REN performance reaches a saturation point (score around 43) after 700 Epochs.

Finally, we evaluate our proposed scheme by combining both two sub-networks together and compare the result with existing two benchmarks, namely LMMSE and Channelnet. For the sake of a fair comparison, the same training and testing datasets are applied to all those three schemes. As illustrated in Fig. 8, a dramatic performance improvement can be achieved compared by our approach with LMMSE and Channelnet under all SNR scenarios. Specific to this testing data, Channelnet performs almost equally good as LMMSE (marginally better under low SNR region). Compared with them, our proposed approach shows much stronger robustness against low SNRs, we postulate that it is because our IDN plays an important role in enhancing signal quality. Our proposed approach outperforms LMMSE by 15 dB in the low SNR regime (SNR around 0 dB), and by 10 dB in the high SNR regime (SNR around 20 dB).

V. Conclusion

We proposed a novel supervised-learning based channel estimation approach, in order to acquire channel state information (CSI) for all payload symbols based on a few pilot symbols.

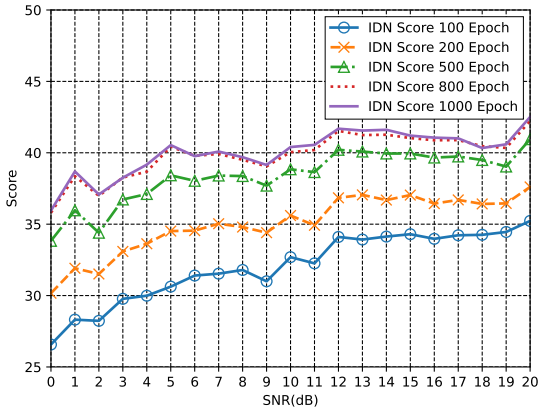


Fig. 7: Evaluation scores of IDN for scenarios with SNRs that range from 0 dB to 20 dB.

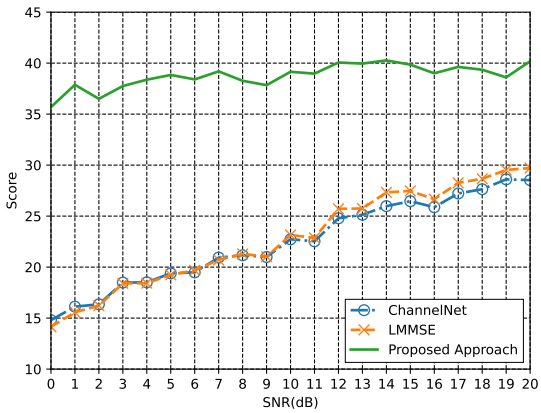


Fig. 8: Evaluation scores of the proposed approach, LMMSE and Channelnet for scenarios with SNRs that range from 0 dB to 20 dB.

The proposed pipeline consists of two cascaded blocks, namely, initial denoise network (IDN), and resolution enhancement network (REN). By applying a novel denoising structure in the IDN, pilot CSI estimation can achieve a significant improvement. With the aid of denoised pilot CSI, a decent estimation result can be achieved from the REN. In comparison to previous techniques such as linear minimum mean-square error method as well as ChannelNet, our proposed pipeline delivers, especially in high noise scenarios, better performance in terms of mean square error of the channel estimates. We leave further investigations on the potentials to apply our methods to multiple antenna systems as well as more complex propagation channels as future targets.

ACKNOWLEDGMENT

The authors gratefully acknowledge the technical support from Mr. Michiel Sandra. The support of Ericsson AB is also acknowledged for funding parts of this work.

References

[1] 3GPP release-17, Accessible online at https://www.3gpp.org/release-17.

[2] The 5G Infrastructure Association, "European Vision for the 6G Network Ecosystem, Accessible online at https://5g-ppp.eu/wp-content/uploads/2021/06/WhitePaper-6G-Europe.pdf.

[3] Ericsson, "Ericsson Mobility Report," June 2021, Accessible online at https://www.ericsson.com/en/reports-and-papers/mobility-report/reports.

[4] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath,"Pilot Contamination and Precoding in Multi-Cell TDD Systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2640-2651, Aug. 2011.

[5] E. Björnson, E. G. Larsson, and M. Debbah, "Massive MIMO for Maximal Spectral Efficiency: How Many Users and Pilots Should Be Allocated?," *IEEE Trans. Wireless Commun.*, vol. 15, no. 2, pp. 1293-1308, Feb. 2016

[6] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590-3600, Nov. 2010.

[7] M. K. Ozdemir, and H. Arslan, "Channel estimation for wireless OFDM systems," *IEEE Commun. Surv. Tutor.*, vol. 9, no. 2, pp. 18–48, Jul. 2007.

[8] S. Haghighatshoar, and G. Caire, "Massive MIMO Pilot Decontamination and Channel Interpolation via Wideband Sparse Channel Estimation," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8316-8332, Dec. 2017.

[9] H. Ye, G. Y. Li, and B. Juang,"Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Commun. Lett.* , vol. 7, no. 1, pp. 114-117, Feb. 2018.

[10] X. Gao, S. Jin, C. Wen, and G. Y. Li, "ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2627-2630, Dec. 2018.

[11] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep Learning-Based Channel Estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652-655, Apr. 2019.

[12] X. Yi, and C. Zhong, "Deep Learning for Joint Channel Estimation and Signal Detection in OFDM Systems," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2780-2784, Dec. 2020.

[13] M. Goutay, F. A. Aoudia, J. Hoydis and J. M. Gorce, "Machine Learning for MU-MIMO Receive Processing in OFDM Systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2318-2332, Aug. 2021.

[14] S. M. Kay, *Fundamentals of Statistical Processing Vol. I: Estimation Theory*, Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[15] https://www.datafountain.cn/competitions/504/datasets

[16] Cai, X., Miao, Y., Li, J., Tufvesson, F., Pedersen, G. & Fan, W., "Dynamic mmWave Channel Emulation in a Cost-Effective MPAC with Dominant-Cluster Concept," *IEEE Trans. Antennas Propag.*, 2021.

Paper IV

Paper IV

Reproduced, with permission, from: IEEE

B. R. Manoj, G. Tian, S. Willhammar, F. Tufvesson and E. G. Larsson, "Sensing and Classification Using Massive MIMO: A Tensor Decomposition-Based Approach," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2649-2653, Dec. 2021, doi: 10.1109/LWC.2021.3110463.

Sensing and Classification Using Massive MIMO: A Tensor Decomposition-Based Approach

B. R. Manoj, *Member, IEEE,* Guoda Tian, *Student Member, IEEE,* Sara Gunnarsson, *Student Member, IEEE,* Fredrik Tufvesson, *Fellow, IEEE,* and Erik G. Larsson, *Fellow, IEEE*

Abstract

Wireless-based activity sensing has gained significant attention due to its wide range of applications. We investigate radio-based multi-class classification of human activities using massive multiple-input multiple-output (MIMO) channel measurements in line-of-sight and non line-of-sight scenarios. We propose a tensor decomposition-based algorithm to extract features by exploiting the complex correlation characteristics across time, frequency, and space from channel tensors formed from the measurements, followed by a neural network that learns the relationship between the input features and output target labels. Through evaluations of real measurement data, it is demonstrated that the classification accuracy using a massive MIMO array achieves significantly better results compared to the state-of-the-art even for a smaller experimental data set.

Index Terms

Activity classification, large-scale sensing, massive MIMO, neural network, tensor decomposition.

I. Introduction

The rapid development of communication technologies over the past decade has paved the way and created a surge of interest in activity sensing applications using radio signals [1]. Exploiting characteristics of measured real-time radio channels for activity sensing is of importance for a variety of applications such as activity monitoring, security surveillance, crowd counting, elderly and children care [2], [3]. To realize these applications, the current state-of-the-art is to exploit channel state information (CSI) from existing WiFi devices that are ubiquitously available indoors [4].

Early work on radio-based activity sensing was based on extraction of features from radio signals and the use of traditional supervised machine learning (ML) algorithms to classify activities [2], [3], [5]. Recently, the trend has been going towards utilizing more powerful neural network (NN)-based classifiers to further improve the classification accuracy when having multiple possible events. These classifiers in general require a huge training data set [6]–[8]. A few examples of

This work was supported in part by ELLIIT, Security Link and a grant from Ericsson AB. The authors also gratefully acknowledge the suggestions by Prof. Bo Bernhardsson.

B. R. Manoj and Erik G. Larsson are with the Department of Electrical Engineering (ISY), Linköping University, 58183 Linköping, Sweden (e-mail: {manoj.banugondi.rajashekara, erik.g.larsson}@liu.se).

Guoda Tian, Sara Gunnarsson, and Fredrik Tufvesson are with the Department of Electrical and Information Technology, Lund University, 22100 Lund, Sweden (e-mail: {guoda.tian, sara.gunnarsson, fredrik.tufvesson}@eit.lth.se).

recent studies on activity sensing are classification of different human activities [3], human identity identification [7], human tracking [8], gesture recognition [9], and line-of-sight (LOS)/non line-of-sight (NLOS) identification in vehicle-to-vehicle (V2V) networks using multiple-input multiple-output (MIMO)-based systems [10].

These applications are either using WiFi-based devices operating at a carrier frequency of 2.4 GHz (or 5 GHz), or a MIMO-based channel sounder for the V2V propagation scenario, operating at 5.9 GHz [10], [11]. Current WiFi-based sensing applications use CSI measured by conventional (small-scale) MIMO systems, limiting the performance due to a smaller number of antennas, which imposes difficulties in exploiting the spatial resolution abilities. To overcome this and take activity sensing and classification one step further, we started in [12] to consider massive MIMO for these applications. A massive MIMO array offers very high spatial resolution along with being robust to interference. By increasing the number of antennas at the base station (BS), we can also exploit the spatial domain when extracting features from the data and hence, improve the performance. For this purpose, we conducted experiments with different activities in an indoor environment using a massive MIMO testbed operating at 3.7 GHz. While our work in [12] focused on binary classification between static and dynamic events, we here further advance this study by instead considering the multi-class classification problem for these activities in both LOS and NLOS propagation scenarios.

More specifically, in [12], we proposed a feature extraction method using principal component analysis (PCA) for the amplitude information and linear regression analysis for the phase information. Herein we further improve our approach by better exploiting the phase information and the spatial domain by developing an algorithm that uses the complex correlation properties across all three domains: time, frequency, and space. Since the measured data is a tensor spanning all three domains, the dimension is large and to reduce the size, we apply a tensor decomposition-based approach to obtain low-rank approximations. To learn the relationship between these features and their corresponding label, we propose a NN architecture with feedforward fully connected multiple layers, which is generalized for both LOS and NLOS scenarios.

II. System model

We consider an uplink massive MIMO system with M antenna elements at the BS using orthogonal frequency division multiplexing (OFDM) with F subcarriers. The multiple user equipments (UEs) utilize different but neighboring frequency subcarriers to simultaneously sound pilot signals, which are captured by an antenna array connected to M radio-frequency (RF) chains. We capture the signal over a certain observation time, recording in total T snapshots of the channel. The received matrix $\mathbf{Y}_f \in \mathbb{C}^{M\times T}$, for each subcarrier index f and each UE, can be written as

$$\mathbf{Y}_f = \mathbf{H}_f \odot \mathbf{\Gamma}_f + \mathbf{N}_f \,, \tag{1}$$

where snapshot $t \in [1, T]$, subcarrier $f \in [1, F]$, RF chain $m \in [1, M]$, $\odot$ denotes the Hadamard product, the complex matrix $\mathbf{H}_f \in \mathbb{C}^{M\times T}$ represents the channel, $\mathbf{\Gamma}_f \in \mathbb{C}^{M\times T}$ represents the frequency responses of the RF chains connected to the antenna array, and $\mathbf{N}_f \in \mathbb{C}^{M\times T}$ the noise

matrix for subcarrier f, where each element of the matrix represents the additive noise to the received signal.

It is a highly challenging task to formulate a precise model of the propagation channel $\mathbf{H}_f$ since the information about geographical locations of the transmitting UEs, as well as scatters, is unknown. In addition, knowledge about the speed and directions of moving objects is not available, which imposes limitations on estimating the corresponding Doppler shifts. Furthermore, each element in the RF chain matrix $\mathbf{\Gamma}_f$, modeled as $\mathbf{\Gamma}_f(m,t) = d_m e^{j(\varphi_m - t\,\eta_{m,f})}$, introduces uncertainties in the measured channel responses, where d_m, φ_m, and $\eta_{m,f}$ represent the amplitude, initial phase drift, and carrier-frequency-offset (CFO), respectively, for the f-th subcarrier and the m-th RF chain at time index t. For the collected measurement data, we define a third-order tensor as $\mathcal{G} \in \mathbb{C}^{T\times F\times M}$, which captures the channel in the time, frequency and spatial domains.

III. Algorithm for activity classification

In this section, we propose a novel algorithm to extract potential features from the measured channel transfer function, which is a third-order complex tensor $\mathcal{G}$. From $\mathcal{G}$, we extract the raw amplitude and the complex correlation across two domains by fixing the third one; thus, obtaining complex correlated tensors across all domains. Further, we utilize the real, imaginary, amplitude, and phase information of these tensors to extract features. The raw channel estimates from the measurement data itself could be provided to the NN model by designing a robust and complex NN to learn the statistical features of the data set. However, this approach is not feasible in our case due to the large dimensions of the measurement data. Hence, instead of utilizing the raw tensors, we further employ a tensor decomposition-based approach in order to obtain the best low-rank approximations [13]. The aim of extracting features is to later utilize these in the learning algorithm to classify activities. Our proposed feedforward NN-based architectures will be presented in Subsection III-C.

A. Tensor decomposition

Before describing our proposed feature extraction algorithm, we provide a brief overview of the employed tensor decomposition-based approach using canonical decomposition and parallel factors, called the CP decomposition [14]. For any third-order real tensor $\mathcal{H} \in \mathbb{R}^{T\times F\times M}$, the CP decomposition can be written as a sum of $r_{\max}$ rank-one tensors [13] as

$$\mathcal{H} \approx \widetilde{\mathcal{H}} = \sum_{l=1}^{r_{\max}} \lambda_l \mathbf{x}_l \circ \mathbf{y}_l \circ \mathbf{z}_l \,, \tag{2}$$

where $\circ$ is the vector outer product, the factor vectors $\mathbf{x}_l \in \mathbb{R}^T$, $\mathbf{y}_l \in \mathbb{R}^F$, and $\mathbf{z}_l \in \mathbb{R}^M$ are normalized to unit length with weights being absorbed into λ_l, which in turn consists of the dominant eigenvalue that could be used as features. The factor matrices are expressed as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{r_{\max}}] \in \mathbb{R}^{T\times r_{\max}}$, $\mathbf{Y} \in \mathbb{R}^{F\times r_{\max}}$, and $\mathbf{Z} \in \mathbb{R}^{M\times r_{\max}}$, respectively.

For calculating the CP decomposition in (2), we use the alternating least squares (ALS) method [13]. The ALS method is used to solve linear least-squares problems by minimizing the mode-n of $||\mathcal{H}_{(n)} - \widetilde{\mathcal{H}}_{(n)}||$ in an iterative manner for a fixed term, where $n = \{1, 2, 3\}$, $||\cdot||$ is the matrix

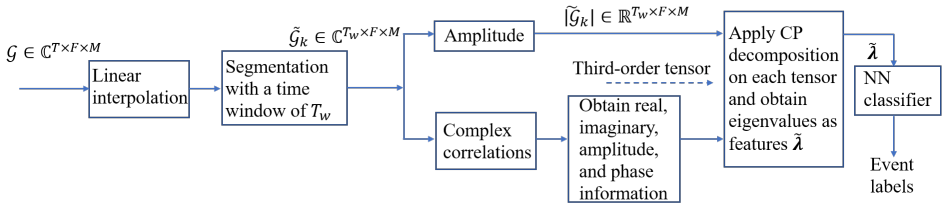


Fig. 1. Block diagram of the proposed activity sensing algorithm.

Frobenius norm, $\mathcal{H}_{(n)}$ and $\widetilde{\mathcal{H}}_{(n)}$ denote the mode-n unfolding of $\mathcal{H}$ and $\widetilde{\mathcal{H}}$, respectively. There is no simple method to determine the rank of a tensor and for any third-order tensor $\mathcal{H} \in \mathbb{R}^{T\times F\times M}$, only a weak bound on the maximum rank of $\mathcal{H}$ is known [13] as $\mathrm{rank}(\mathcal{H}) \leq \min\{TF, TM, FM\}$.

B. Feature extraction

The proposed flowchart for feature extraction and classification is illustrated in Fig. 1. During the measurements, frame losses occur occasionally due to synchronization failures. To account for such frame losses, we apply linear interpolation in the time domain as a first step when pre-processing the measured data, to obtain equally spaced time domain samples. Since the losses are intermittent, it is reasonable to expect that linear interpolation is sufficient for this purpose. Then, for each measurement, we perform segmentation of the received tensor $\mathcal{G}$ in the time domain, that is, a time window is used to segment the received data. The objective of the segmentation is to acquire a larger data set of shorter measurements to be used in the learning algorithm and for the performance evaluation. We choose a time window of length T_w over the entire duration of the observation time T. The value of T_w is chosen such that it is much larger than the channel coherence time so that small changes of the time-varying correlation properties of the channels due to movement of activities are well captured. The time-segmented tensor data is denoted by $\widetilde{\mathcal{G}}_k \in \mathbb{C}^{T_w\times F\times M}$, where $k \in [1, T/T_w]$.

1) Amplitude: Obtaining features from the raw phase information of $\widetilde{\mathcal{G}}_k$ is challenging and may not be accurate since both the initial phase offsets (φ_m) and CFOs ($\eta_{m,f}$) of each RF chain m contribute to the raw phases of the measured tensors. Furthermore, deriving an estimator to obtain those parameters is a non-trivial task. Therefore, we here only extract the feature from the raw amplitude information $|\widetilde{\mathcal{G}}_k| \in \mathbb{R}^{T_w\times F\times M}$, where $|\widetilde{\mathcal{G}}_k|$ is the amplitude of the k-th time window of $\mathcal{G}$.

2) Complex correlations: To exploit the complex correlation properties of $\widetilde{\mathcal{G}}_k$, we determine the correlations as a function of two domains by fixing the third one. Thus, we evaluate the correlation functions of the channel in all three domains: time, frequency and space. For the k-th time window, we denote the third-order tensors $\mathcal{C}^M_{F,k} \in \mathbb{C}^{T_w\times T_w\times M}$ and $\mathcal{C}^M_{T_w,k} \in \mathbb{C}^{F\times F\times M}$ as the complex correlation matrices for each antenna $m \in [1, M]$ across frequency and time, respectively.

More precisely, for the m-th antenna, the correlation matrices across frequency and time are given by

$$\begin{aligned} \mathcal{C}^{M}_{F,k}(:,:,m) &= \widetilde{\mathcal{G}}_k(:,:,m)\,(\widetilde{\mathcal{G}}_k(:,:,m))^{\mathrm{H}}, \quad \text{and} \\ \mathcal{C}^{M}_{T_w,k}(:,:,m) &= (\widetilde{\mathcal{G}}_k(:,:,m))^{\mathrm{H}}\,\widetilde{\mathcal{G}}_k(:,:,m). \end{aligned} \tag{3}$$

where $(\cdot)^{\mathrm{H}}$ is the Hermitian transpose. Similarly to (3), we define $\mathcal{C}^{F}_{M,k} \in \mathbb{C}^{T_w \times T_w \times F}$ and $\mathcal{C}^{F}_{T_w,k} \in \mathbb{C}^{M \times M \times F}$ for each subcarrier $f \in [1, F]$ as tensors describing the spatial and time correlations, respectively. The tensors for each time sample $t_w \in [1, T_w]$ describing the spatial and frequency correlation matrices are denoted by $\mathcal{C}^{T_w}_{M,k} \in \mathbb{C}^{F \times F \times T_w}$ and $\mathcal{C}^{T_w}_{F,k} \in \mathbb{C}^{M \times M \times T_w}$, respectively. Following these definitions, for each subcarrier f, we can write

$$\begin{aligned} \mathcal{C}^{F}_{M,k}(:,:,f) &= \widetilde{\mathcal{G}}_k(:,f,:)\,(\widetilde{\mathcal{G}}_k(:,f,:))^{\mathrm{H}}, \quad \text{and} \\ \mathcal{C}^{F}_{T_w,k}(:,:,f) &= (\widetilde{\mathcal{G}}_k(:,f,:))^{\mathrm{H}}\,\widetilde{\mathcal{G}}_k(:,f,:). \end{aligned} \tag{4}$$

For each time sample t_w, we have

$$\begin{aligned} \mathcal{C}^{T_w}_{M,k}(:,:,t_w) &= \widetilde{\mathcal{G}}_k(t_w,:,:)\,(\widetilde{\mathcal{G}}_k(t_w,:,:))^{\mathrm{H}}, \quad \text{and} \\ \mathcal{C}^{T_w}_{F,k}(:,:,t_w) &= (\widetilde{\mathcal{G}}_k(t_w,:,:))^{\mathrm{H}}\,\widetilde{\mathcal{G}}_k(t_w,:,:). \end{aligned} \tag{5}$$

Each element of the tensors in (3)–(5) is complex-valued. Thus, when utilizing the tensor decomposition-based approach discussed in Subsection III-A that operates in the real domain, we first need to obtain real tensors by extracting the amplitude and unwrapped phase information from each of the tensors in (3)–(5). Then we normalize the amplitudes and phases separately. By considering $\mathcal{C}^{M}_{F,k}$ in (3) as an example, we denote $\mathcal{C}^{M,\mathrm{A}}_{F,k}$ and $\mathcal{C}^{M,\mathrm{P}}_{F,k}$ as the amplitude and unwrapped phase tensors, which are defined as

$$\begin{aligned} \mathcal{C}^{M,\mathrm{A}}_{F,k}(:,:,m) &= |\mathcal{C}^{M}_{F,k}(:,:,m)|/||\mathcal{C}^{M}_{F,k}(:,:,m)|| \\ \mathcal{C}^{M,\mathrm{P}}_{F,k}(:,:,m) &= \arg(\mathcal{C}^{M}_{F,k}(:,:,m))/||\arg(\mathcal{C}^{M}_{F,k}(:,:,m))||, \end{aligned} \tag{6}$$

where $|\cdot|$ and $\arg(\cdot)$ denote magnitude and unwrapped phase, respectively. Following the same procedure as in (6), we obtain the rest of the real tensors from (3)–(5) with the dimension of each real tensor is same as the corresponding complex tensor. The complete list of real tensors is provided in Table I.

Furthermore, we also normalize the inner and outer products of (3)–(5) with the Frobenius norm, and then extract the real, imaginary, and amplitude information from the so-obtained normalized complex tensors. We explain this with an example: let us denote $\widetilde{\mathcal{C}}^{M}_{F,k}$ as the normalized complex tensor of $\mathcal{C}^{M}_{F,k}$, defined for each m as

$$\widetilde{\mathcal{C}}^{M}_{F,k}(:,:,m) = \mathcal{C}^{M}_{F,k}(:,:,m)/||\mathcal{C}^{M}_{F,k}(:,:,m)||, \forall\, m\,. \tag{7}$$

Then the real, imaginary, and amplitude information is obtained as $\widetilde{\mathcal{C}}^{M,\mathrm{Re}}_{F,k} = \mathrm{Re}\{\widetilde{\mathcal{C}}^{M}_{F,k}\}$, $\widetilde{\mathcal{C}}^{M,\mathrm{Im}}_{F,k} = \mathrm{Im}\{\widetilde{\mathcal{C}}^{M}_{F,k}\}$, and $\widetilde{\mathcal{C}}^{M,\mathrm{A}}_{F,k} = |\widetilde{\mathcal{C}}^{M}_{F,k}|$, respectively. We have noticed that the phase information provided in (6) is already sufficient for our classification task. By following the same procedure for the remaining tensors in (3)–(5), we obtain all the real tensors as shown in Table I, which describes the correlation properties as a function of time, frequency and space. After obtaining all the real tensors, we finally apply the CP decomposition using the ALS algorithm to achieve the best

TABLE I
LIST OF BOTH COMPLEX AND REAL TENSORS AND FEATURES

Complex tensors	Real tensors	Features	Complex tensors	Real tensors	Features
$\widetilde{\mathcal{G}}_k$	$\lvert\widetilde{\mathcal{G}}_k\rvert$	$\boldsymbol{\lambda}_1$			
$\mathcal{C}^{M}_{F,k}$	$\mathcal{C}^{M,\mathrm{A}}_{F,k}$	$\boldsymbol{\lambda}_2$	$\mathcal{C}^{M}_{T_w,k}$	$\mathcal{C}^{M,\mathrm{A}}_{T_w,k}$	$\boldsymbol{\lambda}_7$
	$\mathcal{C}^{M,\mathrm{P}}_{F,k}$	$\boldsymbol{\lambda}_3$		$\mathcal{C}^{M,\mathrm{P}}_{T_w,k}$	$\boldsymbol{\lambda}_8$
	$\widetilde{\mathcal{C}}^{M,\mathrm{Re}}_{F,k}$	$\boldsymbol{\lambda}_4$		$\widetilde{\mathcal{C}}^{M,\mathrm{Re}}_{T_w,k}$	$\boldsymbol{\lambda}_9$
	$\widetilde{\mathcal{C}}^{M,\mathrm{Im}}_{F,k}$	$\boldsymbol{\lambda}_5$		$\widetilde{\mathcal{C}}^{M,\mathrm{Im}}_{T_w,k}$	$\boldsymbol{\lambda}_{10}$
	$\widetilde{\mathcal{C}}^{M,\mathrm{A}}_{F,k}$	$\boldsymbol{\lambda}_6$		$\widetilde{\mathcal{C}}^{M,\mathrm{A}}_{T_w,k}$	$\boldsymbol{\lambda}_{11}$
$\mathcal{C}^{F}_{M,k}$	$\mathcal{C}^{F,\mathrm{A}}_{M,k}$	$\boldsymbol{\lambda}_{12}$	$\mathcal{C}^{F}_{T_w,k}$	$\mathcal{C}^{F,\mathrm{A}}_{T_w,k}$	$\boldsymbol{\lambda}_{17}$
	$\mathcal{C}^{F,\mathrm{P}}_{M,k}$	$\boldsymbol{\lambda}_{13}$		$\mathcal{C}^{F,\mathrm{P}}_{T_w,k}$	$\boldsymbol{\lambda}_{18}$
	$\widetilde{\mathcal{C}}^{F,\mathrm{Re}}_{M,k}$	$\boldsymbol{\lambda}_{14}$		$\widetilde{\mathcal{C}}^{F,\mathrm{Re}}_{T_w,k}$	$\boldsymbol{\lambda}_{19}$
	$\widetilde{\mathcal{C}}^{F,\mathrm{Im}}_{M,k}$	$\boldsymbol{\lambda}_{15}$		$\widetilde{\mathcal{C}}^{F,\mathrm{Im}}_{T_w,k}$	$\boldsymbol{\lambda}_{20}$
	$\widetilde{\mathcal{C}}^{F,\mathrm{A}}_{M,k}$	$\boldsymbol{\lambda}_{16}$		$\widetilde{\mathcal{C}}^{F,\mathrm{A}}_{T_w,k}$	$\boldsymbol{\lambda}_{21}$
$\mathcal{C}^{T_w}_{M,k}$	$\mathcal{C}^{T_w,\mathrm{A}}_{M,k}$	$\boldsymbol{\lambda}_{22}$	$\mathcal{C}^{T_w}_{F,k}$	$\mathcal{C}^{T_w,\mathrm{A}}_{F,k}$	$\boldsymbol{\lambda}_{27}$
	$\mathcal{C}^{T_w,\mathrm{P}}_{M,k}$	$\boldsymbol{\lambda}_{23}$		$\mathcal{C}^{T_w,\mathrm{P}}_{F,k}$	$\boldsymbol{\lambda}_{28}$
	$\widetilde{\mathcal{C}}^{T_w,\mathrm{Re}}_{M,k}$	$\boldsymbol{\lambda}_{24}$		$\widetilde{\mathcal{C}}^{T_w,\mathrm{Re}}_{F,k}$	$\boldsymbol{\lambda}_{29}$
	$\widetilde{\mathcal{C}}^{T_w,\mathrm{Im}}_{M,k}$	$\boldsymbol{\lambda}_{25}$		$\widetilde{\mathcal{C}}^{T_w,\mathrm{Im}}_{F,k}$	$\boldsymbol{\lambda}_{30}$
	$\widetilde{\mathcal{C}}^{T_w,\mathrm{A}}_{M,k}$	$\boldsymbol{\lambda}_{26}$		$\widetilde{\mathcal{C}}^{T_w,\mathrm{A}}_{F,k}$	$\boldsymbol{\lambda}_{31}$

low-rank approximation for the individual real tensors and then to obtain the eigenvalues vectors $\boldsymbol{\lambda}_i \in \mathbb{R}^{r_{\max}}$ with $i \in \{1, \cdots, 31\}$, where each vector $\boldsymbol{\lambda}_i$ is sorted in the descending order. These vectors are then used as potential features in the NN learning algorithm.

C. Feedforward neural network

We propose a feedforward NN architecture with fully connected layers comprising an input layer of dimension $\mathbb{R}^u$, K hidden layers, and an output layer of dimension $\mathbb{R}^v$. The reason behind choosing NN instead of classical supervised ML is that, in this work, the classification problem includes different types of activities and the channel and signal models are unknown due to random Doppler shifts, hence the statistical characteristics are non-linear. We therefore consider the NN-based approach, which has the ability to transform the input domain through a linear mapping and utilize a non-linear activation function to learn a non-linear pattern between a known input and the target output labels.

A simple feedforward NN classifier model is considered, denoted as $f(.;\boldsymbol{\theta}) : \mathbf{x} \in \mathcal{X} \rightarrow \mathbf{c} \in \mathcal{C}$, where $\boldsymbol{\theta}$ is the model hyperparameters, $\mathcal{X} \subset \mathbb{R}^u$ is the input to the NN with u as the dimension of the input, and $\mathcal{C} \subset \mathbb{R}^v$ is the label (one-hot) with v being the number of classes. The optimization loss function is denoted as $\mathcal{L}(\boldsymbol{\theta}, \mathbf{x}, \mathbf{c})$; for the classification task, the categorical cross-entropy between $f(\mathbf{x})$ and $\mathbf{c}$ is applied to this function. More specifically, the measurements data set is

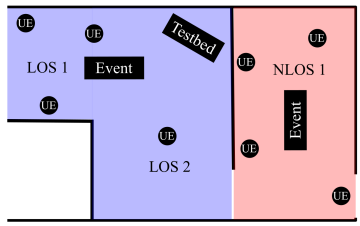


Fig. 2. Map showing the measured scenarios. The UEs were placed either in LOS 1 and 2 or in NLOS 1. When measuring a dynamic environment, the "Event" rectangle was the place where the activity was performed.

defined as $\{\mathcal{G}_j, y_j\}$, $j = 1, \cdots, N$, where N is the size of the data set, $\mathcal{G}_j \in \mathbb{C}^{T\times F\times M}$ is the j-th measurement data tensor of an activity, and y_j is the corresponding label. The extracted feature set with the raw amplitude and correlation properties of the received tensors is given by $\widetilde{\boldsymbol{\lambda}} = [\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_{31}]$, where $\boldsymbol{\lambda}_i \in \mathbb{R}^{r_{\max}}$ with $i = \{1, \cdots, 31\}$. Therefore, the input to the NN becomes the feature set, which is defined as $\mathbf{x} = \mathrm{vec}(\widetilde{\boldsymbol{\lambda}})$ and the corresponding output of the NN is the one-hot encoded label $\mathbf{c}$. We propose a generalized NN architecture model for both the LOS and NLOS propagation scenarios, specifically a feedforward fully connected multi-layer NN with parameters $64 - 32 - 32 - 32 - 5$ and corresponding activation functions after the linear mapping *elu* $-$ *elu* $-$ *elu* $-$ *elu* $-$ *softmax*. The hyperparameter set is designed as: optimizer = Adam, loss function = categorical cross-entropy, and learning rate = 0.001. In the feature set $\widetilde{\boldsymbol{\lambda}}$, for both the LOS and NLOS scenario, we discard the largest eigenvalue from each vector $\boldsymbol{\lambda}_i$, $i \in [1, 31]$, since its value is high and does not distinguish across the activities, making it difficult for accurate classification.

IV. Measurement scenarios and setup

A measurement campaign has been carried out to collect real-time channels. Measurements were done in both LOS and NLOS as well as under static and dynamic conditions. The measurements were done in an indoor laboratory sketched in Fig. 2, with the Lund University massive MIMO testbed (LuMaMi) [15] acting as BS. The measurement campaign was conducted in the presence of a large number of static, scattering objects such as furniture equipment and cabinets located in the surrounding. All UEs were distributed randomly, also with placement at different heights in the LOS or the NLOS area, depending on the scenario. In case of a dynamic scenario, the activity took place in the corresponding "Event" rectangle in the figure.

LuMaMi is based on software-defined radios (SDRs) and operates at a carrier frequency of 3.7 GHz with 20 MHz bandwidth in real-time. It has 100 RF chains, each one connected to a patch antenna, building up to a rectangular array of dimensions 4×25 where the antennas are spaced half a wavelength apart. The antenna in the upper left corner, as seen from the front, is vertically polarized and the polarization is alternating such that two consecutive elements always have different polarizations. The BS is collecting channel estimates for 30 seconds per measurement, based on the received pilot signals from the UEs, which are also SDR-based. The sampling rate is 100 Hz, resulting in a 10 ms time duration between adjacent samples. Each

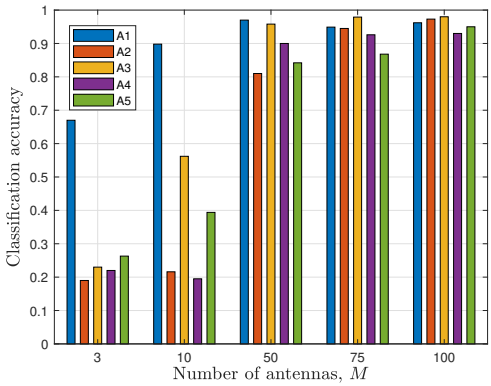


Fig. 3. LOS: Classification accuracy of activities as a function of M.

UE consists of an universal software radio peripheral (NI X310) with two RF chains, equipped with either one or two dipole antennas. During a measurement, the channel from each UE is stored and the resulting channel transfer function will be a tensor containing $T = 3000$ snapshots, $F = 100$ frequency points and $M = 100$ BS antennas; this is referred to as one experiment.

For the activity classification, four different dynamic events were considered besides the static case. The measured events are named as follows: (i) A_1- completely static and in the presence of a static person, static bike wheel, or static aluminium foil balloon, (ii) A_2- waving of an aluminium foil balloon by a volunteer lying on the floor, (iii) A_3- random dancing activities by a volunteer, (iv) A_4- spinning and moving bike wheel by a volunteer lying on the floor, and (v) A_5- spinning bike wheel by a volunteer without other movements. These activities were selected to be specific movements involved in our daily life, and include the properties of stationarity (A_1), periodicity (A_2), randomness (A_3), as well as rotation and shifting (A_4 and A_5). For each dynamic event and per scenario (LOS or NLOS), three measurements were done with six UEs, resulting in a total of 18 experiments, while for the event A_1, we conducted 36 experiments. The measurements of all events were conducted in turn by a few volunteers of different age groups. The experiments under LOS and NLOS scenarios were carried out at different times.

V. Experimental results and discussion

In this section, we evaluate the performance of our algorithm to classify different events under LOS and NLOS scenarios with the proposed NN model. In addition to the classification performance, the potential gain due to the extension in the spatial domain that comes with the massive MIMO array is also investigated by showing performance results for different subsets of the antenna array. In the performance evaluation, we choose $r_{\max} = 100$ such that the low-rank approximation gives a good fit and fulfills the aim of dimension reduction while still preserving most of the information contained in the original tensor. Thus, the dimension of inputs to the NN becomes $\mathbb{R}^{3069}$. Further, we choose $T_w = 200$ so that the data set of each experiment is increased

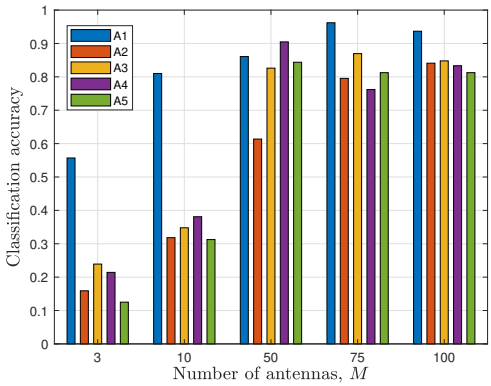


Fig. 4. NLOS: Classification accuracy of activities as a function of M.

by a factor of $T/T_w = 15$. For each dynamic event, the data set size becomes $18 \times 15 = 270$ samples, while for the static case it is $36 \times 15 = 540$ samples, where 18 and 36 denote the number of experiments in dynamic and static environments, respectively. Considering all the activities, in total, there are 1620 samples. We stack the features of the events A_1 to A_5; thereby the input structure of the NN has the dimension $\mathbb{R}^{1620\times 3069}$. We then randomly split those 1620 samples into two parts, 85% and 15% as training and test samples, respectively. During the training phase, the NN architecture is trained with the features obtained for the events from the proposed algorithm and the corresponding target output labels.

The classification accuracy of the activities $A_1 - A_5$ with antenna array sizes ranging from $M = 3$ to $M = 100$ in the LOS and NLOS scenarios is shown in Figs. 3 and 4, respectively. As a special case when $M = 100$, the confusion matrices are depicted in Fig. 5. In the planar antenna array, the antennas are numbered row-wise starting from the upper left corner to the lower right corner. For obtaining different antenna subsets, we subsample the array in the sequence of $M = 3, 10, 50$, etc., that is, the first 3 adjacent antennas, the first 10 adjacent antennas, and so forth.

In both the LOS and NLOS scenarios, the classification of different activities is challenging for conventional MIMO systems (here seen for $M \leq 10$), since the potential of extracting statistical characteristics embedded in the spatial domain is insufficient. However, when increasing the number of antennas, a significant performance improvement, in terms of classification accuracy, can be observed. From $M = 10$ to $M = 50$, we have a substantial gain in performance due to the aperture increase both in the horizontal and vertical directions. With $M = 100$ in the LOS scenario, the classification accuracy is 98%, whereas in the NLOS scenario it is 87%. For performance comparison, we implemented the PCA-based method in [12] for the multi-class classification problem, and obtained the accuracy 83% and 73% for the LOS and NLOS scenarios, respectively. Thus the method proposed here outperforms that in [12] significantly. In terms of complexity, we notice that it is not necessary to spend extra computational resources to achieve this accuracy

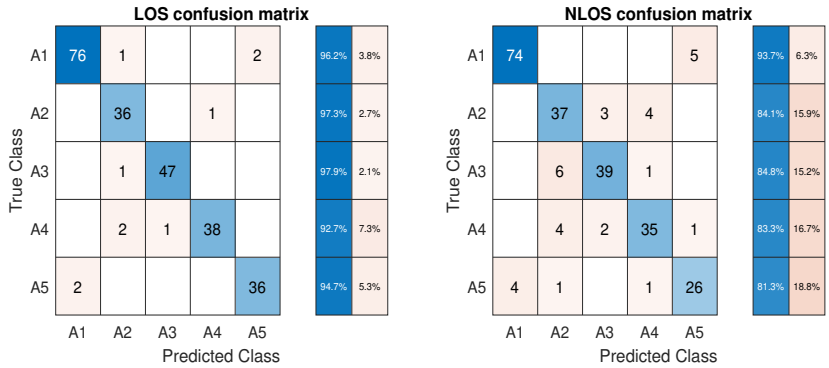


Fig. 5. Confusion matrices for LOS and NLOS scenarios when $M = 100$.

gain. Specifically, [12] has an overall complexity of $\mathcal{O}(TFMT_w)$ while the proposed algorithm herein has $\mathcal{O}(MFT_w \max(T_w, F, M))$, since it manages to avoid PCA computation for each time-segmented data. The classification in LOS is most likely an easier task as changes in the LOS component caused by the different activities will influence the channel more, hence making it easier to distinguish between the different activities. Also, the large-scale fading effects over the antenna array due to blockage will likely be more prominent and therefore have a larger impact on the correlation in the spatial domain; this effect is probably not that prominent in the NLOS scenario. Moreover, the UE locations could also affect the performance, since the placement of both the UE and the local scatterers around the UE influences the received signal. However, it is challenging to predict the actual UE positions in an application scenario, since the UE distribution has a strong randomness.

To cross-validate and guarantee that our proposed algorithm learns the propagation channel characteristics rather than, e.g., gradual variations in the RF characteristics, we divide the measurement data of each experiment into two parts. Specifically, we label the first half of the entire snapshots as "early", and the second half as "late". For each activity, we then randomly divide 80% of the measurement as a training set while 20% as the test set, followed by training a NN with the same structure but with the size of the last layer as 2 (i.e., binary classification). The classification accuracy of this for each activity on the test data set shows around 50%, which indicates that the NN does not pick up irrelevant slow trends of the measurement data. Furthermore, the NN might (as a byproduct) learn a fingerprint associated with a specific location. However, that information is not exploited when discriminating between different activities. The NN only uses the information it learned about what the different activities look like.

VI. Conclusions

We have investigated NN-based multi-class activity classification by utilizing baseband data from a massive MIMO array. Our proposed method is tested on data obtained from an indoor measurement campaign involving both LOS and NLOS scenarios, using the LuMaMi testbed

equipped with 100 antennas. To efficiently exploit the information embedded in the time, frequency, and spatial domains, we employed a tensor decomposition-based approach to obtain the eigenvalues as features from the three-dimensional measurement data. Furthermore, by using the proposed NN architecture, we obtained a multi-class classification accuracy that reached 98% and 87% in the LOS and NLOS scenarios, respectively; these numbers are significantly higher than for a system with few antennas. This showcases the potential benefits of using massive MIMO for wireless sensing applications. The results look very promising for the relatively small experimental data set we used, and further investigation is required by conducting diverse measurements in order to explore the full potential of massive MIMO for sensing applications.

REFERENCES

[1] K. Qian *et al.*, "Enabling contactless detection of moving humans with dynamic speeds using CSI," *ACM Trans. Embed. Comput. Syst.*, vol. 17, no. 2, Jan. 2018.

[2] W. Liu, S. Chang, Y. Liu, and H. Zhang, "Wi-PSG: Detecting rhythmic movement disorder using COTS WiFi," *IEEE Internet of Things Journ.*, vol. 8, no. 6, pp. 4681–4696, 2021.

[3] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Device-free human activity recognition using commercial WiFi devices," *IEEE J. Sel. Areas in Commun.*, vol. 35, no. 5, pp. 1118–1131, May 2017.

[4] I. Nirmal, A. Khamis, M. Hassan, W. Hu, and X. Zhu, "Deep learning for radio-based human sensing: Recent advances and future directions," 2021. [Online]. Available: https://arxiv.org/abs/2010.12717

[5] L. Zhang, X. Ruan, and J. Wang, "WiVi: A ubiquitous violence detection system with commercial WiFi devices," *IEEE Access*, vol. 8, pp. 6662–6672, 2020.

[6] F. Wang, W. Gong, J. Liu, and K. Wu, "Channel selective activity recognition with WiFi: A deep learning approach exploring wideband information," *IEEE Trans. Network Sci. and Eng.*, vol. 7, no. 1, pp. 181–192, 2020.

[7] J. Ding, Y. Wang, and X. Fu, "Wihi: WiFi based human identity identification using deep learning," *IEEE Access*, vol. 8, pp. 129 246–129 262, 2020.

[8] R. Zhou, M. Tang, Z. Gong, and M. Hao, "FreeTrack: Device-free human tracking with deep neural networks and particle filtering," *IEEE Systems Journ.*, vol. 14, no. 2, pp. 2990–3000, 2020.

[9] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conference on Computer Communications (INFOCOM)*, Hong Kong, China, Apr. 26-May 1, 2015, pp. 1472–1480.

[10] C. Huang *et al.*, "Machine learning-enabled LOS/NLOS identification for MIMO systems in dynamic environments," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3643–3657, Jun. 2020.

[11] R. Wang *et al.*, "A real-time MIMO channel sounder for vehicle-to-vehicle propagation channel at 5.9 GHz," in *Proc. IEEE International Conference on Communications (ICC)*, Paris, France, May 21-25, 2017, pp. 1–6.

[12] B. R. Manoj, G. Tian, S. Gunnarsson, F. Tufvesson, and E. G. Larsson, "Moving object classification with a sub-6 GHz massive MIMO array using real data," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, Jun. 6-11, 2021, pp. 8133–8137.

[13] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Sep. 2009.

[14] G. Bergqvist and E. G. Larsson, "The higher-order singular value decomposition: Theory and an application [lecture notes]," *IEEE Sig. Proc. Magaz.*, vol. 27, no. 3, pp. 151–154, 2010.

[15] S. Malkowsky *et al.*, "The world's first real-time testbed for massive MIMO: Design, implementation, and validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017.

Paper V

Paper V

Reproduced, with permission, from: IEEE

G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu and F. Tufvesson, "Deep-Learning-Based High-Precision Localization With Massive MIMO," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 19-33, 2024, doi: 10.1109/TMLCN.2023.3334712.

Deep-learning based high-precision localization with massive MIMO

Guoda Tian, Student Member, IEEE, Ilayda Yaman, Student Member, IEEE, Michiel Sandra, Student Member, IEEE, Xuesong Cai, Senior Member, IEEE, Liang Liu, Member, IEEE, and Fredrik Tufvesson, Fellow, IEEE
Dept. of Electrical and Information Technology, Lund University, Sweden.

Abstract

High-precision localization and machine learning (ML) are envisioned to be key technologies in future wireless systems. This paper presents an ML pipeline to solve localization tasks. It consists of multiple parallel processing chains, each trained using a different fingerprint to estimate the position of the user equipment. In this way, ensemble learning can be utilized to fuse all chains to improve localization performance. Nevertheless, a common problem of ML-based techniques is that network training and fine-tuning can be challenging due to the increase in network sizes when applied to (massive) multiple-input multiple-output (MIMO) systems. To address this issue, we utilize a subarray-based approach. We divide the large antenna array into several subarrays, feeding the fingerprints of the subarrays into the pipeline. In our case, such an approach eases the training process while maintaining or even enhancing the performance. We also use the Nyquist sampling theorem to gain insight on how to appropriately sample and average training data. Finally, an indoor measurement campaign is conducted at 3.7 GHz using the Lund University massive MIMO testbed to evaluate the approaches. Localization accuracy at a centimeter level has been reached in this particular measurement campaign.

Index Terms

Channel measurements, Deep learning, Localization, Massive MIMO

This work has been funded by Ericsson AB, the Horizon Europe Framework Programme under the Marie Skłodowska-Curie grant agreement No. 101059091, the Swedish Research Council (Grant No. 2022-04691), and the strategic research area ELLIIT, Excellence Center at Linköping-Lund in Information Technology. Corresponding authors: Guoda Tian (email: guoda.tian@eit.lth.se) and Xuesong Cai (email: xuesong.cai@eit.lth.se).

I. INTRODUCTION

CELLULAR-based localization is expected to pave the way for various location-aware applications such as robotic navigation, emergency healthcare, and smart transportation [1]–[7]. The technology has undergone a significant improvement over the years, and high-precision wireless localization has currently been included as a key feature in the current fifth generation new radio (NR) standard, with strict requirements on localization accuracy [8].

Traditional localization approaches include proximity, triangulation (trilateration), fingerprint matching, and simultaneous localization and mapping [7]. Proximity approaches examine whether user equipment (UE) is close to pre-known locations by analyzing received wireless signal characteristics such as the received signal strength indicator (RSSI). Triangulation or trilateration technology is used to estimate UE locations from delays or angles according to geometry. The general concept of fingerprint-based localization is to establish a radio map for the area of interest by storing channel features or fingerprints. The UE coordinates are estimated by comparing the received fingerprints with the previously stored fingerprints. Furthermore, with the aid of ultra-wideband (UWB) [9] and/or massive multiple-input multiple-output (MIMO) systems, it is possible to improve positioning accuracy due to the high delay resolution in UWB and the high angular resolution in massive MIMO [10]–[16]. For example, the work in [10], [15], [16] proposed novel estimators to jointly estimate angles and positions with large-scale arrays. The authors in [11]–[13] provided solutions for localization by designing tracking filters to exploit and track important propagation channel characteristics, i.e. the autocorrelation function of the received signal and the phase of multipaths, respectively. Especially, [12] validated their methods via a real massive MIMO testbed and showed that localization accuracy can be significantly enhanced with the 40 MHz bandwidth. [14] presented a direct localization method considering localization as a joint optimization problem, which bypasses the channel estimation step and still achieves good positioning accuracy.

All of the aforementioned localization methods belong to the traditional signal processing family. The main challenges are high algorithm complexity and requirements of the base station (BS) array calibration [7]. On the other hand, machine learning (ML) based localization algorithms have gained significant interest [17]–[32]. It is essential to appropriately select both fingerprints and algorithms. One can choose either the raw transfer function [17]–[19] or various channel fingerprints such as RSS, power delay profile (PDP), angular spectrum, correlation function, etc, [20]–[28] as learning features. Moreover, a variety of ML algorithms have been investigated, which can be mainly classified into two categories, namely the traditional ML family such as the K-nearest neighbors (KNN), support vector machines, kernel methods, random forest, Gaussian process regression, [21]–[23], [32], and the deep learning family [17], [18], [20], [24]–[31]. Considering the features of massive MIMO systems, there is also great potential to apply ML techniques with massive MIMO

systems to solve localization tasks. Early research [17], [18], [24] used convolutional neural networks (CNN) for localization. The work in [30] proposed an algorithm that trains an autoencoder to first calibrate the antenna array. Then, the angle spectrum is computed as a training feature. The work in [22] applied Gaussian regression to perform localization with distributed massive MIMO systems.

However, there are still some research gaps that need to be further addressed: *i)* Most of the existing ML-based localization algorithms directly output the position of the UE without considering the uncertainty of the estimation, thus lacking effective information fusion from different channel fingerprints. *ii)* The size of the neural networks increases significantly with the increasing number of antennas. This may hinder the training and fine-tuning of the network. Therefore, it is essential to develop efficient localization algorithms that are suitable for the massive MIMO system. *iii)* A theoretical analysis of the necessary training density is missing in the literature. It is important to investigate the density of the training sample under different channel conditions, as training data collection is a time-consuming task. To address those limitations, our main contributions are as follows.[1]

- We apply a localization framework that blends channel fingerprints that contain information from the delay and angular domains, respectively. It is not necessary to calibrate the whole BS array to obtain those channel fingerprints.
- By dividing the whole array into subarrays, the network size can be reduced, which facilitates the training process while improving the localization performance.
- We apply the Nyquist sampling theorem to analyze how to appropriately collect and average training data.
- Finally, an indoor measurement campaign with a massive MIMO testbed was conducted to evaluate our approach. The results show that our pipeline can reach centimeter-level positioning accuracy with only 20 MHz bandwidth for this measurement campaign.

The remainder of this paper is organized as follows. In Section II, we introduce the signal model and briefly discuss the selected fingerprints. In Section III, we present the localization algorithms. Section IV illustrates the measurement campaign, and Section V presents the results. Finally, conclusive remarks are included in Section VI.

II. System Model and Fingerprint Generation

We consider the uplink of a single user massive MIMO system, which uses orthogonal frequency division multiplexing (OFDM) with F subcarriers. The UE has one antenna, while the BS is equipped with M antennas. Each antenna element is connected to an RF and a digital processing chain, which allows the BS to simultaneously process the received signals from all antennas. We assume that the UE moves at walking speed, and

[1]A preliminary version of this work [33] has been presented at the 2023 IEEE International Conference on Communication. Unlike [33], this paper presents new material on the subarray method and a detailed analysis of the necessary training density. In addition, the pipeline in this paper is used to estimate both the UE position and error variances.

the 2-D position of this UE is given by $\mathbf{p}_i(x_0, y_0) \in \mathbb{R}^2$, shortened to $\mathbf{p}_i$ in the following sections. Taking into account the propagation channel, the transfer function matrix $\mathbf{Y}_{\mathbf{p}_i} = [\mathbf{y}_{\mathbf{p}_i,1}, ..., \mathbf{y}_{\mathbf{p}_i,F}] \in \mathbb{C}^{M\times F}$ for all subcarriers, corresponding to the position of the UE $\mathbf{p}_i$, can be written as

$$\mathbf{Y}_{\mathbf{p}_i} = \mathbf{H}_{\mathbf{p}_i} \odot \mathbf{\Gamma} + \mathbf{N}, \tag{1}$$

where $\mathbf{H}_{\mathbf{p}_i} \in \mathbb{C}^{M\times F}$ represents the uplink wireless propagation channel, $\mathbf{\Gamma} \in \mathbb{C}^{M\times F}$ the complex coefficients (amplitude scaling and phase drift) of all the M RF chains and F sub-carriers. Additionally, $\odot$ is the Hadamard product, and $\mathbf{N} \in \mathbb{C}^{M\times F}$ denotes receiver noise at all M RF chains. When UE moves, a total of $\mathcal{T}$ snapshots are recorded and $\mathcal{T}$ different receive matrices $\mathbf{Y}_{\mathbf{p}_i}$ are collected. Our aim is to find a functional relationship between $\mathbf{Y}_{\mathbf{p}_i}$ and $\mathbf{p}_i$, which falls into the category of a regression (estimation) task.

ML-based localization algorithms have the potential to achieve good performance if adequate channel fingerprints are selected as the input to the algorithms. Such fingerprints can be extracted from the raw received transfer function $\mathbf{Y}_{\mathbf{p}_i}$. In this paper, we analyze two fingerprints, namely the spatial covariance matrix and the truncated channel impulse response (CIR), since they can be achieved with even an uncalibrated array.

A. Spatial Covariance matrix

It is sometimes challenging to extract calibrated fingerprints such as AoA due to the presence of the RF chain matrix $\mathbf{\Gamma}$, see (1). Therefore, we consider using the covariance matrix $\mathbf{C}_i = \mathbb{E}\{\mathbf{y}_{p_i}\mathbf{y}_{p_i}^H\} \in \mathbb{C}^{M\times M}$ as a fingerprint. The main diagonal elements of $\mathbf{C}_i$ (auto-correlation) indicate the received signal power for each antenna, whereas the off-diagonal elements of $\mathbf{C}_i$ represent the cross-correlation between different antennas. Note that typically one can only estimate the covariance matrix in practice with a limited number of samples to conduct the expectation operation. Suppose that for each position $\mathbf{p}_i$, there exist in total $\mathcal{N}_{p_i}$ positions in the neighborhood region of $\mathbf{p}_i$, whose channel responses are accessible. Those $\mathcal{N}_{p_i}$ samples are inside a circular area, with $\mathbf{p}_i$ as the center and d as the diameter, i.e., $||\mathbf{p}_j - \mathbf{p}_i||^2 \leq \frac{d}{2}, j = 1, 2, ..., \mathcal{N}_{p_i}$. We then define the sample covariance matrix $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} \in \mathbb{C}^{M\times M}$ to estimate $\mathbf{C}_i$. Specifically,

$$\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} = \frac{1}{\mathcal{N}_{p_i}} \sum_{j=1}^{\mathcal{N}_{p_i}} \mathbf{Y}_{p_j}\mathbf{Y}_{p_j}^H. \tag{2}$$

As shown in (2), $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ depends on $\mathcal{N}_{p_i}$ and thus d. A special case is that if $d = 0$ and $\mathcal{N}_{p_i} = 1$, $\mathbf{C}_i$ is estimated by only correlating across all subcarriers of $\mathbf{Y}_{p_i}$ at a fixed position $\mathbf{p}_i$. We name this specific matrix the one-sample covariance matrix. Note that it is challenging to estimate $\mathbf{C}_i$ with this matrix for a narrowband system, since channel responses with respect to different subcarriers are strongly correlated. In contrast, when d is larger than half a wavelength, a major difference in the propagation channel can be

observed and $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ can therefore better approach $\mathbf{C}_i$. If d is large enough, the fingerprint $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ changes much more slowly than the one-sample covariance matrix with the movement of the UE, since the influence of small-scale fading is reduced due to the average operation. Due to this, fewer training samples are needed.

Since the sample covariance matrix $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ is a Hermitian matrix, i.e., $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}} = \tilde{\mathbf{C}}^H_{i,\mathcal{N}_{p_i}}$, the upper minor diagonal elements contain the same information as the lower. To decrease the computation complexity, we introduce another matrix $\breve{\mathbf{C}}_{i,\mathcal{N}_{p_i}} \in \mathbb{R}^{M\times M}$ and a vector $\tilde{\mathbf{c}}_{i,\mathcal{N}_{p_i}} \in \mathbb{R}^{M^2}$ as

$$\begin{aligned}\breve{\mathbf{C}}_{i,\mathcal{N}_{p_i}} &= \text{ltril}\big\{\Re\big[\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}\big]\big\}+\text{sltril}\big\{\Im\big[\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}\big]\big\}\\ \tilde{\mathbf{c}}_{i,\mathcal{N}_{p_i}} &= \text{vec}\big\{\breve{\mathbf{C}}_{i,\mathcal{N}_{p_i}}\big\},\end{aligned} \tag{3}$$

where $\Re\{.\}$ and $\Im\{.\}$ denote the operation to take the real and imaginary parts of a given matrix, respectively. ltril$\{.\}$ represents a matrix operation that replaces all values above the diagonal as zero while maintaining the other matrix elements. The operation sltril$\{.\}$ keeps all elements below the diagonal and substitutes all the remaining matrix elements (including the diagonal elements) for zero. The vec$\{.\}$ operator denotes the operation of converting a matrix to a vector.

B. Truncated channel impulse response

The fingerprint $\mathbf{C}_i$ does not contain channel information from the delay domain; however, it is still important to utilize the delay information to further improve the accuracy of localization. To this end, the truncated CIR matrix $\boldsymbol{\Xi} \in \mathbb{C}^{M\times L}$ is generated by calculating the inverse discrete Fourier transform (IDFT) along each row of $\mathbf{Y}_{p_i}$, followed by choosing the first L delay elements. We introduce a vector $\boldsymbol{\xi} \in \mathbb{R}^{2ML}$, which includes all elements of $\boldsymbol{\Xi}$. Specifically, $\boldsymbol{\xi} = [\text{vec}\{\Re(\boldsymbol{\Xi})\}^T, \text{vec}\{\Im(\boldsymbol{\Xi})\}^T]^T$.

III. ML-BASED LOCALIZATION APPROACH

A. Neural Network Basics

Neural networks have been widely used to solve various tasks such as channel estimation, wireless sensing, etc., owing to their excellent abilities to learn non-linear complex models [34]. These models can generally be represented as a multivariate function $f : \mathbb{R}^{V_1} \to \mathbb{R}^{V_2}$, where V_1 and V_2 represent the dimensions of the learning characteristics and the goals, respectively. An example of a typical fully connected neural network (FCNN) is illustrated in Fig. 1, consisting of an input layer, several hidden layers, and an output layer. Regarding the input and output layers, the number of their nodes is identical to V_1 and V_2, respectively. Specific to this 2-D localization task, we view the output of the neural network as a Gaussian distribution function, which can be determined by the estimated position of the UE ($\hat{\mathbf{p}} = [\hat{p}_x, \hat{p}_y]^T \in \mathbb{R}^2$) and the variance ($\hat{\boldsymbol{\sigma}}^2 = [\hat{\sigma}_x^2, \hat{\sigma}_y^2]^T \in \mathbb{R}^2$).

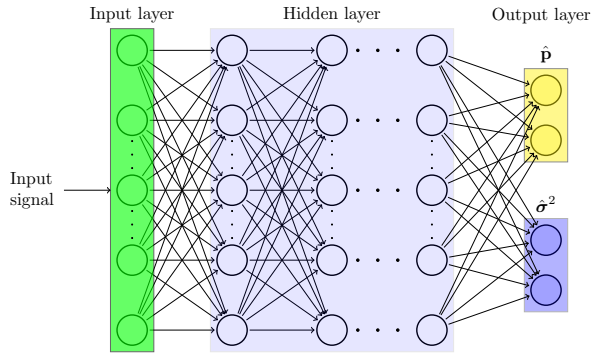


Fig. 1: A typical structure of an FCNN.

Two processes are usually involved when training a neural network, namely the forward and backward propagation processes. In the forward propagation process, the input signals enter the neural network through the input layer. Then it propagates through multiple hidden layers and ultimately reaches the output layer. At each layer, the output of a node is determined by the inputs from the previous layers, the respective weights and biases, and a non-linear activation function that is specific to that node. For example, we suppose that an FCNN has γ_i nodes in the i-th layer, the values of which are collected by a signal vector $\mathbf{x}^i = [x_1^i, ..., x_{\gamma_i}^i] \in \mathbb{R}^{\gamma_i}$. The value of the k-th node is calculated by applying a weight vector $\mathbf{w} = [w_1^{i-1}, ..., w_{\gamma_i-1}^{i-1}] \in \mathbb{R}^{\gamma_{i-1}}$ to the signal vector $\mathbf{x}^{i-1}$ in the previous layer. Specifically, x_k^i is computed as

$$x_k^i = g_i\left(\sum_{j=1}^{\gamma_{i-1}} x_j^{i-1} w_j^{i-1}\right) + b_i, \tag{4}$$

where b_i represents an optional bias term and $g_i(.)$ the activation function. The same propagation pattern is followed for each layer, generating an output vector $\boldsymbol{\nu} = [\hat{\mathbf{p}}, \hat{\boldsymbol{\sigma}}^2]$.

To train the network, it is important to select an appropriate training criterion, or the so-called loss function. A popular criterion is the mean-square error (MSE), which measures the differences between estimated localization coordinates and the ground truth labels. However, the uncertainty of the predictions is not evaluated by MSE and therefore we consider the negative log-likelihood (NLL) loss function instead [35]. Suppose that the entire training dataset contains in total N_{tr} training samples. For the i-th sample, the network outputs estimate the UE coordinate $\hat{\mathbf{p}}_i = [\hat{p}_{x_i}, \hat{p}_{y_i}]^T \in \mathbb{R}^2$ and the variance vector $\hat{\boldsymbol{\sigma}}_i^2 = [\hat{\sigma}_{x_i}^2, \hat{\sigma}_{y_i}^2]^T \in \mathbb{R}^2$, while the ground truth is $\mathbf{p}_i = [p_{x_i}, p_{y_i}]^T \in \mathbb{R}^2$. Taking into account all N_{tr} training samples, the loss function ψ is

$$\psi = \frac{1}{2N_{tr}} \sum_i \left(\frac{\log \hat{\sigma}_{x_i}^2 \hat{\sigma}_{y_i}^2}{2} + \frac{(p_{x_i} - \hat{p}_{x_i})^2}{2\hat{\sigma}_{x_i}^2} + \frac{(p_{y_i} - \hat{p}_{y_i})^2}{2\hat{\sigma}_{y_i}^2} \right). \tag{5}$$

Observe that ψ can be negative owing to the log term. After selecting the training criterion, all hyperparameters, namely all weights and bias terms in (4) in each layer, need to be fine-tuned to minimize ψ. This optimization procedure can be carried out by backward propagation, which propagates the error signal back to each neural network layer to update the weights. Due to page limitations, we avoid presenting the mathematical derivations; however, the relevant material can be found in [34].

As an evaluation procedure, we collect the test datasets and select the NLL loss as the evaluation criterion [35]. As indicated in (5), an under-confident variance estimate results in the increase of the first term, while an over-confident variance results in the increase of the second and third terms.

B. ML-based Localization Pipeline

We apply the idea of ensemble learning to the localization task. As a popular ML approach, ensemble learning targets performance improvements by training multiple base learners and then fusing their outputs [36]. Each learner itself should individually deliver decent results, and it is also important to embed enough diversity when selecting those base learners. Based on this insight, we apply the processing pipeline illustrated in Fig. 2. We select χ fingerprints and feed each fingerprint to an individual processing chain. Those fingerprints can be either the entire covariance matrix or sub-matrices (see the subarray method in the following section) or the truncated CIR. Each processing chain estimates 2-D UE coordinates as well as the variances. Suppose that the j-th processing chain estimates the position of the UE and the variance as $\hat{\mathbf{p}}_{i,j} = [\hat{p}_{x_{i,j}}, \hat{p}_{y_{i,j}}] \in \mathbb{R}^2$ and $\hat{\boldsymbol{\sigma}}^2_{i,j} = [\hat{\sigma}^2_{x_{i,j}}, \hat{\sigma}^2_{y_{i,j}}] \in \mathbb{R}^2$. By fusing all χ processing chains according to the maximum ratio combining (MRC) approach [37], $\hat{\mathbf{p}}_i$ and $\hat{\boldsymbol{\sigma}}^2_i$ are calculated as:

$$\hat{\sigma}^2_{x_i} = \frac{1}{\sum_j 1/\hat{\sigma}^2_{x_{i,j}}},\ \hat{\sigma}^2_{y_i} = \frac{1}{\sum_j 1/\hat{\sigma}^2_{y_{i,j}}} \tag{6}$$

$$\hat{p}_{x_i} = \hat{\sigma}^2_{x_i}(\sum_j \frac{1}{\hat{\sigma}^2_{x_{i,j}}}\hat{p}_{x_{i,j}}),\ \hat{p}_{y_i} = \hat{\sigma}^2_{y_i}(\sum_j \frac{1}{\hat{\sigma}^2_{y_{i,j}}}\hat{p}_{y_{i,j}}). \tag{7}$$

However, the estimated variance by (6) may be overconfident, especially when the network is overfitted. According to (6), $\hat{\sigma}^2_{x_i}$ and $\hat{\sigma}^2_{y_i}$ are less than each individual $\hat{\sigma}^2_{x_{i,j}}$ and $\hat{\sigma}^2_{y_{i,j}}$, respectively. This may increase the NLL, since $\hat{\sigma}^2_{x_i}$ and $\hat{\sigma}^2_{y_i}$ act as the denominators of the second and third terms, respectively. To address this issue, we multiply a factor χ with $\hat{\boldsymbol{\sigma}}^2_i$ to get the modified vector $\hat{\boldsymbol{\sigma}}^2_{i,mod} \in \mathbb{R}^2$, which is the harmonic averages of all estimated variances. Specifically,

$$\hat{\boldsymbol{\sigma}}^2_{i,mod} = \chi\hat{\boldsymbol{\sigma}}^2_i. \tag{8}$$

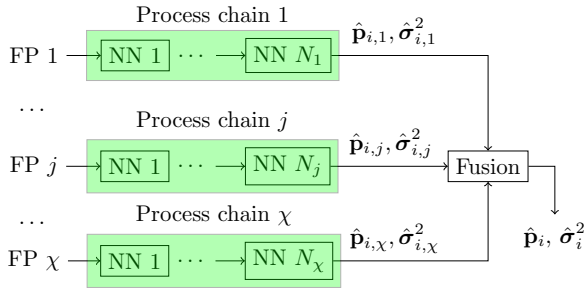


Fig. 2: The positioning neural network structure

C. Training on the subarrays

The size of the neural network increases significantly with the number of antennas, which leads to a risk of over-fitting problems. To address this problem, subarray methods can be considered. In this paper, we use the covariance matrix as an example, however, this method can be generalized to other fingerprints. We assume that an $M_1 \times M_2$ rectangular antenna array is equipped at the BS side. The spatial correlation between channel responses of two antennas is reduced to a large extent if they are separated larger than the coherence distance. Enlightened by this fact, we divide the whole antenna into I subarrays and train I neural networks instead of feeding the whole covariance matrix into the processing chain. The subarrays are selected as follows.

We define a rectangular sliding kernel with a size of N_1 rows and N_2 columns, which captures in total $N_1 N_2$ antennas. We first place the kernel in the upper left corner of the whole array, to select antennas that belong to the first N_1 rows and N_2 columns. The sliding kernel then moves S_2 columns to the right and assigns its antennas to a new group. When the sliding kernel reaches the last column, it moves S_1 rows downward, followed by moving S_2 columns to the left until the kernel hits the first column. This procedure is repeated until the entire array is scanned by the kernel and $I = (\lfloor \frac{M1-N1}{S1} \rfloor + 1)(\lfloor \frac{M2-N2}{S2} \rfloor + 1)$ training groups are formulated, where $\lfloor . \rfloor$ denotes the floor function. We then formulate I sample covariance matrices that correspond to the UE position $\mathbf{p}_i$, which are denoted as $\hat{\mathbf{C}}^1_{i,\mathcal{N}_{p_i}}, ..., \hat{\mathbf{C}}^I_{i,\mathcal{N}_{p_i}} \in \mathbb{C}^{N_1 N_2 \times N_1 N_2}$. These covariance matrices are fed into the pipeline shown in Fig. 3, to obtain the estimated UE positions and variances.

D. Training density

A fundamental question of ML-based localization is to determine the number of necessary training samples. According to the Nyquist sampling theorem, insufficient numbers of training samples result in aliasing, which has a detrimental effect on system performance. To figure out the necessary training density, we apply this theorem to investigate maximum separation distances between two adjacent training samples during the training process.

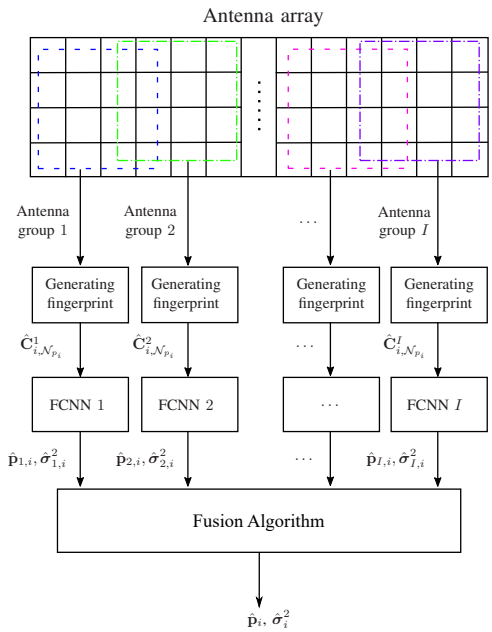


Fig. 3: The subarray method.

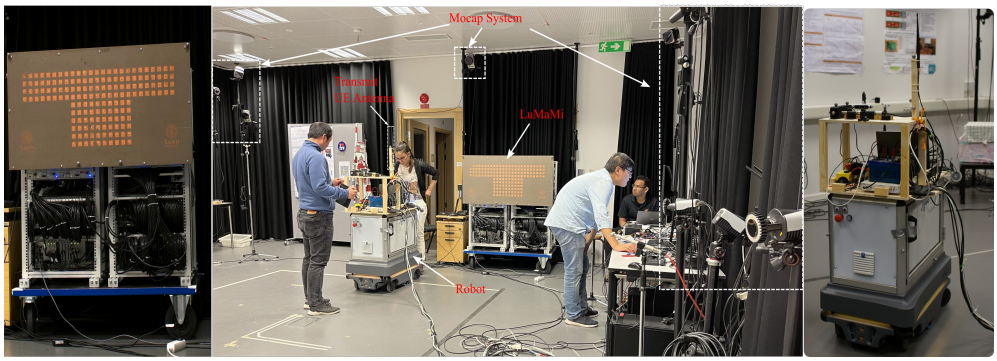
Some degree of aliasing is allowed since our task is to estimate the UE localization, rather than perfectly reconstructing the propagation channels. For convenience, we confine the scope of our approach to uniform sampling.

We consider the vector $\tilde{\mathbf{c}}_{i,\mathcal{N}_{p_i}}$ generated by (3), which varies when UE moves to Q different positions. To simplify this analysis, the UE position labels are assumed to be evenly distributed along a straight line, and the geographical distances between these Q positions are δ_d. We define a matrix $\check{\mathbf{C}} = [\tilde{\mathbf{c}}_{1,\mathcal{N}_{p_i}}, ..., \tilde{\mathbf{c}}_{Q,\mathcal{N}_{p_i}}] \in \mathbb{R}^{M^2 \times Q}$ to collect all those Q channel response vectors. By performing the 1-D discrete Fourier transform of $\check{\mathbf{C}}$ along the horizontal axis, we can formulate a matrix $\mathbf{\Psi} \in \mathbb{C}^{M^2 \times Q}$ that characterizes the channel variations along those Q positions. Specifically,

$$\mathbf{\Psi} = \check{\mathbf{C}}\mathbf{\Lambda}, \tag{9}$$

where $\mathbf{\Lambda} \in \mathbb{C}^{Q \times Q}$ is the DFT matrix. We then define a spectrum window L, which covers consecutive L columns w.r.t. the lower frequency components of $\mathbf{\Psi}$. Once L is selected, the corresponding sampling distance Δ_d between two adjacent samples can be calculated as

$$\Delta_d = Q\delta_d / L. \tag{10}$$



(a) BS LuMaMi (b) Measurement Scenario (c) UE on the robot

Fig. 4: The indoor measurement campaign

We then form those L columns into a new matrix $\mathbf{\Psi}^L \in \mathbb{C}^{M^2 \times L}$ and define η as the ratio between the Frobenius norm of $\mathbf{\Psi}^L$ and $\mathbf{\Psi}$, that is $\eta = \frac{||\mathbf{\Psi}^L||_F^2}{||\mathbf{\Psi}||_F^2}$. Here, η shows the extent of aliasing of different sampling intervals. In the following sections, we will analyze the influence of η on localization accuracy and discuss the choice of Δ_d.

IV. Massive MIMO Measurement Campaign

To validate our approach, an indoor measurement campaign was carried out in the Lund University Humanities Lab motion capture studio. Photos of the mocap studio are shown in Fig. 4. We give a brief introduction to the measurement campaign, while more details can be found in [38].

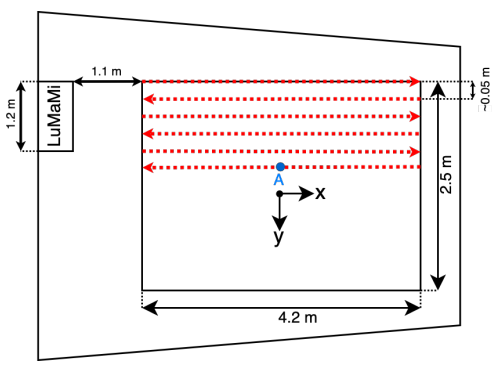


Fig. 5: Measurement arrangement in the mocap studio, the red dotted arrow shows the trajectory of the antenna.

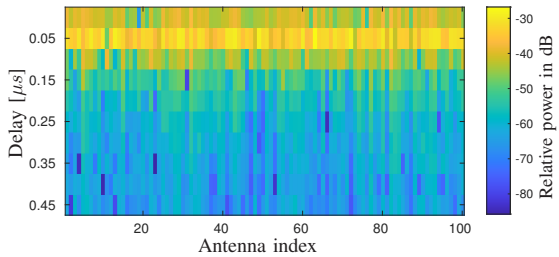
A. Introduction to the measurement campaign

In this measurement, we use a robot to carry the UE with a single dipole antenna that is placed at a height of 1.73 m. The parameter settings of our measurement system are similar to those of the LTE system. Specifically, our system occupies 20 MHz bandwidth which consists of 100 physical resource blocks (PRBs), and each PRB has 12 subcarriers. The subcarrier resource is allocated to multiple users in such a way that each UE occupies every 12 subcarrier and in total 100 subcarriers. Specific to this measurement, the UE transmits uplink pilots on the $1^{st}, 13^{th}, 25^{th}, ..., 1188^{th}$ subcarriers to estimate the uplink channel and the estimated channel responses are recorded every 10 ms. Those pilots are received by the Lund University massive MIMO testbed (LuMaMi) [39], with 100 active patch antennas operating at a center frequency of 3.7 GHz (wavelength $\lambda \approx 0.081$ m). The antennas are separated by a distance of around 4 cm (half wavelength at 3.7 GHz) in both the vertical and horizontal directions. Since our objective is to exploit more information from the *azimuth* compared to the *elevation* domain, a wide 4×25 antenna configuration is selected.

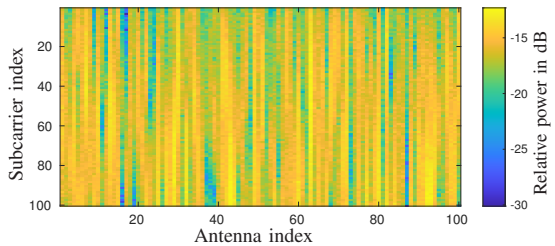
We analyzed 75 pre-defined robot trajectories, where the robot was the only moving object and all other objects were static. $\mathcal{T}_i$ channel snapshots have been recorded on the i-th trajectory, and each snapshot was represented by a matrix with dimension $M \times F$ ($M = F = 100$). A complex tensor $\mathcal{A}_i \in \mathbb{C}^{\mathcal{T}_i \times M \times F}$ was then formulated to collect all snapshots. While the robot was moving, the position of the antenna was continuously recorded every 10 ms by the Mocap system. The measurements began with locating the robot at the edge of the predefined $4.2 \times 2.5\,\mathrm{m}^2$ measurement area. The robot moved at a speed 0.1 m/s straight along the x direction; see Fig. 5. Between different measurements, the robot was moved approximately 5 cm along the y direction while maintaining its orientation. This procedure was repeated 75 times to densely scan the entire measurement area with approximately a resolution of 5 cm in the y direction and 1 mm in the x direction. When scanning the whole measurement area, we collect $\mathcal{T} = \sum_i \mathcal{T}_i = 302500$ channel snapshots. We define a tensor $\mathcal{A}' \in \mathbb{C}^{\mathcal{T} \times M \times N}$ that combines all $\mathcal{A}_i$. $\mathcal{A}'$ is then normalized by multiplying itself with a scalar so that the Euclidean norm of $\mathcal{A}'$ is equal to $\mathcal{T}MN$. All $\mathcal{T}$ collected samples are divided into two datasets, namely the training dataset with $\mathcal{X}$ samples and the testing dataset with $\mathcal{T} - \mathcal{X}$ samples. Training samples are evenly distributed with a distance along the x-axis as Δ_d. If channel samples are not selected for training purposes, they are used as testing data unless otherwise noted.

B. Measured propagation channel characteristics

One UE position is selected (position A, see Fig. 5) to illustrate the measured indoor channel properties. We present the power delay profile and the power of the transfer functions for all 100 antennas in Fig. 6. The power delay profile shows a typical indoor short-range channel characteristic: the first few delay bins contain the majority of the power in the delay domain. Such characteristics are also revealed in the transfer functions in Fig.



(a) Power delay profile



(b) Power of the transfer function

Fig. 6: Power delay profile and power of the transfer function at position A.

6, showing significant variations in channel responses among different antennas, and the frequency correlation is rather high. In contrast, the channel responses vary much smoother between different sub-carriers for every single antenna.

We evaluate the spatial correlation of the channel at different UE positions by computing the correlation coefficient $\rho(\Delta_d)$ as

$$\rho(\Delta_d) = \frac{1}{\mathcal{P}'} \sum_{\mathbf{p}_x} \left\{ \frac{\tilde{\mathbf{y}}_{\mathbf{p}_x}^H \tilde{\mathbf{y}}_{\mathbf{p}_{x+\Delta_d}}}{\sqrt{||\tilde{\mathbf{y}}_{\mathbf{p}_x}||^2 ||\tilde{\mathbf{y}}_{\mathbf{p}_{x+\Delta_d}}||^2}} \right\}, \tag{11}$$

where $\tilde{\mathbf{y}}_{\mathbf{p}_x} \in \mathbb{C}^{MN}$ is achieved by reorganizing the received channel matrix $\mathbf{Y}_{\mathbf{p}_i}$ as a vector. $\mathcal{P}'$ denotes the total number of UE positions, while Δ_d denotes the distances between two adjacent UE positions. To visualize the spatial correlation, the absolute value of $\rho(\Delta_d)$ with respect to the first UE trajectory is plotted according to (11) in Fig. 7. The separation distance Δ_d ranges from 0 to 2λ. As shown, a strong spatial correlation can be expected when $\Delta_d \leq \frac{1}{8}\lambda$, however, it decreases significantly for larger separations.

For all measurement data, the signal frequency point SNR ranges from 1dB-11dB, which depends on the distance between UE and BS and the constructive or destructive influence of small-scale fading.

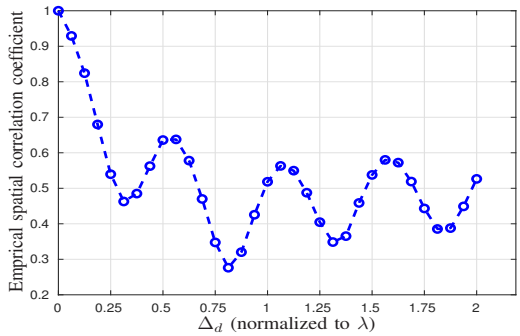


Fig. 7: Empirical spatial correlation function w.r.t one UE moving trajectory.

V. Results and Discussion

In this section, we evaluate our localization pipeline using the measurement data set. We first investigate various channel fingerprints and then demonstrate the localization accuracy gain achieved by the subarray method. Spatial spectra of estimated covariance matrices are generated, in order to further evaluate the impact of training density on the localization accuracy leveraging the Nyquist sampling theorem. Finally, we compare our approach with a classic K-nearest neighbors (KNN) based and a CNN-based algorithm [40].

A. Investigation on channel fingerprints

We first investigate two commonly used channel fingerprints, namely, the truncated CIR and the one-sample covariance matrix, which respectively capture the delay and spatial domain CSI. It is a straightforward process to generate these fingerprints. Their localization performances are compared to the case when only using the channel transfer function. To this end, we train 3 neural networks: network 1 trains on the raw received transfer function $\mathbf{Y}_{\mathbf{p}_i}$ itself; network 2 the one-sample covariance matrix of the whole array with $M = 100$ antennas; network 3 trains on the truncated CIR in the first $L_w = 10$ delay bins, which considers the limited system bandwidth (20 MHz) and the typical indoor measurement scenario with a strong line of sight (LoS) component; The frameworks of the three FCNNs are programmed based on Fig.1, which are illustrated in Table I. Since it is important to avoid the problem of vanishing gradient [41], we apply a leaky rectified linear unit (LReLU) as the nonlinear activation function at the input layer and all hidden layers as well. At the output layer, *softmax* is applied as the activation function to estimate the variances of the position, while LReLU is applied to estimate UE positions. We initially set the learning rates for the first FCNN as 10^{-5} while the second and third as 10^{-4} and all the learning rates are reduced 20% every 10 epochs. Compared to our previous work [33], we reduced the time complexity of Network 2 from $O(M^4)$ to $O(M^3)$.

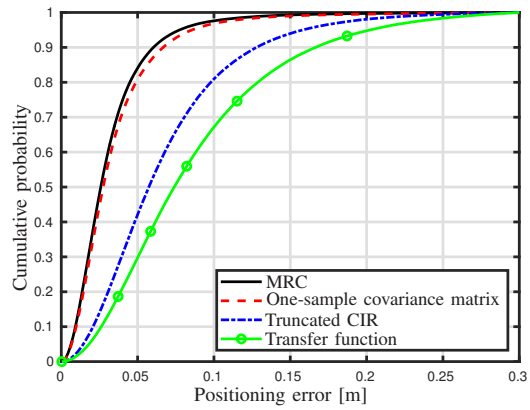
TABLE I: Parameter settings of the neural network

	Network 1	Network 2	Network 3
Input Features	Transfer function	Cov-matrix	Trunc. CIR
Input layer	$2MN \times MN$	$M^2 \times 20M$	$2ML \times ML$
Hidden layer 1	$MN \times MN/2$	$20M \times 20M$	$ML \times ML$
Hidden layer 2	$MN/2 \times MN/4$	$20M \times 20M$	$ML \times ML$
Hidden layer 3	$MN/4 \times MN/4$	$20M \times 20M$	$ML \times 512$
Hidden layer 4	$MN/4 \times 1024$	$20M \times 1024$	512×512
Hidden layer 5	1024×512	1024×512	512×256
Hidden layer 6	512×128	512×128	256×128
Hidden layer 7	128×32	128×32	128×32
Hidden layer 8	32×4	32×4	32×4
Output Layer	4×4	4×4	4×4
Batch Size	64	64	64
Epoch	200	200	200
Complexity	$O(M^2N^2)$	$O(M^3)$	$O(M^2L^2)$

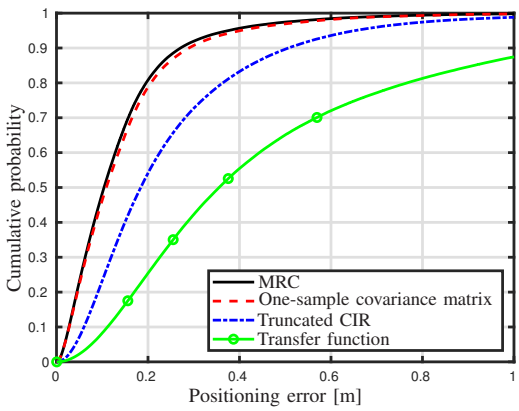
TABLE II: The NLL loss evaluated on the testing dataset

	NLL ($\frac{1}{8}\lambda$)	NLL ($\frac{3}{4}\lambda$)
Network 1	-1.95	-0.13
Network 2	-2.13	-0.43
Network 3	3.09	12.26
Fuse networks 2 and 3, using (6)	3.23	12.66
Fuse networks 2 and 3, using (8)	-0.07	4.95

Fig. 8 shows the localization accuracy of applying three networks individually, as well as the accuracy when fusing networks 2 and 3 according to (7). In Fig. 8 (a), Δ_d equals to $\frac{1}{8}\lambda$ along all 75 robot trajectories, compared to Fig. 8 (b) where $\Delta_d = \frac{3}{4}\lambda$. As presented in Fig. 8, training in truncated CIR outperforms the raw transfer function, although they embed the same CSI. We postulate that when training on truncated CIR, the reduction in network size facilitates the training process. The signal-to-noise ratio (SNR) is also enhanced if the tail part of the CIRs is truncated since this part includes only noise instead of useful CSI. Localization accuracy when training on the one-sample covariance matrix significantly outperforms the raw transfer function and the CIR, although the delay domain information is not embedded in this fingerprint. There are two potential explanations: *i)* It is challenging to resolve multipath components due to limited bandwidth and the propagation channel has a strong LoS property. *ii)* Owing to the pre-processing, the angular information can be better exploited by the neural network. The fusion algorithm results in a slight improvement in localization accuracy in comparison to using the pure one-sampled covariance matrix, since the system bandwidth is limited to 20 MHz and it is challenging to provide a good delay resolution. However, the CSI in the delay domain is



(a)



(b)

Fig. 8: Positioning error cumulative density function with respect to different training densities: (a) $\Delta_d = \frac{1}{8}\lambda$, (b) $\Delta_d = \frac{3}{4}\lambda$.

still beneficial for localization tasks even with limited bandwidth. Thus, we believe that the delay domain information can contribute more, under scenarios with rich multipath or for a system occupying wider bandwidth. Compared to Fig. 8 (a), the localization accuracy shown in Fig. 8 (b) significantly decreases. We postulate that when $\Delta_d = \frac{3}{4}\lambda$, the training density is not sufficient to represent the instantaneous channel properties.

We then calculate the NLL loss of all the aforementioned localization algorithms in the training and test dataset, and the results of the test data set are illustrated in Table II. As

TABLE III: Antenna indexes in 5 groups

subarray group number	Antenna index
I	1-8, 26-33, 51-58, 76-83
II	5-12, 30-37, 55-62, 80-87
III	11-18, 36-43,61-68, 86-93
IV	15-22, 40-47,65-72, 90-97
V	18-25, 43-50, 68-75, 93-100

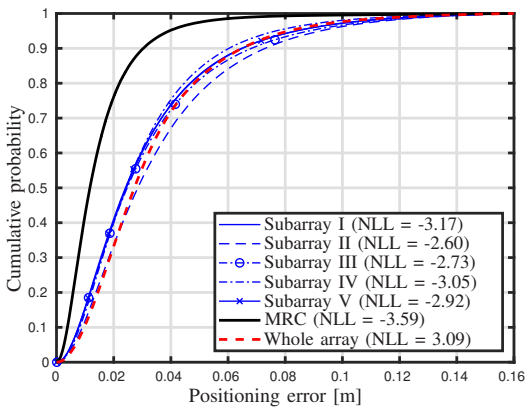
TABLE IV: Network Structure for each subarray

Layer Number	FCNN size	Layer Number	FCNN size
Input Layer 1	1024×1024	Hidden Layer 5	1024×512
Hidden Layer 2	1024×1024,	Hidden Layer 6	512×128
Hidden Layer 3	1024×1024,	Hidden Layer 7	128×32
Hidden Layer 4	1024×1024,	Output Layer 8	32×4

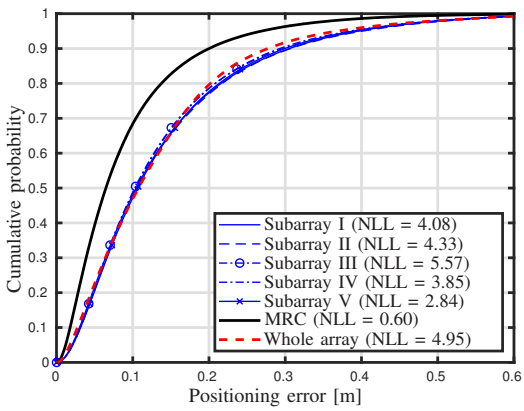
mentioned in Section III, this loss function considers the localization accuracy and the estimated variance jointly. The NLL loss of Network 3 is higher than that of the other two networks, even though it delivers better localization accuracy. We observe that the standard deviation predicted by network 3 is much smaller than the position error, which results in a significant increase of the second and third terms of (5). Based on this observation, network 3 is overconfident. This problem is even more severe if we fuse the outputs of Network 2 and Network 3 according to (6), because (6) produces a fused variance that is smaller than that of all individuals. In contrast, this problem can be alleviated by calculating the harmonic averages of the estimated variances of Network 2 and Network 3 according to (8). In the following section, we apply the subarray method to further address this overconfidence problem and focus merely on spatial channel fingerprints, considering the limited system bandwidth.

B. Enhancement by subarray method

We apply the subarray method in order to address the overconfidence problem and further enhance localization performance. Specific to this measurement setup, we consider the trade-off between complexity and localization accuracy and formulate in total 5 subarrays, and each subarray has 32 antennas ($N_1 = 4$ and $N_2 = 8$). We present the antenna indexes for each subarray in Table III. The antenna indexes are grouped in such a way that the physical distances between each antenna are close to each other; therefore, the signals captured by those antennas are strongly correlated. Note that a few antennas belong to multiple groups, and thus the spatial correlation information among antennas from different subarray groups is included as well. These subarrays are fed into 5 subnetworks that have identical network structures, which are presented in Table IV. Compared to Network 2, the size of each



(a)



(b)

Fig. 9: Positioning errors of using subarrays and the whole-array w.r.t. different training densities: (a) $\Delta_d = \frac{1}{8}\lambda$, (b) $\Delta_d = \frac{3}{4}\lambda$.

subnetwork is significantly reduced, which facilitates the training process since it is easier to avoid overfitting. For all those 5 networks, the activation functions and total training epochs are the same as in Network 2. The initial learning rates for all 5 networks are set at 2×10^{-4} and reduced 20 % every 10 epoch.

Fig. 9 compares the localization performances of the subarray method with the whole array method. The localization accuracy and the NLL loss, when Δ_d is $\frac{1}{8}\lambda$ and $\frac{3}{4}\lambda$, are shown in Fig. 9 (a) and (b), respectively. Fig. 9 shows that the localization errors of all 5 groups are close to each other, which is comparable to using the one-sample covariance

matrix of the entire array. The NLL loss w.r.t. the subarray is much lower than the whole array. This indicates that the subarray method better estimates the uncertainty. Localization performance, in terms of both accuracy and NLL loss, can be further improved by applying the MRC method to fuse the outputs of all 5 subarrays. This result illustrates the importance of selecting a proper training input, since the performance gain can be clearly seen, even if the covariance matrices of the entire array contain the same necessary information as all subarrays altogether. However, we still observe that if the training density is decreased, the localization accuracy will be degraded. Therefore, we address this problem in the following sections by first investigating the influence of training density on localization performance. At the next step, more accurate estimated covariance matrices are calculated by averaging more samples at different positions, and those matrices are applied as the training fingerprints.

C. Training density analysis using Nyquist Theorem

We apply the Nyquist Theorem to the measurement dataset to analyze the influence of training densities. This paper focuses on the covariance matrix as an example, but our method can be generally applied to other channel fingerprints. The spatial spectra of the covariance matrices with respect to different average distances ($d = 0, \frac{\lambda}{2}, 2\lambda$) for all 75 measurements are computed, according to (2)-(3) and (9). The spatial spectra of the i-th measurement w.r.t. three distances are denoted as $\mathbf{\Psi}_{i,d=0}, \mathbf{\Psi}_{i,d=\frac{\lambda}{2}}, \mathbf{\Psi}_{i,d=2\lambda} \in \mathbb{C}^{M^2 \times \mathcal{T}_i}$. To visualize the spectra, we then select $i = 75$ and plot those three spatial spectra in Fig. 10. As shown, when d increases, most of the spectral energy is concentrated in the low-frequency region, indicating that the channel changes more slowly when the UE moves to different positions. This phenomenon can be explained from a channel propagation perspective: small-scale fading is smoothed out by the averaging operation so that the swift change of channel responses cannot be observed. This allows us to further reduce the necessary training samples.

We further investigate the relationship between the Δ_d along the x-axis (see Fig. 5) and the level of aliasing noise introduced to the system. Once Δ_d increases, the captured spectrum window $L_i = \frac{\mathcal{T}_i}{\Delta_d}\delta_d$ in the i-th measurement decreases, see (10), and more aliasing noise will be introduced. To simplify the evaluation of the effect of aliasing noise, Δ_d is selected to be the same for all 75 measurement trajectories during the training phase. Based on this, we define a parameter $\eta' = \frac{\sum_i ||\mathbf{\Psi}_{i,d}^{L_i}||}{\sum_i ||\mathbf{\Psi}_{i,d}||}$, to characterize the extent of frequency aliasing. If η' is closer to 1, the aliasing noise is weak. We plot η' with regard to three covariance matrices in Fig. 11. If $d = 0$ and Δ_d exceeds the Nyquist distance $\Delta_{nqt} = \frac{1}{4}\lambda$, η' drops apparently, and the influence of aliasing noise is not trivial. Compared to $d = \frac{1}{2}\lambda$, η' drops more smoothly after $\Delta_{nqt} = 0.5\lambda$. If $d = 2\lambda$, Δ_{nqt} increases to λ. Even if Δ_d exceeds the Nyquist distance, η' drops very slowly, compared to $d = \frac{1}{2}\lambda$. This indicates that the influence of small-scale fading is rather weak.

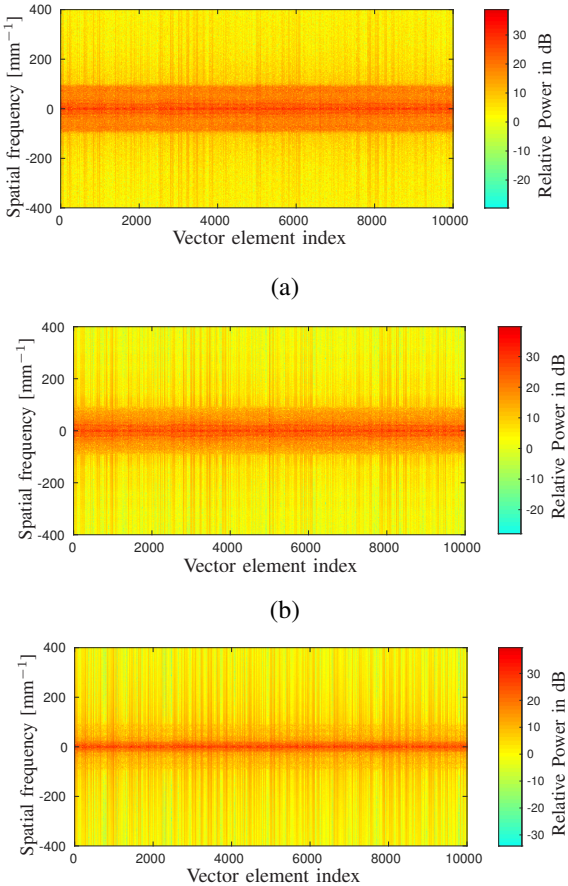


Fig. 10: Spatial spectra of covariance matrices w.r.t. different average distances: (a) $d = 0$ (one-sample covariance matrix), (b) $d = \lambda/2$ (c) $d = 2\lambda$.

D. Comparison between different covariance matrices

Fig. 12 illustrates the localization performances of three aforementioned covariance matrices with different training densities. To fairly compare performance, subarray methods are applied and antennas are grouped in the same way as in Table III. All three networks are programmed according to Table. IV. It can be observed from Fig. 12 that when Δ_d further exceeds Δ_{nqt}, positioning accuracy decreases more because the negative effect of the aliasing noise cannot be ignored. The Nyquist distance Δ_{nqt} can be extended by increasing the average distance d to cover more samples when formulating the covariance matrix. By

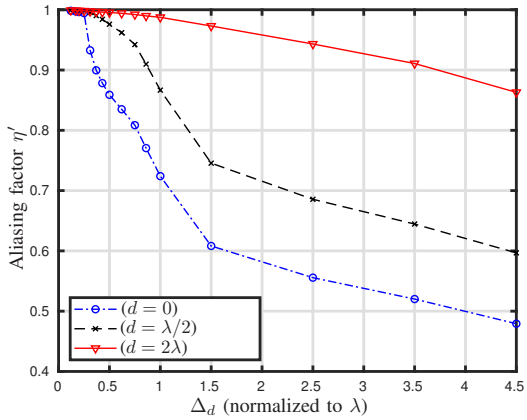


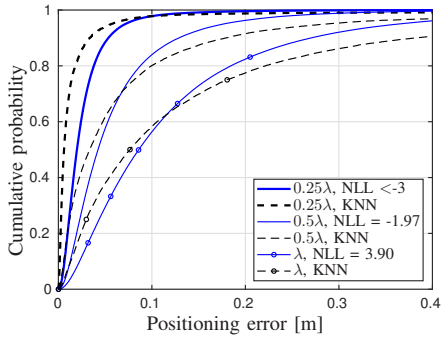
Fig. 11: η' with respect to sampling distance Δ_d.

comparing Fig. 12 (a), (b), and (c), we see that when d increases, both the localization accuracy and the NLL improve, especially under low training density. We postulate that three important factors contribute to this improvement: (1) by averaging more samples, the noise energy is reduced and the SNR is increased; (2) the system bandwidth is limited to 20 MHz, while the coherence bandwidth for the channel is around 10 MHz. Under this condition, $\mathbf{C}_i = \mathbb{E}\{\mathbf{y}_{p_i}\mathbf{y}_{p_i}^H\}$ cannot be represented by the one-sample covariance matrix since many subcarriers are still strongly correlated. However, if we consider different positions far enough from each other but within the wide-sense stationary region, their corresponding channel responses are weakly correlated. The estimated covariance matrix $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ approaches better $\mathbf{C}_i$. (3) When d is large, $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ changes much more smoothly with different positions due to the absence of small-scale fading and η' drops much more slowly. This guarantees that with the same training density, less aliasing noise is introduced to the system.

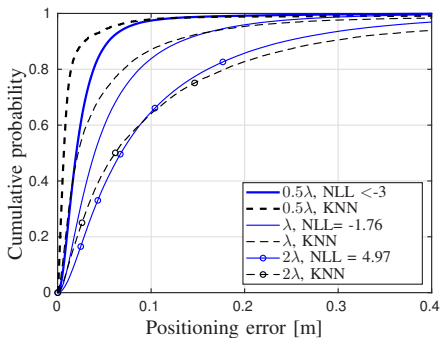
E. Comparison with other approaches

We now compare our pipeline with other two representative approaches, namely, the traditional KNN localization (naive fingerprinting) approach and the deep-learning-based approach using CNNs [40].

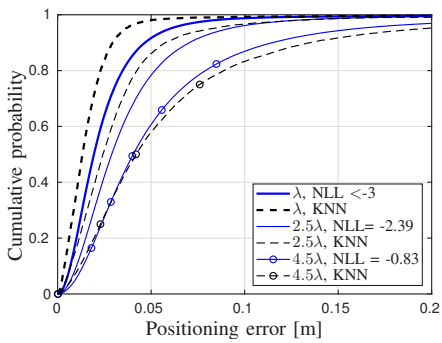
1) KNN approach: This approach first establishes a database that stores all training fingerprints. When receiving a new localization requirement, the BS finds the first K closest fingerprints from the database. In this paper, the estimated covariance matrix is selected as the fingerprint. We denote $\hat{\mathbf{C}}_{Tr,i}$ as the i-th training fingerprint stored in the database and $\hat{\mathbf{C}}_{Te}$ as the fingerprint with respect to a testing sample. We then define a scalar $l_i = ||\hat{\mathbf{C}}_{Tr,i} - \hat{\mathbf{C}}_{Te}||_F^2$. After calculating all N_{tr} distances l_i, we select the first $k = 4$ lowest l_i



(a)



(b)



(c)

Fig. 12: Comparison between positioning error of the pipeline and kNN method when using covariance matrices w.r.t. different average distances as training fingerprints. (a) $d = 0$, one-sample covariance matrix, (b) $d = \lambda/2$ (c) $d = 2\lambda$.

and denote their coordinates as $\tilde{\mathbf{p}}_i$. Applying the weighted KNN algorithm [42], the final estimated position $\breve{\mathbf{p}} \in \mathbb{R}^2$ is calculated as

$$\breve{\mathbf{p}} = \sum_{i=1}^{4} w_i \tilde{\mathbf{p}}_i, \tag{12}$$

where the weight w_i is defined as $w_i = \frac{1/l_i}{\sum_j 1/l_j}$.

Fig. 12 displays the localization accuracy of the KNN method with respect to different Δ_d. For a fair comparison, the same training data are used here as our pipeline. As a concern of the complexity issue, we randomly select 20000 testing channel samples rather than using all available ones. We observe that the KNN methods perform better than the pipeline if the channel is densely sampled. However, when the training density is reduced, the neural network method approaches and outperforms the KNN. This can be explained as follows: when the channel is heavily oversampled, it is possible to find a few pre-stored channel fingerprints in the database, which are very similar to the test channel fingerprint. The localization accuracy is already good by directly reading the coordinates of the closest fingerprints in the database, let alone the further improvement by the interpolation operation shown in (12). In comparison, the neural network mechanism estimates the UE position based on interpolating on *the whole* training datasets rather than the few closest fingerprints, which results in suboptimal localization accuracy. However, when the training density is reduced, it cannot be guaranteed that the the channel fingerprints in the database are close to the test channel fingerprint. Therefore, it only performs well if there exists such a fingerprint. In contrast, the neural network is likely more suitable for the localization task thanks to its nonlinear interpolation ability. Bear in mind that KNN methods generally require computing the Euclidean distance between the testing fingerprint and $\mathcal{N}_r$ prestored training fingerprints in total. This leads to a high complexity in time, which is $o(M^2\mathcal{N}_r)$ according to (12). In comparison, our pipeline has a better time complexity, that is $o(M^3)$, since the antenna number M is much smaller than $\mathcal{N}_r$ for most commercial devices. Even when M becomes larger, one can use the subarray method to reduce the number of antennas in each group and to reduce running time in practice. On the other hand, if one wants to achieve a better localization result using the KNN method, it is necessary to pre-store sheer numbers of measurement samples in the database. However, it is a resource-intensive endeavor to construct such very densely sampled indoor measurement datasets both in terms of finance and time manners, if the distance between adjacent samples is smaller than the Nyquist distance (only a few centimeters at sub-6 GHz frequency). For most of the applications, one would spend less resources to collect data, and thus lower densities are expected. Therefore, the usage of the KNN method for a real-time operation scenario is rather limited. From this perspective, the processing pipeline still has its advantages even under the condition of a high training density.

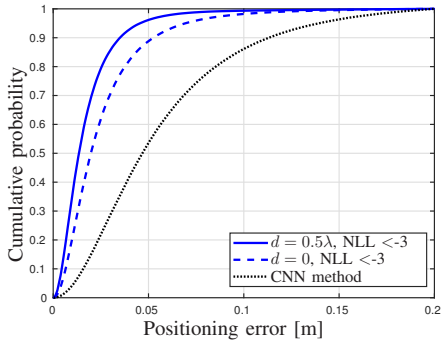


Fig. 13: Positioning errors of our pipeline and localization approach in [40] under LoS scenario.

2) CNN based approach: We then compare our localization approach with [40] that trained a deep residual CNN to perform the indoor localization tasks, where the open-source code is available. The author in [40] first formulated a tensor $\mathcal{Y}_{p_i} \in \mathbb{R}^{M\times F\times 2}$ by collecting the real and imaginary part of the raw received complex transfer function Y_{p_i}. Then Y_{p_i} was converted to the polar domain by calculating the amplitudes and phases of each entry of the received transfer function matrix. This formulated a tensor $\tilde{\mathcal{Y}}_{p_i} \in \mathbb{R}^{M\times F\times 2}$. The inverse Fourier transform was also performed to obtain the CIR matrix $\tilde{\Xi} \in \mathbb{C}^{M\times F}$ and the corresponding tensor $\hat{\mathcal{Y}}_{p_i} \in \mathbb{R}^{M\times F\times 2}$. In the next step, the author formulated a tensor $\mathcal{I} \in \mathbb{R}^{M\times F\times 6}$ by concatenating $\mathcal{Y}_{p_i}$, $\tilde{\mathcal{Y}}_{p_i}$, and $\hat{\mathcal{Y}}_{p_i}$ and sent this tensor to a residual CNN. The network structure is programmed according to [40], as well as the open source code. We modify the size of the input layer, since the antenna number in our case is 100 instead of 64. We plot the localization accuracy of this approach in Fig.13, where $\Delta_d = \frac{1}{4}\lambda$ and the training percentage is around 5%. As illustrated, our pipeline has better localization accuracy compared to [40] even with the use of a one-sample covariance matrix for training. Localization accuracy can be slightly improved if we increase the average distance d to $\frac{1}{2}\lambda$ under this training density. We postulate that our pipeline benefits from the pre-processing step as well as the subarray method. We have also performed a time complexity analysis of the convolutional neural network, which is $o(MFl_1l_2C_{in}C_{out})$, where l_1 and l_2 represent the 2-D size of the convolutional kernel and C_{in} and C_{out} the numbers of input and output channels at each layer. In comparison, the time complexity of our pipeline is $o(M^3)$. Consider that $M^2 \leq Fl_1l_2C_{in}C_{out}$ for most commercial systems, we also have advantages in terms of time complexity. Furthermore, our system has the ability to predict uncertainty, which is an additional advantage.

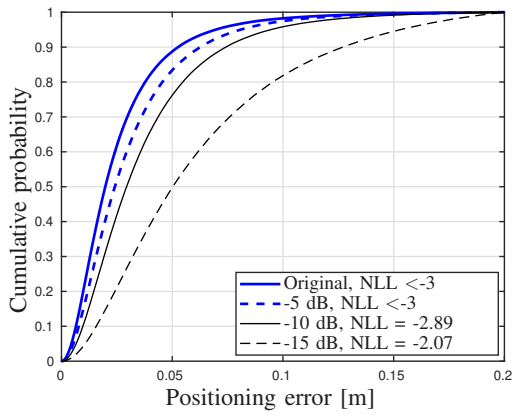


Fig. 14: Positioning errors of our pipeline under different SNR.

F. Investigating different cases

1) The influence of SNR on positioning accuracy: As stated in Section IV, the SNRs at the subcarriers range between 1 and 11 dB in our measurement scenario. To further test the performance of our localization pipeline, especially under low SNR, synthetic white Gaussian noise is added to our measurement data to emulate measurement environments with mean SNR -5 dB, -10 dB, and -15 dB. Fig. 14 illustrates the localization accuracy of our algorithm that trains on the covariance matrix ($d = \frac{1}{2}\lambda$) in different SNR scenarios. The percentage of training is around 2.5% and $\Delta_d = \frac{1}{2}\lambda$. As illustrated, our algorithm still delivers good localization performance under the -10 dB SNR scenario. Even when the SNR drops to -15 dB, the localization accuracy is still acceptable for applications such as indoor navigation. This is because our processing pipeline can harvest the SNR gain from correlating over other frequencies and averaging over other one-sample covariance matrices in the neighborhood region according to (2).

2) Investigating NLoS measurement scenarios: We investigate the localization performance of our proposed pipeline in none-line-of-sight (NLoS) measurement scenarios. To this end, our localization is applied to an open source indoor measurement dataset [43]. We provide a brief introduction to the NLoS measurement campaign and parameter settings, while more details can be found in [40]. Fig. 16 illustrates the arrangement of indoor measurement, where 4 UEs, which occupy different subcarriers, move within the four gray squares, each with a size 1.2×1.2 m. Each UE is equipped with a dipole antenna that is placed at the height of 0.4 m. The UE trajectories are densely sampled, resulting in up to 252004 channel samples with geographical distance between each sample 5 mm. The ground truth positions of UE are recorded by a mechanical device with an error of less than 1 mm. The base station consists of 64 patch antennas operating at the center frequency

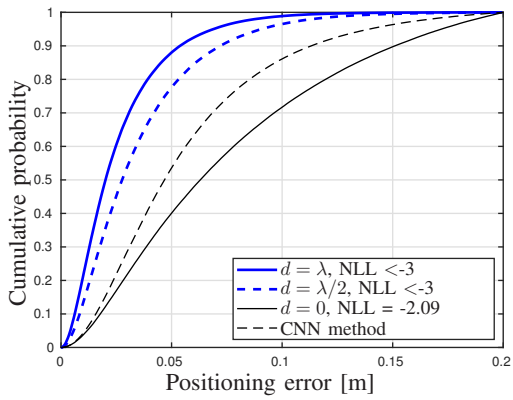


Fig. 15: Positioning errors of our pipeline and localization approach in [40] under NLoS scenario.

TABLE V: Antenna indexes in 4 groups

Subarray group number	Antenna index
I	1-6, 9-14, 17-22, 25-30, 33-38,41-46
II	2-8, 11-16, 19-24, 27-32, 35-40, 43-48
III	17-22, 25-30, 33-38, 41-46, 49-54, 57-62
IV	19-24, 27-32, 35-40, 43-48, 51-56, 59-64

2.6 GHz, and all antennas formulate a uniform rectangular array with size 0.56×0.56 m. The index of the antenna on the i -th row and the j -th column is $8(i-1)+j$. A metal blocker with size 1.6×1.3 m is placed between the base station and the UE, blocking the LoS component. Each UE sounds the OFDM signal from the uplink as a pilot for channel estimation, which has 100 subcarriers, occupying in total 20 MHz.

The subarray method is also applied and 4 groups are formulated, each group contains 36 antennas, and the antenna indices are shown in Table. V. The covariance matrices of each subarray are formulated and sent to four individual FCNNs for training. Each FCNN has the same structure as Table.IV, except the input layer size is 1296×1024. Figure.15 illustrates the localization performances of our pipeline when using the estimated covariance matrix, compared with the approach illustrated in [40]. Specifically, when we estimate the covariance matrix, three average distances are investigated, namely, $d = 0$ (one-sample covariance matrix), $d = 0.5\lambda$, and $d = \lambda$. The distances of the training samples are $\frac{1}{4}\lambda$ in both horizontal and vertical directions. As illustrated, the algorithm in [40] achieves better performance than training on the one-sample covariance matrix in the NLoS scenario. However, if the average distance d increases over $\frac{1}{2}\lambda$, the localization error decreases, and they outperform [40]. We postulate that the one-sample covariance matrix is more unstable

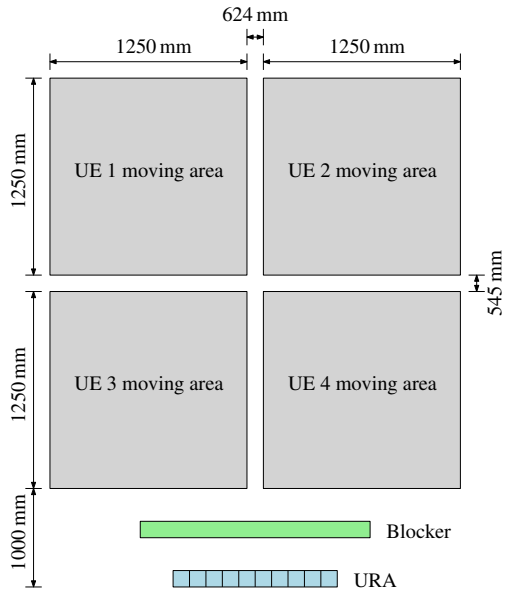


Fig. 16: A demonstration of the indoor NLoS environment.

under the NLoS scenario because the absence of a prominent LoS component amplifies the effect of small-scale fading. Therefore, in this scenario, it is challenging to collect only one channel sample to estimate the covariance matrix C_i at the position p_i. Compared to other studies, when the average distance d increases, the effect of small-scale fading becomes weak and $\tilde{\mathbf{C}}_{i,\mathcal{N}_{p_i}}$ can better approach $\mathbf{C}_i$. This shows that if the covariance matrices are selected as a training fingerprint in the NLoS scenario, it is more important to cover more samples in the neighborhood region of p_i than in the LoS scenario. On the basis of the observation above, our pipeline is also suitable for the NLoS scenario and can still achieve better performances than the literature.

3) Random selection of the training samples: We now investigate the localization performance of our pipeline, when the training data set is constructed by randomly selecting training samples from the robot trajectory. To enable a fair comparison, the network structures and all other parameters, such as the training percentages of the two datasets (evenly and randomly sampled), are the same. Specific to our data set, the percentage of training is 5% when the distance between two adjacent samples is $\Delta_d = \frac{1}{4}\lambda$. If $\Delta_d = \frac{1}{2}\lambda$, the training percentage drops to 2.5%. Fig. 17 illustrates the localization accuracy of our pipeline when we train on the one-sample covariance matrix. As shown, the localization accuracy deteriorates when the training samples are randomly selected. Therefore, the channel property may not be well captured, which deteriorates the localization performance. This example illustrates the importance of appropriately selecting training samples when

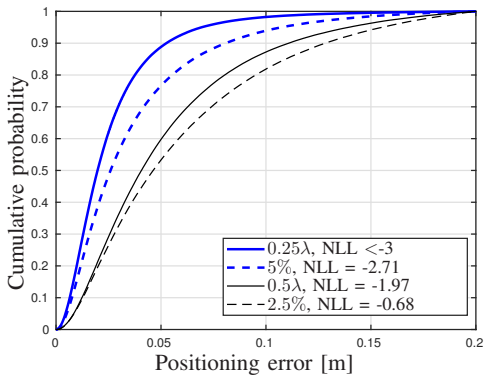


Fig. 17: Comparison between different ways of constructing the dataset.

we construct the training datasets.

VI. Conclusion and Future Work

This paper investigated the potential to apply ML to a massive MIMO system for solving localization tasks. We analyzed a novel ML-based localization pipeline, which estimated UE positions and variances by using different channel fingerprints, including covariance matrices and truncated CIR. For a system with a massive number of antennas, a subarray method was applied to facilitate the training process. Furthermore, we applied the Nyquist sampling theorem to investigate the effect of training density. An indoor massive MIMO measurement campaign was conducted at 3.7 GHz using 20 MHz bandwidth to evaluate our approaches, where centimeter-level localization accuracy has been achieved. Measurement results show that: 1) The information from both the delay and angle domains contributes to the localization performance, although in our case the delay domain CSI contributes less than the angle domain CSI due to the limited available bandwidth. 2) Compared to training on the whole antenna array, the subarray method can achieve significant enhancements in both positioning accuracy and better uncertainty prediction quality. 3) As expected, the localization accuracy decreases when the sampling interval is larger than the Nyquist sampling distances. It is worth mentioning that during the measurement campaign, the channel remained stationary and no individuals were present. In upcoming research, we will examine how the presence of people and other moving objects, as well as the difference in the properties of the UE antenna at the training and testing phase, affect the accuracy of the localization and apply transfer learning to address possible problems. We will also investigate localization pipelines that jointly process information from multiple snapshots.

Acknowledgment

The authors thank Lund University Humanities Lab and the LTH robotic lab for help with the equipment. Special thanks to Henrik Garde, Alexander Dürr, Steffen Malkowsky, Sara

Willhammar, and Sirvan Abdollah Poor for their assistance throughout the measurements. The authors also appreciate the discussion with Ph.D candidate Ziliang Xiong regarding the uncertainty prediction.

REFERENCES

[1] X. Cai, X. Cheng, and F. Tufvesson, "Toward 6G with terahertz communications: Understanding the propagation channels," *IEEE Communications Magazine, to appear*, 2023. ArXiv: 2209.07864.

[2] R. Whiton, "Cellular localization for autonomous driving: a function pull approach to safety-critical wireless localization," *IEEE Veh. Technol. Mag.*, pp. 2–11, 2022.

[3] A. A. Abdallah, C.-S. Jao, Z. M. Kassas, and A. M. Shkel, "A pedestrian indoor navigation system using deep-learning-aided cellular signals and ZUPT-aided foot-mounted IMUs," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5188–5198, 2022.

[4] M. Maaref and Z. M. Kassas, "Ground vehicle navigation in GNSS-challenged environments using signals of opportunity and a closed-loop map-matching approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2723–2738, 2020.

[5] X. Mu, Y. Liu, L. Guo, J. Lin, and R. Schober, "Intelligent reflecting surface enhanced indoor robot path planning: a radio map-based approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4732–4747, 2021.

[6] R. Whiton, J. Chen, T. Johansson, and F. Tufvesson, "Urban navigation with LTE using a large antenna array and machine learning," in *IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5.

[7] J. A. del Peral-Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of cellular mobile radio localization methods: from 1G to 5G," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 2, pp. 1124–1148, 2018.

[8] S. Dwivedi et.al, "Positioning in 5G networks," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 38–44, 2021.

[9] E. Leitinger, P. Meissner, C. Rüdisser, G. Dumphart, and K. Witrisal, "Evaluation of position-related information in multipath components for indoor positioning," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 11, pp. 2313–2328, 2015.

[10] A. Hu, T. Lv, H. Gao, Z. Zhang, and S. Yang, "An ESPRIT-based approach for 2-D localization of incoherently distributed sources in massive MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 996–1011, 2014.

[11] X. Zeng, F. Zhang, B. Wang, and K. J. R. Liu, "Massive MIMO for high-accuracy target localization and tracking," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10 131–10 145, 2021.

[12] X. Li, E. Leitinger, M. Oskarsson, K. Åström, and F. Tufvesson, "Massive MIMO-based localization and mapping exploiting phase information of multipath components," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4254–4267, 2019.

[13] L. Lian, A. Liu, and V. K. N. Lau, "User location tracking in massive MIMO systems via dynamic variational Bayesian inference," *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5628–5642, 2019.

[14] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct localization for massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, 2017.

[15] X. Cai, W. Fan, X. Yin, and G. F. Pedersen, "Trajectory-aided maximum-likelihood algorithm for channel parameter estimation in ultrawideband large-scale arrays," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 10, pp. 7131–7143, 2020.

[16] X. Cai and W. Fan, "A complexity-efficient high resolution propagation parameter estimation algorithm for ultra-wideband large-scale uniform circular array," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5862–5874, 2019.

[17] S. D. Bast, A. P. Guevara, and S. Pollin, "CSI-based positioning in massive MIMO systems using convolutional neural networks," in *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[18] S. De Bast and S. Pollin, "MaMIMO CSI-based positioning using CNNs: peeking inside the black box," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[19] J. R. Sánchez, O. Edfors, and L. Liu, "Positioning for distributed large intelligent surfaces using neural network with probabilistic layer," in *2021 IEEE Globecom Workshops (GC Wkshps)*, 2021, pp. 1–6.

[20] E. Gönültaş, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-based multi-antenna and multi-point indoor positioning using probability fusion," *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2162–2176, 2022.

[21] K. N. R. Surya Vara Prasad, E. Hossain, and V. K. Bhargava, "Low-dimensionality of noise-free RSS and its application in distributed massive MIMO," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 486–489, 2018.

[22] K. N. R. S. V. Prasad, E. Hossain, and V. K. Bhargava, "Machine learning methods for RSS-based user positioning in distributed massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8402–8417, 2018.

[23] X. Guo and N. Ansari, "Localization by fusing a group of fingerprints via multiple antennas in indoor environment," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 9904–9915, 2017.

[24] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *IEEE 28th Annual*

International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017, pp. 1–6.

[25] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-based localization for massive MIMO-OFDM system with deep convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 846–10 857, 2019.

[26] Y. Lin, K. Yu, L. Hao, J. Wang, and J. Bu, "An indoor Wi-Fi localization algorithm using ranging model constructed with transformed RSSI and BP neural network," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 2163–2177, 2022.

[27] Y. Yuan, X. Liu, Z. Liu, Z. He, and Z. Xu, "Indoor localization with wireless heterogeneous devices by composite fingerprint sets and hybrid classification," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 11, pp. 12 117–12 127, 2022.

[28] P. Ferrand, A. Decurninge, and M. Guillaud, "DNN-based localization from channel estimates: feature design and experimental results," in *IEEE Global Communications Conference*, 2020, pp. 1–6.

[29] J. Fan, X. Dou, W. Zou, and S. Chen, "Localization Based on improved sparse bayesian learning in mmWave MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 354–361, 2022.

[30] S. De Bast, E. Vinogradov, and S. Pollin, "Expert-knowledge-based data-driven approach for distributed localization in cell-free massive MIMO networks," *IEEE Access*, vol. 10, pp. 56 427–56 439, 2022.

[31] X. Ye, X. Yin, X. Cai, A. Pérez Yuste, and H. Xu, "Neural-network-assisted UE localization using radio-channel fingerprints in LTE networks," *IEEE Access*, vol. 5, pp. 12 071–12 087, 2017.

[32] R. Bharadwaj, A. Alomainy, and S. K. Koul, "Experimental investigation of body-centric indoor localization using compact wearable antennas and machine learning algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 2, pp. 1344–1354, 2022.

[33] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and F. Tufvesson, "High-precision machine-learning based indoor localization with massive MIMO system," ICC, 2023. [Online]. Available: https://arxiv.org/abs/2303.03743

[34] S. Haykin, *Neural Networks: A comprehensive foundation*, 1998.

[35] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Conference on Neural Information Processing Systems*. Curran Associates Inc., 2017, p. 6405–6416.

[36] Z. H. Zhou, *Ensemble methods foundations and algorithms*, 2012.

[37] T. P. HILL, "Conflations of probability distributions," *Transactions of the American Mathematical Society*, vol. 363, no. 6, pp. 3351–3372, 2011.

[38] I. Yaman et.al, "The LuViRA dataset: measurement description," 2023. [Online]. Available: https://arxiv.org/abs/2302.05309

[39] S. Malkowsky et.al, "The world's first real-time testbed for massive MIMO: design, implementation, and validation," *IEEE Access*, vol. 5, pp. 9073–9088, 2017.

[40] A. Colpaert, S. De Bast, R. Beerten, A. P. Guevara, Z. Cui, and S. Pollin, "Massive mimo channel measurement data set for localization and communication," *IEEE Communications Magazine*, pp. 1–7, 2023.

[41] M. Khalid, J. Baber, M. K. Kasi, M. Bakhtyar, V. Devi, and N. Sheikh, "Empirical evaluation of activation functions in deep convolution neural network for facial expression recognition," in *International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 204–207.

[42] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers - a tutorial," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, 2021.

[43] S. De Bast and S. Pollin, "Ultra dense indoor mamimo csi dataset," 2021. [Online]. Available: https://dx.doi.org/10.21227/nr6k-8r78

Paper VI

Paper VI

Reproduced, with permission, from: IEEE

G. Tian, D. Pjanić, X. Cai, B. Bernhardsson, and F. Tufvesson, "Attention-aided Outdoor Localization in Commercial 5G NR Systems," *Submitted to IEEE Transactions on Machine Learning in Communication and Networking*, Second round, after major revision.

Attention-aided Outdoor Localization in Commercial 5G NR Systems

Guoda Tian[1,2,*], Member, IEEE, Dino Pjanić[1,2,*], Student Member, IEEE,
Xuesong Cai[1], Senior Member, IEEE,
Bo Bernhardsson[3], and Fredrik Tufvesson[1], Fellow, IEEE
[1] Dept. of Electrical and Information Technology, Lund University, Sweden
[2] Ericsson AB, Sweden
[3] Dept. of Automatic Control, Lund University, Sweden
[*] Equal contribution

Abstract

The integration of high-precision cellular localization and machine learning (ML) is considered a cornerstone technique in future cellular navigation systems, offering unparalleled accuracy and functionality. This study focuses on localization based on uplink channel measurements in a fifth-generation (5G) new radio (NR) system. An attention-aided ML-based single-snapshot localization pipeline is presented, which consists of several cascaded blocks, namely a signal processing block, an attention-aided block, and an uncertainty estimation block. Specifically, the signal processing block generates an impulse response beam matrix for all beams. The attention-aided block trains on the channel impulse responses using an attention-aided network, which captures the correlation between impulse responses for different beams. The uncertainty estimation block predicts the probability density function of the user equipment (UE) position, thereby also indicating the confidence level of the localization result. Two representative uncertainty estimation techniques, the negative log-likelihood and the regression-by-classification techniques, are applied and compared. Furthermore, for dynamic measurements with multiple snapshots available, we combine the proposed pipeline with a Kalman filter to enhance localization accuracy. To evaluate our approach, we extract channel impulse responses for different beams from a commercial base station. The outdoor measurement campaign covers Line-of-Sight (LoS), Non Line-of-Sight (NLoS), and a mix of LoS and NLoS scenarios. The results show that sub-meter localization accuracy can be achieved.

This work has been funded by Ericsson AB, the Swedish Foundation for Strategic Research, and partly by the Horizon Europe Framework Programme under the Marie Skłodowska-Curie grant agreement No. 101059091.

Index Terms

5G New Radio, Sounding Reference Signal, self-attention, uncertainty estimation, radio-based positioning

I. INTRODUCTION

RADIO-based positioning is envisioned to pave the way for numerous sophisticated yet practical applications, including vehicle navigation, intelligent traffic management, and autonomous driving [1]–[7]. In contemporary 5th generation mobile network (5G) systems, there is a pronounced demand for precise localization capabilities. Currently, most localization-aware applications are facilitated by Global Navigation Satellite Systems (GNSS). However, the effectiveness of these systems is limited by many factors, such as shadowing, multipath propagation, and clock drifts between the GNSS transmitter and receiver [8]. Consequently, there is an increasing need to investigate cellular-based technologies and seamlessly integrate those techniques into existing localization systems.

Existing cellular-based localization methods can be broadly classified into two categories, namely conventional signal processing methods [7], [9]–[15], and machine learning (ML) based methods [16]–[24]. Conventional signal processing methods, such as Time of Arrival (ToA), Angle of Arrival (AoA), and Time Difference of Arrival (TDoA), require the estimation of essential channel parameters, such as signal propagation time between user equipment (UE) and base stations (BS). In the next step, the location of the UE can be estimated using these parameters. Although some of these methods have reached maturity, they can be constrained by calibration needs and algorithmic complexities [7]. On the other hand, ML methods present a promising solution but require access to data for training and a radio environment with enough unique features that can be learned. To implement an ML-based localization approach, the initial step involves obtaining various channel fingerprints, such as the raw transfer function [21], [22], received signal strength [16], angle-delay spectrum [17], [20], [23] and/or covariance matrix [18], [19]. These fingerprints then serve as input for the ML algorithms. It should be noted that an effective method of combining several different fingerprints has the potential to significantly increase the localization accuracy, see [18], [19]. ML-based localization algorithms can also be divided into two categories, namely classical ML approaches such as K-nearest neighbors (KNN) [19], Gaussian process regression [16], adaptive boosting [18], and deep learning based approaches, such as fully connected neural networks (FCNN) [22], [23], convolutional neural network [16], [18], [24], [25], and attention-aided networks [21]. In particular, the attention-aided approach holds significant promise, as its embedded attention mechanism enables ML algorithms to recognize relationships between different input feature vectors, irrespective of their actual spatial or temporal separation among those vectors. This mechanism is also the core of widely used transformer techniques, producing fruitful results in various domains such as language translations, image recognition, and speech recognition [26]. Another crucial

aspect for localization is uncertainty prediction, which is particularly important in life-critical tasks such as autonomous driving. This research problem has been initially tackled by previous works [19], [27], which provide not only the estimated location coordinates but also the corresponding variances using the negative log-likelihood (NLL) loss function.

However, to the best of our knowledge, there are still notable research gaps. Primarily, the application of attention-aided localization algorithms in 5G new radio (NR) systems represents a novel, yet unexplored, area. Secondly, the NLL uncertainty estimation technique assumes a Gaussian distribution for the estimation error of the UE position. However, such an assumption often diverges from reality. Consequently, it becomes crucial to explore further uncertainty estimation methods capable of estimating distributions other than Gaussian. To address the issues stated above, we propose a novel localization pipeline and evaluate it using data from a commercial 5G NR BS. Very few studies in the literature have been conducted on commercial grade 5G NR systems. Our research contributions are listed as follows:

- We apply attention-aided neural networks as the backbone to perform localization, we also demonstrate the advantages of this network in terms of localization accuracy.
- We apply a novel regression-by-classification method that can predict the uncertainty of localization estimates. Compared with the NLL approach, this approach provides better uncertainty estimation since it is not bounded by the assumption of Gaussianity.
- We further enhance localization accuracy by applying a Kalman filter to exploit temporal correlation between multiple channel snapshots, which smoothes the estimated trajectory.
- Finally, we verify the novel ML-powered pipeline with real measurement data obtained using a commercial 5G NR test setup, covering both Line-of-Sight (LoS) and non-Line-of-Sight (NLoS) scenarios. The results show that our approach achieves submeter-level localization accuracy.

Our initial outdoor UE localization results have been presented in the conference paper [28]. Differ from [28], we utilize a higher subchannel resolution of the UL SRS channel estimates and a high-accuracy GNSS receiver. Furthermore, we apply more advanced ML approaches such as attention mechanisms and uncertainty estimation algorithms. Compared with [28], the localization accuracies have significantly improved.

The remainder of this paper is organized as follows. Section II introduces the signal model and discusses the selected fingerprints. In Section III, we elaborate on the localization algorithms. Section IV illustrates the measurement campaign and Section V presents the results. Finally, conclusive remarks are included in Section VI.

II. System model and dataset generation

We consider a commercial 5G NR system in a single-user massive Multiple-Input Multiple-Output (MIMO) scenario, where the BS processes uplink (UL) Sounding Reference Signal

(SRS) data. The system utilizes orthogonal frequency division multiplexing (OFDM) with F subcarriers, and the SRS data is a time series of UL measurements in the beam domain. With this approach, we essentially capture the angular delay spectrum of the radio channel, an approach that has been shown to be advantageous for accurate localization based on ML [20], [29]. The BS is equipped with M_{BS} antenna ports, half of which is vertically polarized and the other half horizontally polarized, while the UE is equipped with M_{UE} antenna ports. We suppose that the number of multipath components is P, and denote $\tau_{p,t}$ as the time delay between UE and BS w.r.t. the p-th path at time t, and $\alpha_{p,m,t}$ indicates the complex coefficient of each multipath component. The BS utilizes all vertical-polarized antennas to formulate N_{V} beams, the response of the i-th beam w.r.t. the p-th path is $\beta_{\text{V},i}(\phi_p, \theta_p, f)$, where f denotes frequency, and ϕ_p and θ_p represent the azimuth and elevation arrival angles for the p-th multipath, respectively. Similarly, another N_{H} set of beams uses all horizontal polarized antennas, and the response of the i-th beam is $\beta_{\text{H},i}(\phi_p, \theta_p, f)$. For the m-th UE port, the propagation channel model for each beam at time index t can be formulated as

$$\begin{aligned} h_{\text{V},i,m,t}(f) &= \sum_{p=1}^{P} \beta_{\text{V},i}(\phi_p, \theta_p, f)\, \alpha_{p,m,t}\, \exp\{-j2\pi f \tau_{p,t}\} \\ h_{\text{H},i,m,t}(f) &= \sum_{p=1}^{P} \beta_{\text{H},i}(\phi_p, \theta_p, f)\, \alpha_{p,m,t}\, \exp\{-j2\pi f \tau_{p,t}\}. \end{aligned} \tag{1}$$

By collecting all $h_{\text{V},i,m,t}(f)$ and $h_{\text{H},i,m,t}(f)$ for the F subcarriers, we can formulate two beam space matrices of the channel transfer function (CTF), $\mathbf{H}_{\text{V},m,t} \in \mathbb{C}^{N_V \times F}$ and $\mathbf{H}_{\text{H},m,t} \in \mathbb{C}^{N_{\text{H}} \times F}$ at time t, which correspond to the vertical and horizontal polarized antenna groups, respectively. We further define matrix $\mathbf{H}_t \in \mathbb{C}^{N \times F} = \left[\mathbf{H}^T_{\text{H},1,t}, \mathbf{H}^T_{\text{V},1,t}, ..., \mathbf{H}^T_{\text{H},M_{\text{UE}},t}, \mathbf{H}^T_{\text{V},M_{\text{UE}},t}\right]^T$ that combines channel matrices of all UE antenna ports. Specifically, $N = M_{\text{UE}}(N_{\text{H}} + N_{\text{V}})$. This matrix depends strongly on the UE position, therefore they can be selected as raw channel fingerprints to perform ML-based localization.

III. The ML-based localization approach

In this paper, our study focuses on car navigation applications, where two-dimensional (2-D) localization is adequate for most scenarios. However, a similar approach can be extended to three-dimensional (3-D) coordinate-based localization by altering the dimension of the output layer in our neural network. The ML-based localization pipeline, as described in Fig. 1, consists of five sequential blocks. First, the raw CTF $\mathbf{H}_t$ is fed into a data cleaning block to evaluate the validity of the input data. After this, valid CTFs are forwarded to a digital signal processing block to generate an impulse response beam matrix $\mathbf{G}_t \in \mathbb{C}^{N \times F}$. The amplitudes in this matrix then serve as input to a deep neural network, which incorporates a self-attention mechanism at its core. The network's final layer outputs an estimated probability density function (PDF) representing the location, thereby facilitating

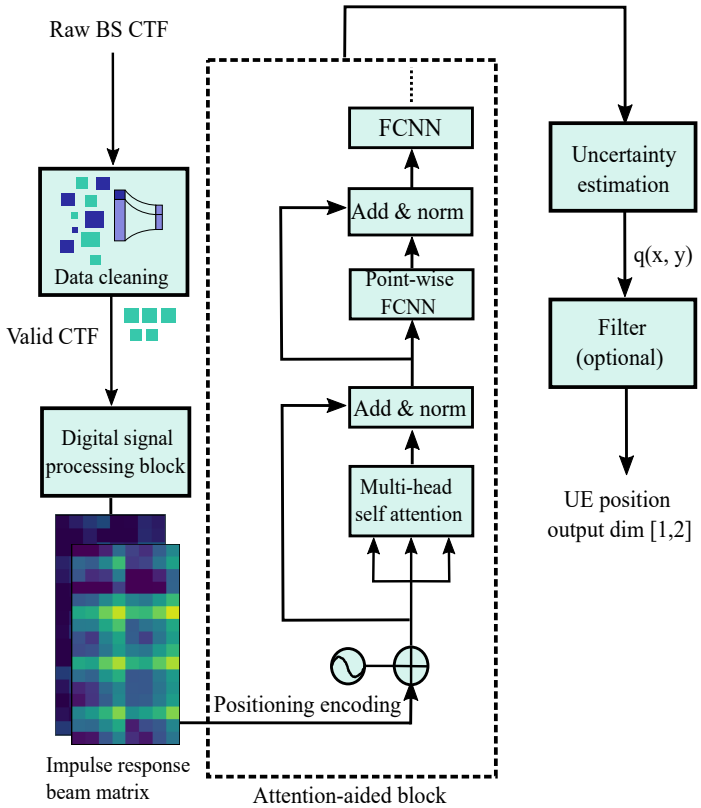


Fig. 1: The ML-based localization pipeline for the 5G NR system.

uncertainty estimation. To further enhance localization accuracy, a filter may be applied after the final layer of the pipeline, provided that information from multiple snapshots is available.

A. The attention mechanism

1) Fundamental basics of the attention operation: An example of the attention block is illustrated in Fig. 2, which takes a matrix $\mathbf{X} \in \mathbb{R}^{A_1 \times A_2}$ as the input, generating the output matrix $\mathbf{Z} \in \mathbb{R}^{A_1 \times A_3}$. Initially, $\mathbf{X}$ are multiplied by three matrices, namely, the query matrix $\mathbf{W}_q \in \mathbb{R}^{A_2 \times A_3}$, the key matrix $\mathbf{W}_k \in \mathbb{R}^{A_2 \times A_3}$ and the value matrix $\mathbf{W}_v \in \mathbb{R}^{A_2 \times A_3}$. The multiplication operations yield three matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{A_1 \times A_3}$, specifically,

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_q,\ \mathbf{K} = \mathbf{X}\mathbf{W}_k,\ \mathbf{V} = \mathbf{X}\mathbf{W}_v. \tag{2}$$

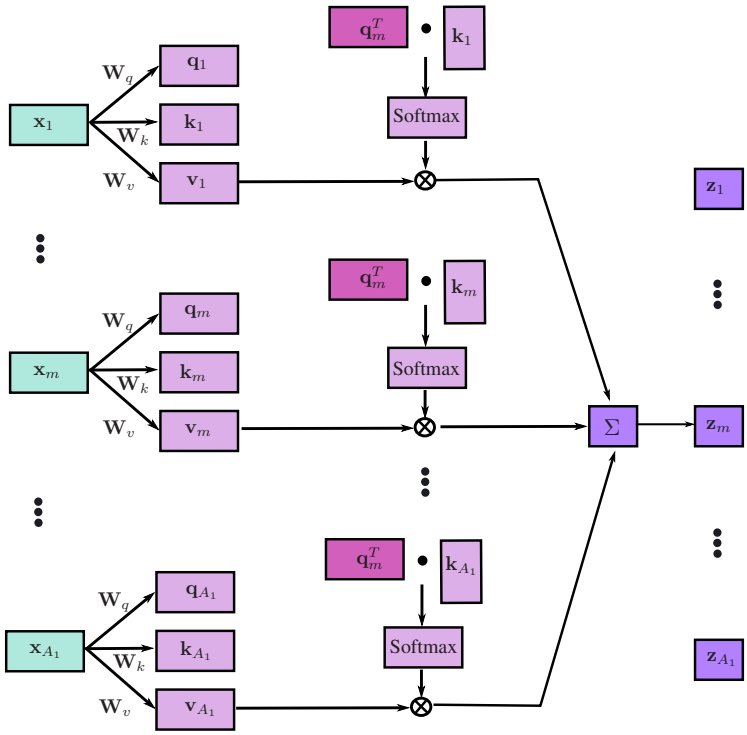


Fig. 2: An illustration of basic attention mechanism to generate $\mathbf{z}_j$ and same mechanism can be applied to generate $\mathbf{Z}$.

In the self-attention mechanism, the query ($\mathbf{Q}$) and key $\mathbf{K}$ matrices play a crucial role in determining the relevance of each row vector in the matrix $\mathbf{X}$ to the other row vectors. The elements of these three matrices act as hyperparameters that can be fine-tuned during the training process. The second step is to calculate the pairwise correlations between all columns of matrices $\mathbf{Q}$ and $\mathbf{K}$, resulting in a new matrix $\mathbf{A} \in \mathbb{R}^{A_3 \times A_3}$, specifically,

$$\mathbf{A} = \frac{1}{\sqrt{A_2}} \mathbf{Q}^T \mathbf{K}. \tag{3}$$

These correlations reflect the similarities between each pair of row vectors in $\mathbf{X}$. We then apply the *softmax* operation to normalize $\mathbf{A}$ and obtain another matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{A_3 \times A_3}$. Each element $\tilde{\mathbf{A}}_{i,j}$ is positive and the sum of all the elements in each column is equal to 1. Specifically, $\tilde{\mathbf{A}}_{i,j}$ is calculated as

$$\tilde{\mathbf{A}}_{i,j} = \frac{\exp \mathbf{A}_{i,j}}{\sum_k \exp \mathbf{A}_{i,k}}. \tag{4}$$

Finally, the output matrix $\mathbf{Z}$ is calculated as

$$\mathbf{Z} = \mathbf{V}\,\tilde{\mathbf{A}}, \tag{5}$$

where each column of $\mathbf{Z}$ represents a weighted sum, and the weights are determined by the corresponding column in $\tilde{\mathbf{A}}$. In addition to the fundamental attention operation, we further introduce the *multi-head attention* mechanism that can improve model capabilities. This mechanism employs a total of $\mathcal{P}$ attention heads, each associated with sets of query matrices $(\mathbf{W}_q^1, ..., \mathbf{W}_q^{\mathcal{P}})$, key matrices $(\mathbf{W}_k^1, ..., \mathbf{W}_k^{\mathcal{P}})$, and value matrices $(\mathbf{W}_v^1, ..., \mathbf{W}_v^{\mathcal{P}})$. The multi-head attention mechanism operates in $\mathcal{P}$ steps. In the initial step, the matrices $\mathbf{W}_q^1, \mathbf{W}_k^1, \mathbf{W}_v^1$ are applied to the input matrix $\mathbf{X}$ following equations (2)-(5), resulting in the output $\mathbf{Z}_1 \in \mathbb{R}^{A_1 \times A_3}$. This process is then repeated $\mathcal{P} - 1$ times, generating additional output matrices $\mathbf{Z}_2, ..., \mathbf{Z}_P \in \mathbb{R}^{A_1 \times A_3}$. Finally, we concatenate all output matrices obtained from each step, formulating a matrix $\mathbf{Z}_{tl} \in \mathbb{R}^{A_1 \times \mathcal{P} A_3}$. The final output matrix $\mathbf{Z}' \in \mathbb{R}^{A_1 \times A_2}$ can then be expressed as

$$\mathbf{Z}' = \mathbf{Z}_{tl}\,\mathbf{W}_O, \tag{6}$$

where $\mathbf{W}_O \in \mathbb{R}^{\mathcal{P} A_3 \times A_2}$ is another hyperparameter matrix.

2) Positioning encoding: It is important to note that the attention mechanism neglects the inherent sequence order of the input vectors in $\mathbf{X}$. Consequently, when employing such a mechanism, particularly for tasks dependent on the order of vector arrangement, it is imperative to apply a *positioning encoding* technique to incorporate and preserve this sequential information. The idea of positioning encoding is to add another fixed matrix $\mathbf{X}_k \in \mathbb{R}^{A_1 \times A_2}$ to $\mathbf{X}$ [26], a standardized positioning encoding matrix $\mathbf{X}_k$ is

$$\begin{aligned} \mathbf{X}_k(x, y) &= \sin\left(\frac{x}{10000^{y/A_2}}\right), \text{for odd } y; \\ \mathbf{X}_k(x, y) &= \cos\left(\frac{x}{10000^{(y-1)/A_2}}\right), \text{for even } y. \end{aligned} \tag{7}$$

The matrix $\mathbf{X}_k$ is fixed and will not be fine-tuned during the training process. The advantages of using cosine and sine structures are as follows:

- The values of the sine and cosine functions are bounded between -1 and 1, providing stable input magnitudes for the model.
- The smooth variation of sine and cosine functions allows the model to capture gradual changes in positions.
- The use of sine and cosine functions, as given by Eq. (7), ensures that each position is uniquely encoded.

3) Residual mechanism, Layer normalization and position-wise FCNN : After collecting the matrix $\mathbf{Z}'$, we add the input matrix $\mathbf{X}$ to $\mathbf{Z}'$ to obtain the matrix $\tilde{\mathbf{Z}} \in \mathbb{R}^{A_1 \times A_2}$. We apply the *residual mechanism* since it preserves the original information of the input matrix. The matrix $\tilde{\mathbf{Z}}$ is then fed to a *layer normalization block*, which first vectorizes $\tilde{\mathbf{Z}}$ into a vector

$\tilde{\mathbf{z}} \in \mathbb{R}^{A_1 A_2}$. Subsequently, each element $\tilde{z}_i$ in $\tilde{\mathbf{z}}$ is scaled to derive a new vector $\hat{\mathbf{z}} \in \mathbb{R}^{A_1 A_2}$ as in [26], specifically,

$$\hat{z}_i = \gamma \frac{\tilde{z}_i - \mu}{\sigma} + \beta, \tag{8}$$

where μ and σ^2 represent the mean and variance of vector $\tilde{\mathbf{z}}$. The parameters γ and β denote the amplitude scaling and the bias, respectively. By default, $\gamma = 1$ and $\beta = 0$, although these parameters can be adjusted as learning hyperparameters. We then reformulate $\hat{\mathbf{z}}$ into a matrix $\hat{\mathbf{Z}} \in \mathbb{R}^{A_1 \times A_2}$. To enhance the capacity to capture nonlinear relationships, we feed the output matrix $\hat{\mathbf{Z}}$ into a pointwise FCNN to get $\hat{\mathbf{Z}}' \in \mathbb{R}^{A_1 \times A_2}$ [26], specifically,

$$\hat{\mathbf{Z}}' = \mathbf{W}_2 f_{\text{Relu}}(\mathbf{W}_1 \hat{\mathbf{Z}} + \mathbf{B}_1) + \mathbf{B}_2, \tag{9}$$

where $f_{\text{Relu}}(.)$ represents the rectifier activation function, and $\mathbf{W}_1, \mathbf{W}_2, \mathbf{B}_1, \mathbf{B}_2$ are hyperparameter matrices, and the bias matrices $\mathbf{B}_1, \mathbf{B}_2$ are optional. After collecting $\hat{\mathbf{Z}}'$, we apply the same residual mechanism and layer normalization to derive $\breve{\mathbf{Z}} \in \mathbb{R}^{A_1 \times A_2}$. Finally, $\breve{\mathbf{Z}}$ is vectorized and fed into another FCNN. Such an operation can also help to match the vector sizes for possible subsequent blocks.

B. Data cleaning and signal processing

The collection of UL SRS channel measurements in a commercial 5G NR BS builds limitations when retrieving data-intense structures such as SRS channel measurement samples. The vast amounts of SRS data generated at milliseconds level are normally enclosed within the baseband entity of a BS and primarily intended for internal processing, whereas external access to these data sets may be compromised by hardware and software restrictions. To mitigate these challenges, it is essential to equip our pipeline with the ability to discern the validity of the input data. As retrieving all the necessary data in a complete format has been challenging, we implemented a threshold that defines a cut-off point for discarding datasets when insufficient information has been retrieved from the BS and introduced a *data-cleaning* block to pre-process the measurement data. Its primary objective is to determine whether the raw transfer function is valid or invalid. A raw transfer function is labeled invalid under the following conditions:

- Insufficient CSI in the beam or frequency domain: the number of non-zero elements in $\mathbf{H}_t$ is lower than a given threshold.
- Update failure: the values of all subcarriers or all beams remain the same.

After filtering out all invalid data, the next step is to process the raw CTF to generate impulse response beam matrices. To suppress the side lobes, we apply Hann windowing across all rows of the matrix $\mathbf{H}_t$ to obtain matrix $\hat{\mathbf{H}}_t \in \mathbb{C}^{N \times F}$. The F-length Hann window in the frequency domain is given by

$$w[f] = \sin^2\left(\frac{\pi f}{F}\right), \quad f = 0, \ldots, F-1. \tag{10}$$

After the windowing operation, the impulse response beam matrix $\mathbf{G}_t$ is produced by performing the inverse discrete Fourier transform along each row of $\hat{\mathbf{H}}_t$. Given the potential difficulty in achieving a stable phase for $\mathbf{G}_t$, here we opt to use its amplitude $|\mathbf{G}_t|$ as the training fingerprint, although this means throwing away potentially useful information.

C. Single-snapshot localization

We hereby introduce our single-snapshot localization approach, which focuses on performing the localization task using only a single channel sample of the received transfer function, generated at an SRS reporting periodicity of 20 ms containing 64 symbols, transformed from antenna to beam space. The proposed positioning model analyzes the time series of these samples. As illustrated in Fig. 1, the architecture comprises multiple attention-aided blocks, followed by an output layer that has three alternatives corresponding to three loss functions, namely the Mean Square Error (MSE), NLL, and Regression-by-classification loss functions. We use $\mathbf{p}_i = [p_{x,i}, p_{y,i}]^T$ to represent the 2-D ground truth of the moving UE at the i-th position. Notably our approach can be readily adapted for 3-D localization.

1) Alternative 1: MSE loss function: This approach directly estimates the UE locations by setting a 2-D regression head at the output layer of the last attention block. Let $f_{\text{MSE}}(.)$ denote the overall function and vector $\boldsymbol{\theta}_2$ all hyperparameters, $\hat{\mathbf{p}}_i = [\hat{p}_{x,i}, \hat{p}_{y,i}]^T$ the estimated i-th UE locations generated by $f_{\text{MSE}}(\boldsymbol{\theta}_2, |\mathbf{G}_t|)$, the loss Ψ_1 can be expressed as

$$\Psi_1 = \frac{1}{N_{tr}} \sum_{i \in \Omega'_{tr}} ||\mathbf{p} - \hat{\mathbf{p}}||_F^2, \tag{11}$$

where Ω'_{tr} and N_{tr} denote the training set and the number of training samples, respectively, and $||.||_F$ denotes the Frobenius matrix norm.

2) Alternative 2: NLL loss function: Unlike the first approach, this method employs the NLL criterion, which models the estimated UE position as a multivariate Gaussian distribution defined by its mean $\breve{\mathbf{p}} = [\breve{p}_{x_i}, \breve{p}_{y_i}]^T$ and variance $\breve{\boldsymbol{\sigma}}_i^2 = [\breve{\sigma}_{x_i}^2, \breve{\sigma}_{y_i}^2]^T$. Consequently, a 4-dimensional regression head is required at the output layer. Similar to [19], the NLL loss Ψ_2 is expressed as

$$\Psi_2 = \frac{1}{2N_{tr}} \sum_{i \in \Omega'_{tr}} \Big(\frac{\log \breve{\sigma}_{x_i}^2 \breve{\sigma}_{y_i}^2}{2} + \frac{(p_{x_i} - \breve{p}_{x_i})^2}{2\breve{\sigma}_{x_i}^2} + \frac{(p_{y_i} - \breve{p}_{y_i})^2}{2\breve{\sigma}_{y_i}^2} \Big). \tag{12}$$

3) Alternative 3: Regression-by-Classification (RbC): The core of this approach [30], [32] lies in converting a regression task to a classification task. This is achieved by first defining a feasible range for the target parameter and then dividing this range into discrete bins. For the localization task, the lower and upper bounds of the UE x-coordinates are denoted as $B_{lw,x}$ and $B_{up,x}$, respectively. Similarly, $B_{lw,y}$ and $B_{up,y}$ represent the bounds for

the y-coordinates. To accomplish this discretization, we divide the x-coordinate range into L_x equally sized bins. The y-coordinate range is divided into L_y bins in a similar fashion. For each bin, we denote $\bar{l}_{x,k}$ and $\bar{l}_{y,k}$ as the lower endpoint values of the k-th interval for the x- and y-coordinates, respectively.

Unlike the NLL method, RbC does not inherently model the output probability as a Gaussian distribution. Instead, it estimates the probability and bias values of each bin for both the x- and y-coordinates. The bias value can be used to reduce the quantization error. To this end, in total 4 vectors are generated: the probability vectors $\boldsymbol{\omega}_x \in \mathbb{R}^{L_x}$ and $\boldsymbol{\omega}_y \in \mathbb{R}^{L_y}$, as well as the deviation vectors $\mathbf{d}_x \in \mathbb{R}^{L_x}$ and $\mathbf{d}_y \in \mathbb{R}^{L_y}$. Note that $\boldsymbol{\omega}_x$ refers to the probability vector corresponding to a specific position, and there are N_{tr} instances of $\boldsymbol{\omega}_x$ when considering the entire training dataset; the same applies to $\boldsymbol{\omega}_y$. It is crucial to apply a *softmax* operation as shown in (4) when generating $\boldsymbol{\omega}_x$ and $\boldsymbol{\omega}_y$ to ensure that the elements within each vector sum to 1. One special case for deviation vectors is when all L_x elements in $\mathbf{d}_x$ have the same value, and the same for $\mathbf{d}_y$. In other words, a uniform shift is applied to the probability density function, which also aids in the reduction of the output vector dimensions. We denote $\omega_{x,k}$ and $d_{x,k}$ as the k-th elements of $\boldsymbol{\omega}_x$ and $\mathbf{d}_x$, similarly for $\omega_{y,k}$ and $d_{y,k}$. Inspired by [30], the η-norm loss Ψ_3^η is formulated as

$$\Psi_3^\eta = \frac{1}{2N_{tr}} \sum_{i \in \Omega'_{tr}} \Big(|| \sum_{j=1}^{L_x} \omega_{x,j,i} \bar{l}_{x,j,i} - p_{x,j,i} + d_{x,j,i} ||^\eta + || \sum_{j=1}^{L_y} \omega_{y,j,i} \bar{l}_{y,j,i} - p_{y,j,i} + d_{y,j,i} ||^\eta + \gamma_1 ||\mathbf{d}_x|| + \gamma_2 ||\mathbf{d}_y|| \Big). \tag{13}$$

Here, η is usually chosen as $\eta = 1$ or $\eta = 2$, which corresponds to the *Taxicab* and *Euclidean* norms, respectively. Two penalty terms, $\gamma_1 ||\mathbf{d}_x||$ and $\gamma_2 ||\mathbf{d}_y||$, are added to the cost function. The estimated coordinate $\hat{\mathbf{p}}_i^{\text{RbC}} = [\hat{p}_{x,i}^{\text{RbC}}, \hat{p}_{y,i}^{\text{RbC}}] \in \mathbb{R}^2$ is then given by

$$\begin{aligned} \hat{p}_{x,i}^{\text{RbC}} &= \sum_j \omega_{x,j,i} \bar{l}_{x,j,i} + d_{x,j,i}, \\ \hat{p}_{y,i}^{\text{RbC}} &= \sum_j \omega_{y,j,i} \bar{l}_{y,j,i} + d_{y,j,i}. \end{aligned} \tag{14}$$

4) Comparison between different uncertainty estimates: Our previous work [19] used the NLL score in the test data set to assess the effectiveness of uncertainty estimation. However, applying the same criterion to evaluate the RbC method presents challenges because of the non-Gaussian nature of its output. To address this challenge, another criterion named Area Under the Sparsification Error (AUSE) [31], [33] is used. Sparsification is a way to assess the quality of uncertainty estimates. It works by progressively discarding fractions of the predictions that the model is most uncertain about and verifying whether this corresponds to a proportional decrease in the remaining average endpoint error. To calculate AUSE, the

first step is to compute the discrete entropy u_H based on the predicted probability. In the following discussion, we illustrate this process using the predicted $\boldsymbol{\omega}_{x,i}$ vector for the x-coordinate as an example, noting that the result can be readily extended to the y-coordinate. The entropy $u_{H,x,i}$ for $\boldsymbol{\omega}_{x,i}$ is given by [32]

$$u_{H,x,i}(\boldsymbol{\omega}_{x,i}) = -\sum_{k=1}^{L_x} \omega_{x,k,i} \log \omega_{x,k,i}. \tag{15}$$

To enable a fair comparison between the NLL and RbC methods, we need to discretize the predicted Gaussian distributions determined by $\breve{\mathbf{p}}$ and $\breve{\sigma}_i^2$. To this end, the x-axis is segmented into L_x bins. As detailed in [34], the value for the k-th bin of the discretized function, denoted $\breve{p}_{x,k}$, is calculated as

$$\breve{p}_{x,k}^{ds} = \frac{\frac{1}{\breve{\sigma}_k} \exp(-\frac{(\breve{p}_{x,k}-\bar{l}_k)^2}{2\breve{\sigma}_k^2})}{\sum_j \frac{1}{\breve{\sigma}_j} \exp(-\frac{(\breve{p}_{x,j}-\bar{l}_j)^2}{2\breve{\sigma}_j^2})}. \tag{16}$$

We now organize the discrete entropies for the N_{ts} testing samples calculated from (15) in descending order to form the vector $\mathbf{u}_{H,x} \in \mathbb{R}^{L_x}$. Similarly, we calculate the absolute errors between the estimated values $\hat{p}_{x,i}^{\text{WS}}$ and the ground truth p_x for all testing samples, arranging these errors in descending order to create the vector $\boldsymbol{\xi}_x \in \mathbb{R}^{L_x}$. Let ξ_{max} be the maximum absolute error. We scale all elements in $\mathbf{u}_{H,x}$ by a factor such that the first element of the resulting vector $\hat{\mathbf{u}}_{H,x}$ equals ξ_{max}.

Next, we define a *sparsification* function $s(\varphi)$, which is calculated by removing the initial φ-fraction of samples from $\hat{\mathbf{u}}_{H,x}$ and averaging the remaining data, with φ ranging from 0 to 1. A similar process is applied to $\boldsymbol{\xi}_x$, which yields the oracle function $g(\varphi)$. Finally, AUSE is calculated as

$$\text{AUSE} = \int_0^1 |s(\varphi) - g(\varphi)| d\varphi, \tag{17}$$

which represents the area between the sparsification and the oracle curves. A smaller area indicating a better uncertainty estimator.

D. Kalman-Filter-based trajectory smoothing

To further improve the localization accuracy, we exploit the temporal correlation between successive positions by applying a Kalman filter as a straightforward method for trajectory smoothing. The BS can select the appropriate motion model based on several factors, including the sampling rate of channel snapshots, the vehicle's velocity, and the availability of velocity or acceleration parameters. In this paper, we introduce the constant velocity model as the simplest option. This model is effective for scenarios with low vehicle speed and a high sampling rate. However, the same concepts can be extended to more advanced models that account for changes in velocity or even acceleration. While these advanced

models may deliver better performance, especially in high-speed scenarios, they also require more complex hardware. More detailed information see be refered to [35].

We define a vector $\boldsymbol{\xi}_t \in \mathbb{R}^4 = [p_{x,t}, v_{x,t}, p_{y,t}, v_{y,t}]^T$ to represent the UE position and velocity at time t, where $v_{x,t}$ and $v_{y,t}$ denote the speed in the x and y-directions, respectively. The state-space model for the UE is given by

$$\boldsymbol{\xi}_t = \mathbf{F}\boldsymbol{\xi}_{t-1} + \boldsymbol{\lambda}_t, \tag{18}$$

where $\mathbf{F} \in \mathbb{R}^{4\times4}$ denotes the state-transition matrix, while $\boldsymbol{\lambda}_t \in \mathbb{R}^4$ the additive noise. Specifically,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta_t & 0 \\ 0 & 1 & 0 & \Delta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{19}$$

where Δ_t denotes the time differences between snapshots. We then define $\boldsymbol{\Xi}_t \in \mathbb{R}^{4\times4}$ as the covariance matrix of $\boldsymbol{\xi}_t$. The relationship between $\boldsymbol{\Xi}_t$ and $\boldsymbol{\Xi}_{t-1}$ can be written as

$$\boldsymbol{\Xi}_t = \mathbf{F}\boldsymbol{\Xi}_{t-1}\mathbf{F}^T + \boldsymbol{\Lambda}, \tag{20}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{4\times4}$ is the covariance matrix of the noise vector $\boldsymbol{\lambda_t}$. We further denote $\breve{\mathbf{p}}_t \in \mathbb{R}^2 = [\breve{p}_{t,x}, \breve{p}_{t,y}]$ as the predicted UE position and express the observation model as

$$\breve{\mathbf{p}}_t = \boldsymbol{\Phi}_t\boldsymbol{\xi}_t + \boldsymbol{\zeta}, \tag{21}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^2$ represents observation noise and $\boldsymbol{\Phi}_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. Given the error signal $\mathbf{e}_t = \hat{\mathbf{p}}_t - \breve{\mathbf{p}}_t$, the state vector $\boldsymbol{\xi}_t^+$ is updated as

$$\boldsymbol{\xi}_t^+ = \boldsymbol{\xi}_t + \boldsymbol{\Gamma}_t\mathbf{e}_t. \tag{22}$$

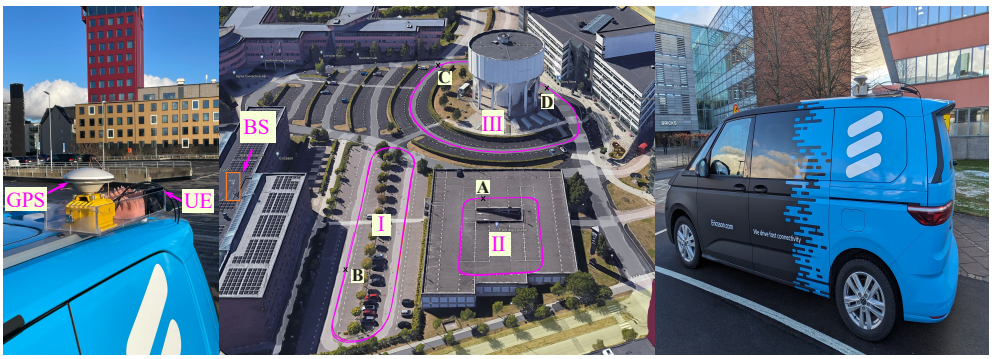
In (22), $\boldsymbol{\Gamma}_t$ represents the Kalman gain matrix, which balances the predictions from the state-space model and the ML-based pipeline, specifically,

$$\boldsymbol{\Gamma}_t = \boldsymbol{\Xi}_t\boldsymbol{\Phi}_t^T\left[\boldsymbol{\Phi}_t\boldsymbol{\Xi}_t\boldsymbol{\Phi}_t^T + \mathbf{R}\right]^{-1}, \tag{23}$$

where $\mathbf{R}$ is the covariance matrix of $\boldsymbol{\zeta}$. After computing $\boldsymbol{\Gamma}_t$, the covariance matrix $\boldsymbol{\Xi}_t$ is updated using

$$\boldsymbol{\Xi}_t^+ = (\mathbf{I} - \boldsymbol{\Gamma}_t\boldsymbol{\Phi}_t)\boldsymbol{\Xi}_t, \tag{24}$$

where $\mathbf{I}$ denotes the identity matrix. By applying the process outlined by (18)-(24), we can significantly mitigate the impact of prediction outliers, as will be further illustrated in Section V.



(a) (b) (c)

Fig. 3: The 5G NR BS was equipped with an antenna integrated radio with 64 transmitters and receivers, placed on top of a 20 m high building. In this measurement campaign, a vehicle moves along three pre-defined routes: **I** A route on a 10 meter-high garage for LOS measurements. **II**: A ground-level route for NLoS measurements below the building of the BS. **III**: A ground-level route for combined LoS and NLoS measurements. (a) GPS and UE, (b) Measurement Scenario, (c) Measurement van.

IV. Outdoor 5G NR measurement campaign

To evaluate our localization pipeline, an outdoor vehicular measurement campaign was conducted at a parking lot outside of the Ericsson office in Lund, Sweden. Photos of the test vehicle, the BS antenna, the UE as well as the measurement areas are presented in Fig. 3.

A. Introduction to the measurement campaign

During the measurement campaign, the test vehicle carried a GNSS receiver, and a commercial UE, see Fig. 3(a). Centimeter-level ground truth positioning accuracy was achieved using a Swift Duro high-performance GNSS receiver with real time kinematics technology, GNSS multi-band and multi-constellation support. To ensure that the UE remained in connected state, it simultaneously downloaded data at a 750 Mbit/s rate enabling continuous SRS UL transmission. The UL SRS pilot signals were received and processed by a commercial Ericsson 5G BS operating in mid-band at 3.85 GHz center frequency. The BS was compliant to the 5G NR 3rd Generation Partnership Project (3GPP) standard 38.104 Rel15 [36] and equipped with a time division duplexing (TDD) antenna integrated radio with 64 transmitters/receivers (TX/RX) consisting of 32 dual-polarized antennas covering a 120 degree sector. As for digital beam forming, 64 TX/RX formulate 64 beams in downlink (DL) and UL respectively. As illustrated in Fig. 4, the SRS channel estimates are reported for 273 physical resource blocks (PRBs) over a 100 MHz bandwidth.

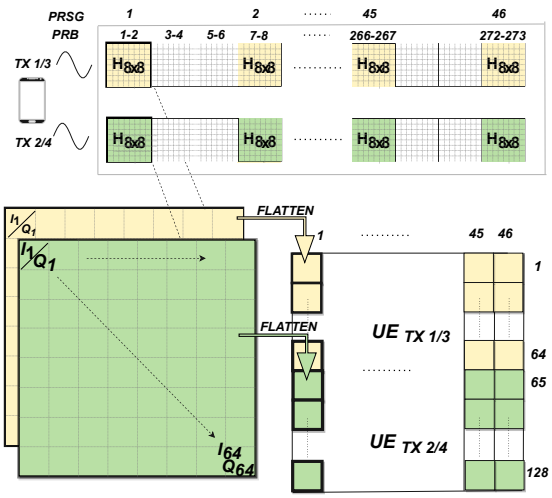


Fig. 4: SRS data collection and CTF generation. I/Q means in-phase/quadrature.

Each channel snapshot contains the 273 PRBs for all 64 beams. The PRBs are grouped and averaged in pairs, resulting in 137 Physical Resource Blocks Sub Groups (PRSG). Down sampling was done so that every third PRSG was further used generating 46 PRSGs in total. The UE was equipped with 4 antenna ports, i.e. 4 UE layers, sounding SRS pilots. Due to the capacity of our data-streaming system, the BS recorded the channel responses of 2 UE antenna ports which formulate two channel transfer function matrices $\mathbf{H}_1, \mathbf{H}_2 \in \mathbb{C}^{N \times F}$. We define a matrix $\mathbf{H}' \in \mathbb{C}^{2N \times F}$ to collect those two matrices, specifically, $\mathbf{H}' = [\mathbf{H}_1, \mathbf{H}_2]$ ($N = 64, F = 46$).

As illustrated in Fig. 3, our measurement campaign comprises three distinct scenarios: LoS, NLoS, and a mixed scenario. In all scenarios, the velocity of the vehicle is approximately 15 km/h. The trajectory for each of the three measurement scenarios consists of 5 laps. In the LoS scenario, the test vehicle drove at an open parking lot, while in the NLoS scenario, the vehicle was driving next to a tall building that obstructed the LoS path. As for the mixed scenario, NLoS conditions occurred when the LoS was blocked by the water tower. For all three measurements, the BS station recorded channel snapshots with 20 ms periodicity, resulting in $\mathcal{T}_1 = 22000$, $\mathcal{T}_2 = 24603$ and $\mathcal{T}_3 = 27087$ channel snapshots. We formulate three tensors $\mathcal{A}_{\text{LoS}} \in \mathbb{C}^{\mathcal{T}_1 \times 2N \times F}$, $\mathcal{A}_{\text{NLoS}} \in \mathbb{C}^{\mathcal{T}_2 \times 2N \times F}$, $\mathcal{A}_{\text{mix}} \in \mathbb{C}^{\mathcal{T}_3 \times 2N \times F}$ to collect all snapshots. Those three tensors are normalized by multiplying each with a scalar so that their Euclidean norms equals $\mathcal{T}_i MN$, where $i = 1, 2, 3$.

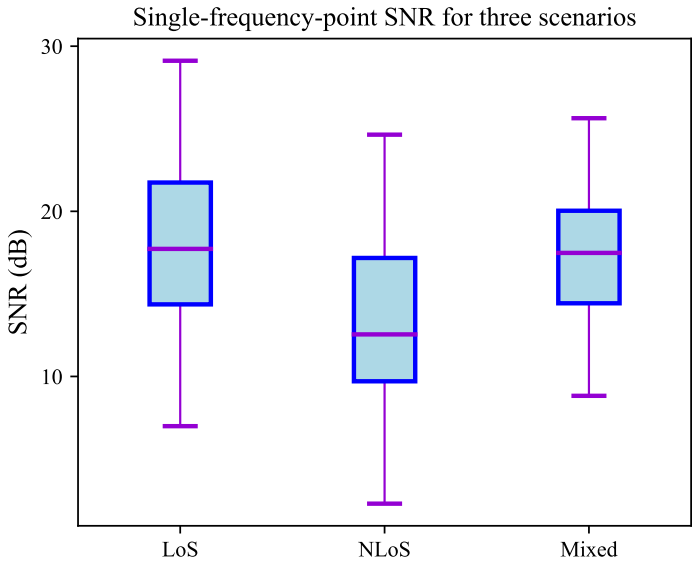


Fig. 5: Single-frequency-point SNR for three scenarios.

B. Measured propagation channel characteristics

In Fig. 5, we illustrate the range of single-frequency point SNR across three typical scenarios. As shown in the figure, most SNR samples in the LoS (Line-of-Sight) scenario are concentrated between 14.2 and 21.6 dB, with a median value of 17.6 dB. Similarly, in the mixed scenario, most of the SNR values fall within the range of 14.3 dB to 19.6 dB, with a median of 17.2 dB. In contrast, the NLoS (Non-Line-of-Sight) scenario exhibits significantly lower SNR values, ranging from 9.4 dB to 16.9 dB, with a median of 12.3 dB. It is important to note that our processing pipeline can achieve approximately 15 dB gain through antenna beamforming. To futher display the measured channel property, we choose four UE positions (positions A-D, see Fig. 3 (b)) from the three measurement scenarios and show representative channel impulse responses (CIR) in Fig. 6 (a)-(d). To be specific, Fig. 6 (a) illustrates a typical LoS scenario where a dominant LoS path can be seen from both the CIR and the beam patterns. Few beams exhibit dominant power levels, while others remain comparatively weaker. Although few NLoS-paths can still be observed, their strengths are much weaker compared to the direct path. This is because the UE is located in an open parking lot, where the reflected signals from other buildings are relatively weak. From the beam power pattern, one can observe the signal strength variations of different BS antenna polarizations and UE transmission layers as well. In contrast, Fig. 6 (b) displays NLoS

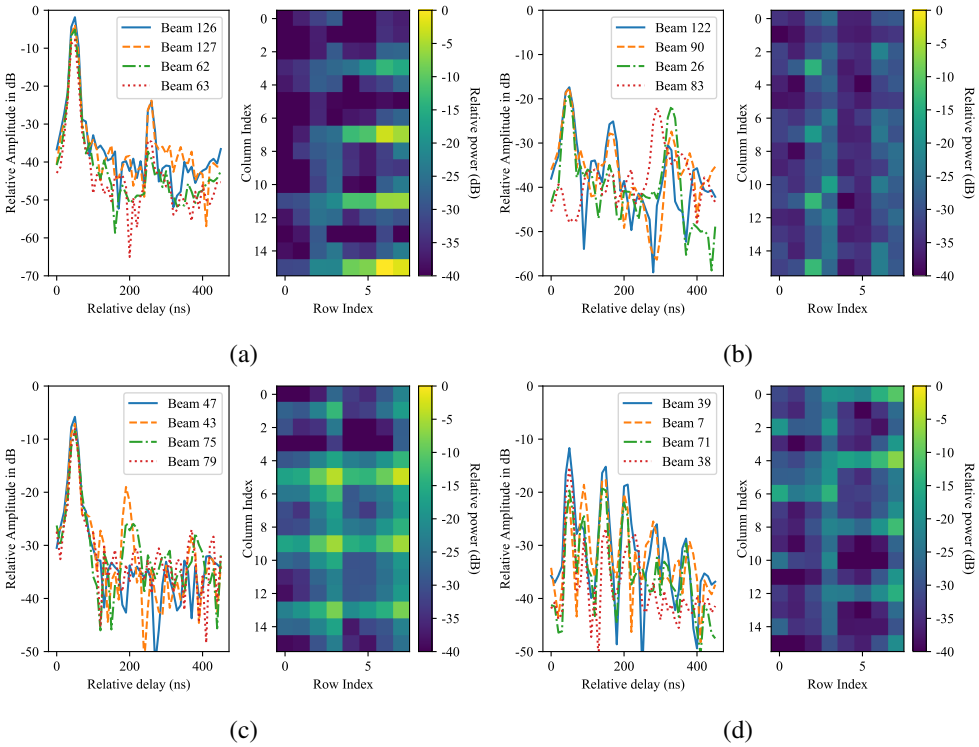


Fig. 6: CIR and relative power of all 128 beams of four locations (a) LoS at point A, (b) NLoS at point B, (c) LoS at point C, (d) NLoS at point D. Beam diagrams are arranged as follows: row $0-3$ and row $4-7$ represent the 32 vertical and 32 horizontal-polarized beams respectively for UE layer 1; row $8-11$ and row $12-15$ represent the co-polarized beams for UE layer 2. Beam index is $8*(i-1)+j$, where i and j denote the row and column index respectively. We select the first 4 strongest beam and plot the relative amplitude of CIR. The strongest beam among all figures (a)-(d) is normalized to 0-dB. The relative amplitude refers to the power difference of a specific beam to the strongest beam among all 4 figures.

channel characteristics where the BS captures several reflected paths and there is no path with a dominant power. Thus, the signal strength in Fig. 6 (b) is lower compared to the case in Fig. 6 (a). Fig. 6 (c) and Fig. 6 (d) present the measured channels in a mixed scenario, where more local scatters surround the UE. The distance between UE position C and the BS is greater than that of UE position A, resulting in a decrease in the strength of the received LoS signal. Nevertheless, the BS is capable of detecting stronger reflective paths in addition to the LoS path, attributed to reflections from surrounding buildings. Similarly, in Fig. 6 (d), a rich number of multipath components can be observed in both the CIR and the beam pattern, despite the LoS path being obstructed.

TABLE I: Overview of our ML-based single snapshot localization pipeline

Item	Network Structures or Parameters
Input Features	Amplitude of CIRs for all beams
Network Output	Estimated position labels or probabilities
Intermediate block 1	Residual 2-Heads Self-attention Network
Intermediate block 2	Residual Position-wise FCNNs
Intermediate block 3	3 cascaded ordinary FCNNs
Time Complexity	NF^2

We focus on simple deployments and mobility scenarios to showcase the novel approach, specifically targeting typical urban and rural environments, including LoS, NLoS, and mixed scenarios common in commercial networks. The dense, controlled test scenarios provide a robust evaluation of the proposed positioning algorithm. In contrast, larger network deployments would increase complexity and pose significant challenges in data generation, collection, and processing, which fall beyond the scope of this study.

V. Results and discussion

In this section, we evaluate our ML-based localization pipeline using the measurements. We initially compare the single-snapshot localization performance for different ML algorithms under different scenarios. Then, we demonstrate the performance gain achieved by smoothing multiple position estimates with a Kalman filter.

A. Single snapshot localization

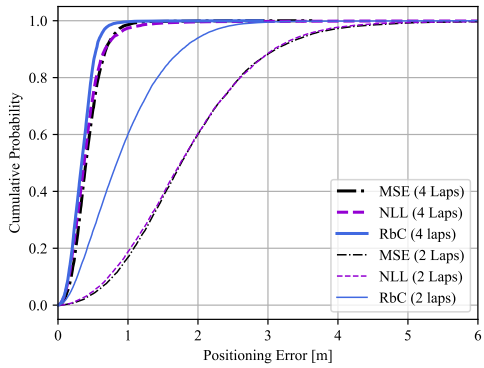
Our approach starts with assessing the validity of the input channel snapshot, as outlined in Section. III. B. The first criterion, related to the CTF matrix $\mathbf{\Xi}$, employs a cut threshold set at 3500 out of 5888 (128×46) available physical resource elements, approximately 60%, so that the channel information is sufficient. With such threshold setting, signal paths can be clearly visualized from the channel impulse responses. After discarding snapshots with insufficient data, we generate the amplitude of impulse response beam matrix $|\mathbf{G}_t|$ and feed it to the attention-aided localization block. This block, with detailed parameters in Table. I, comprises three cascaded sub-blocks. Initially, positioning encoding is applied to $|\mathbf{G}_t|$ using (7). Subsequently, a layer normalization procedure follows according to (8). The normalized matrix is then input into a simple 2-head self-attention block with a single self-attention layer, generating matrix $\mathbf{Z}'$ via (2-6). The pairwise correlation values in matrix $\mathbf{A}$ reflect the similarities between each pair of row vectors in $\mathbf{H}_t$ in the beam domain, which provides valuable information for UE localization. Considering the simplicity of future hardware implementation work, the exact parameter settings are displayed as follows: $A_1 = 128, A_2 = 46, A_3 = 64$. After the Add & Norm operation, the output is transferred to the second sub-block, consisting of two FCNNs with sizes $\mathbf{W}_1 \in 46 \times 128$ and $\mathbf{W}_2 \in 128 \times 46$. Following this, the output matrix of the second sub-block is vectorized to yield a vector of

TABLE II: Structures and parameter settings of the third FCNN sub-block using three different loss functions. Lr: Learning rate.

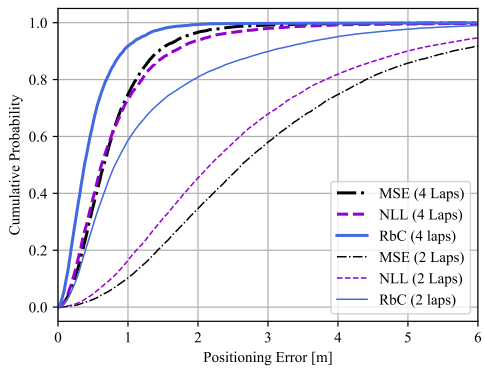
Items \ Loss F.	MSE	NLL	RbC
Input layer size	5888×1	5888×1	5888×1
Hidden layer 1	5888×32	5888×32	5888×128
Hidden layer 2	32×2	32×2	$128 \times \tilde{L}$
Batch size	64	64	64
Lr: LoS (4 laps)	0.0006	0.0006	0.0006
Lr: NLoS (4 laps)	0.0006	0.0006	0.0006
Lr: Mixed (4 laps)	0.0006	0.0006	0.0006
Lr: LoS (2 laps)	0.0002	0.0002	0.0002
Lr: NLoS (2 laps)	0.0001	0.0001	0.0001
Lr: Mixed (2 laps)	0.0002	0.0002	0.0002
Learning Epoch	500	500	500
Dropout Rate	0.05	0.05	0.05
Cost function	(11)	(12)	(13)
Network Size	1.137 MB	1.138 MB	7.44 MB

length 5888. This vector is fed into the last FCNN sub-block, with sizes as given in Table II. Network sizes are shown in Table II, and they vary based on the selected cost functions. As seen in Table II, the neural network requires more resources when RbC is used as the cost function. This is because RbC needs to calculate the probability of each bin in the final layer, rather than simply estimating the 2-D position of the UE. Despite this, the size of all three networks remains under 10 MB, classifying them as lightweight neural networks. We compare the localization performance when using three different loss functions and in three typical scenarios. As illustrated, the output matrix of the second intermediate block is first vectorized and fed to the input layer of the third sub-block, which consists of 2-3 FCNNs depending on the choice of loss functions. When the loss function RbC is used, its corresponding network delivers the probability of all L_x and L_y bins. In scenario I, $L_x = L_y = 200$ while in the other two scenarios $L_x = L_y = 100$. The deviation vectors $\mathbf{d}_x$ and $\mathbf{d}_y$ are set as: $\mathbf{d}_x = \delta_x \mathbf{1}, \mathbf{d}_y = \delta_y \mathbf{1}$, where $\mathbf{1}$ denotes the all-ones vector, δ_x and δ_y denote the deviation value of the x- and y-axis, respectively. Accordingly, the output dimension $\tilde{L}$ equals $L_x + L_y + 2$. The penalty term γ_1 and γ_2 are set as: $\gamma_1 = \gamma_2 = 1$. In addition, the Euclidean norm loss function is utilized, i.e. $\eta = 2$.

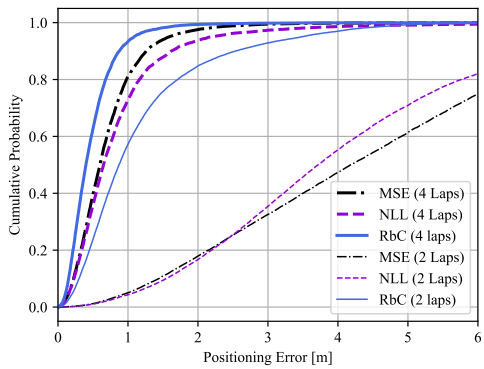
1) Comparisons of different uncertainty estimations: Fig. 7 compares the positioning accuracy of our single-snapshot localization pipeline using three loss functions in three scenarios under different training densities. As shown, the RbC method outperforms the other two methods in all three scenarios and under both high and low training densities. Compared to the other two methods, RbC learns better the non-Gaussian probability dis-



(a)

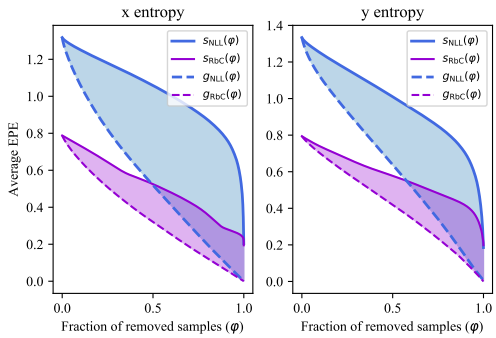


(b)

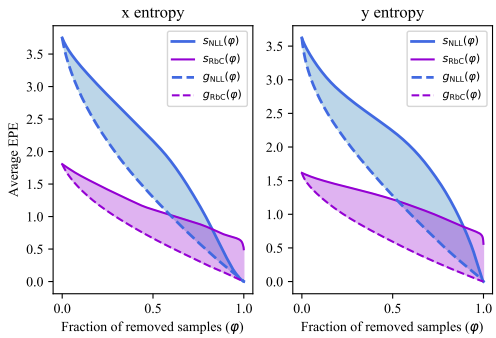


(c)

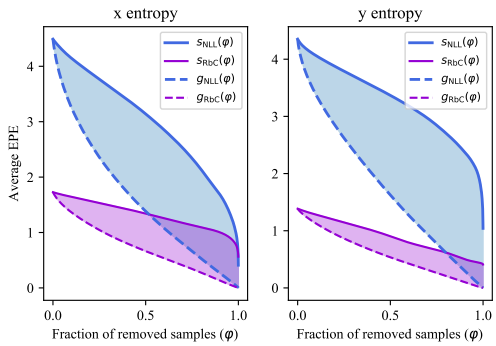
Fig. 7: Positioning errors of different training densities in the three scenarios: (a) LoS, (b) NLoS, (c) Mixed.



(a)



(b)



(c)

Fig. 8: Sparsification curves of NLL and RbC methods under high training density (4 laps as training data) and across three scenarios: (a) LoS, (b) NLoS, (c) Mixed. EPE: Endpoint error.

TABLE III: AUSE values of two uncertainty estimation algorithms under different training densities across three channel scenarios.

	NLL-x	RbC-x	NLL-y	RbC-y
LoS (4 laps)	0.480	0.179	0.351	0.163
NLoS (4 laps)	0.579	0.427	0.704	0.548
Mixed (4 laps)	1.428	0.616	1.543	0.325
LoS (2 laps)	1.951	0.968	2.023	1.181
NLoS (2 laps)	3.816	1.868	3.407	2.475
Mixed (2 laps)	4.682	0.809	3.540	1.138

tribution of the UE position, while the performance of the NLL method is constrained by its underlying Gaussian assumption, and the MSE method does not estimate uncertainty. The performance of these three methods differ less in the LoS scenario and high training density, because the estimated UE position has less uncertainty in this situation. However, in other scenarios or lower training density, the uncertainty of the estimated UE position increases due to reduced SNR or training samples. Consequently, an accurate uncertainty estimation is more essential, and thus the RbC method performs much better. At both high and low training densities, our pipeline performs best in LoS scenarios, the mixed scenario ranks 2nd, while the localization performance in the NLoS scenario is the worst. We postulate that in the LoS scenario, the much higher SNR contributes to very good positioning accuracy.

To further compare the uncertainty estimation quality of the NLL and RbC methods, we demonstrate the sparsification and oracle curves of the probability density functions of the estimated UE-x and y coordinates under high training density in Fig. 10. Specifically for the NLL method, we discretize the predicted Gaussian functions to achieve the same number of discrete bins as the RbC method, according to (16). The AUSE values for all training densities are calculated according to (17) and are displayed in Table III. To reduce the effect of outliers, the starting point of the sparsification and oracle curves equals 99% of the positioning error. As depicted in Fig. 10, the discrepancies between the sparsification (entropy) and oracle curves are significantly reduced in all three scenarios when the RbC method is used. This improvement is reflected in the improved AUSE values presented in Table III. These findings underscore the quality of the uncertainty estimation achieved with our approach.

We finally compare the NLL method to another popular approach, the Monte Carlo (MC) dropout method [37]. This technique estimates uncertainty by applying dropout to a trained neural network. During testing, the network is evaluated multiple times, with a percentage of neurons randomly deactivated on each run. This randomness results in slightly different predictions on each evaluation. The mean of these predictions provides

TABLE IV: Negative-log-likelihood values of MC Dropout methods compared with NLL.

	LoS (4 laps)	NLoS (4 laps)	Mixed (4 laps)
NLL	-0.144	0.069	0.066
MC dropout	2.142	0.014	0.138

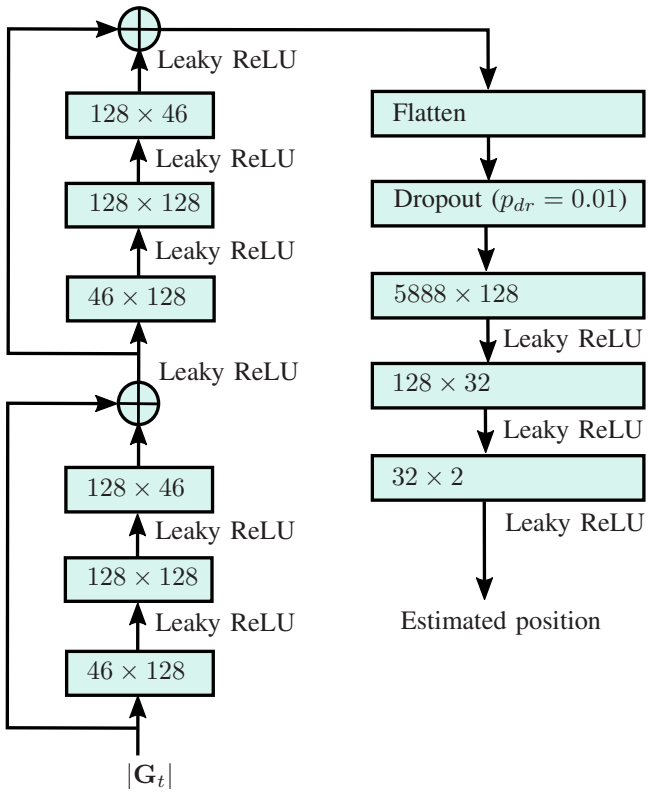


Fig. 9: Structure of the residue network.

the final estimate, while the variance among them represents the uncertainty. Using this approach, we evaluate our pipeline with MSE as the cost function. During testing, the dropout rate is set to 0.05, and for each input $\mathbf{G}_t$, the network is evaluated 50 times, after which we compute the mean predictions $\tilde{\mathbf{p}} = [\tilde{p}_{x_i}, \tilde{p}_{y_i}]^T$ and variances $\breve{\boldsymbol{\sigma}} i^2 = [\breve{\sigma}^2_{x_i}, \breve{\sigma}^2_{y_i}]^T$. Four laps are used for training, with the remainder allocated for testing. To assess performance, we calculate the negative log-likelihood (NLL) score of the MC dropout and NLL methods on the testing dataset, according to (12). The results are presented in Table IV. As shown, the MC dropout method performs similarly to the NLL method in both NLoS and Mixed

scenarios. However, under the LoS scenario, the MC dropout method shows overconfidence, with a significantly lower estimated variance compared to the NLL method. We attribute this to the fact that MC dropout primarily captures uncertainty related to the network's weights, but it does not fully account for other sources of uncertainty, such as model misspecification or uncertainty in the underlying data distribution.

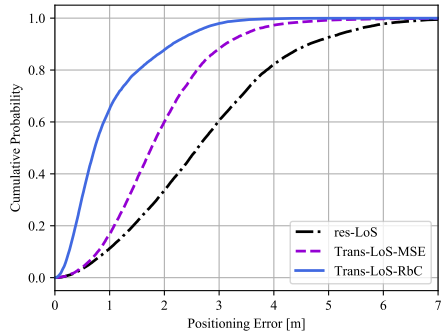
2) Compare with the start-of-the-art: We compare performances of our approach with a residue neural network (ResNet), which is widely used in solving regression and classification tasks. The structure and parameter settings of the neural network are illustrated in Fig. 9. As seen in Fig. 9, the residue network consists of two residue blocks, followed by three fully connected layers. The time complexity of this model is $O(N\max(F^2, FN, N^2))$. Dropout is also applied to avoid overfitting and Leaky ReLU (with negative slope -0.3) is selected as the activation function. The learning rate is set as follows: LoS 0.00001, NLoS 0.00005 and Mixed 0.00005. Fig. 9 illustrates the localization errors of attention-aided and ResNet based pipeline under all three scenarios. Channel data of two laps are selected as training and the rest three laps are used for testing purposes. Compared with the residue network, our transformer-based approach performs better under both LoS and NLoS scenarios, if MSE is selected as the loss function. Localization accuracy can be further significantly improved, if we use RbC approach to estimate uncertainty. We postulate that compared with the state-of-the-art, our processing pipeline benefits both from the attention mechanism and the advanced uncertainty estimation algorithms.

B. Smoothing the trajectory by Kalman filtering

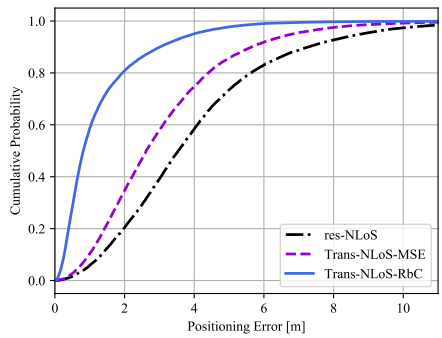
Next, we investigate the performance when using a Kalman filter for smoothing within our pipeline. To clearly visualize the effect of the Kalman filter, we apply a low training density, using two laps for training and one lap for testing. First, the validity of each channel CSI is assessed by the data cleaning block. All test channel samples classified as valid are then utilized for evaluation. Similarly to Section V.B, we apply an attention-aided block as the backbone and the output layer utilizes the RbC uncertainty estimation. For simplicity, the matrix $\mathbf{\Lambda}$ in (20) and the matrix $\mathbf{R}$ in (23) are set as

$$\mathbf{\Lambda} = \epsilon_1^2\mathbf{I}, \;\; \mathbf{R} = \epsilon_2^2\mathbf{I}, \tag{25}$$

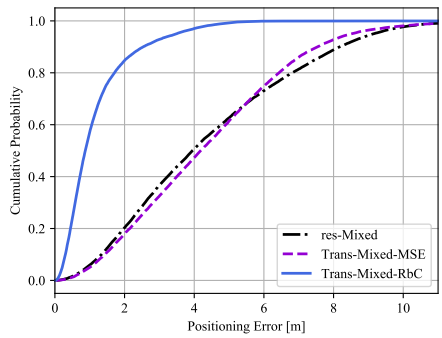
where ϵ_1 and ϵ_2 denote the standard deviation, which indicates the state and observation noise levels, respectively. Their exact values for the three scenarios are listed in Table V. Fig. 10 shows the predicted UE trajectories both with (right) and without (left) the Kalman filter for the three scenarios. The MSE between the predicted trajectories and their ground truths is shown in Table. V. As expected, the results demonstrate a significant improvement with the inclusion of the Kalman filter: the trajectories become considerably smoother, and outliers are mitigated to a large extent. Consequently, there is a substantial enhancement in localization accuracy, particularly evident in NLoS and mixed propagation scenarios. This



(a)

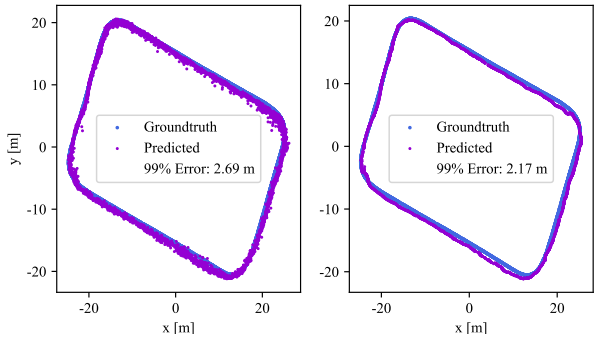


(b)

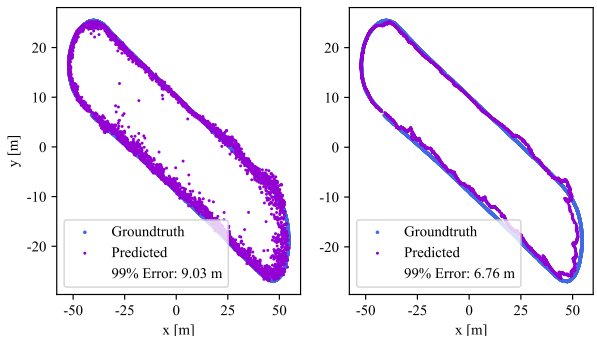


(c)

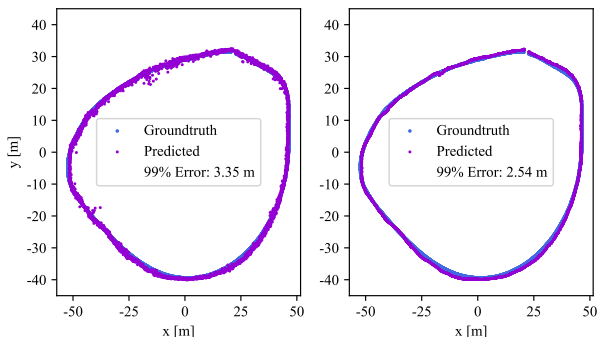
Fig. 10: Comparison between positioning error of our pipeline and a residue neural network under three different scenarios. (a) LoS, (b) NLoS, (c) Mixed. For all three scenarios, two laps are used for training and the rest for testing.



(a)



(b)



(c)

Fig. 11: Comparison between the raw (the left) and Kalman-filtered trajectory (a) LoS, (b) NLoS, (c) Mixed.

TABLE V: Parameter settings and rooted mean square errors (RMSE) when applying the Kalman Filtering

	ϵ_1	ϵ_2	RMSE (m), before filter	RMSE (m)
LoS	0.05	1.2	0.99	0.93
NLoS	0.05	1.2	2.00	1.76
Mixed	0.05	1.2	1.01	0.82

improvement can be attributed to the ability of the Kalman filter to utilize relationships between different snapshots, which effectively balances the newly predicted UE position with previous positional states, leading to more accurate localization.

VI. Conclusions

In this paper, machine learning is applied to a 5G NR cellular system for UE localization. A novel ML-based localization pipeline is presented, which utilizes attention-aided techniques to estimate UE positions by employing impulse response beam matrices as channel fingerprints. In addition, we implement two uncertainty estimation techniques, namely the NLL and RbC methods, to estimate the probability density function of the UE position error and compare their performances. Finally, a Kalman filter is applied to smooth consecutive position estimates. To evaluate our pipeline, an outdoor cellular 5G measurement campaign was conducted at 3.85 GHz with a 100 MHz bandwidth, covering both LoS and NLoS scenarios, achieving submeter-level localization accuracy. The measurement results indicate several key findings: 1) The attention-aided block shows promising potential to deliver high-precision localization accuracy. 2) The RbC uncertainty method outperforms the traditional NLL method, particularly with low training density or in more complex channel propagation scenarios. This advantage likely stems from the fact that the RbC method is not constrained by a Gaussian assumption on position errors. 3) Applying a Kalman filter to smooth consecutive position estimates significantly reduces position outliers, thereby enhancing localization accuracy. In future work, we plan to increase the diversity of our training data and expand the evaluation scenarios by testing our approach in various urban environments. Additionally, we will explore combining model-based and data-driven methods to further enhance the generalizability and robustness of our approach.

ACKNOWLEDGMENT

The authors thank PhD candidate Ziliang Xiong for valuable suggestions regarding the uncertainty prediction.

APPENDIX: LIST OF ABBREVATIONS

Abbreviation	Definition
2-D	Two-Dimensional
3-D	Three-Dimensional
3GPP	3rd Generation Partnership Project
5G	Fifth Generation
AoA	Angle of Arrival
AUSE	Area Under the Sparsification Error
BS	Base Station
CIR	Channel Impulse Response
CSI	Channel State Information
CTF	Channel Transfer Function
DL	Downlink
EPE	Endpoint Error
FCNN	Fully Connected Neural Network
GNSS	Global Navigation Satellite Systems
KNN	K-Nearest Neighbors
Lr	Learning Rate
LoS	Line-of-Sight
MC	Monte Carlo
ML	Machine Learning
MSE	Mean Square Error
NLL	Negative Log Likelihood
NLoS	None Line-of-Sight
NR	New Radio
OFDM	Orthogonal Frequency Division Multiplexing
PRB	Physical Resource Block
PRSG	Physical Resource Blocks Sub Groups
RbC	Regression-by-Classification
Rel	Release
RMSE	Rooted Mean Square Error
RX	Receivers
SNR	Signal-to-noise Ratio
SRS	Sounding Reference Signal
ToA	Time of Arrival
TDD	Time-division Duplexing
TDoA	Time Difference of Arrival
Tx	Transmitters
UE	User Equipment
UL	Uplink

References

[1] X. Cai, X. Cheng, and F. Tufvesson, "Toward 6G with Terahertz Communications: Understanding the Propagation Channels," *IEEE Communications Magazine.*, vol. 62, no. 2, pp. 32–38, Feb. 2024, doi: 10.1109/MCOM.001.2200386.

[2] R. Whiton, "Cellular Localization for Autonomous Driving: A Function Pull Approach to Safety-Critical Wireless Localization," *IEEE Veh. Technol. Mag.*, vol. 17, no. 4, pp. 28–37, Dec. 2022, doi: 10.1109/mvt.2022.3208392.

[3] A. A. Abdallah, C. S. Jao, Z. M. Kassas, and A. M. Shkel, "A Pedestrian Indoor Navigation System Using Deep-Learning-Aided Cellular Signals and ZUPT-Aided Foot-Mounted IMUs," *IEEE Sensors Journal.*, vol. 22, no. 6, pp. 5188–5198, Mar. 2022, doi: 10.1109/jsen.2021.3118695.

[4] M. Maaref and Z. M. Kassas, "Ground Vehicle Navigation in GNSS-Challenged Environments Using Signals of Opportunity and a Closed-Loop Map-Matching Approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2723–2738, Jul. 2020, doi: 10.1109/tits.2019.2907851.

[5] X. Mu, Y. Liu, L. Guo, J. Lin, and R. Schober, "Intelligent reflecting surface enhanced indoor robot path planning: A Radio Map-based approach,"*IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4732–4747, Jul. 2021, doi:10.1109/twc.2021.3062089.

[6] R. Whiton, J. Chen, T. Johansson and F. Tufvesson, "Urban Navigation with LTE using a Large Antenna Array and Machine Learning," 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 2022, pp. 1-5.

[7] J. A.del P Rosado, R. Raulefs, J. A. López-Salcedo, and G. Seco-Granados, "Survey of Cellular Mobile Radio Localization Methods: From 1G to 5G," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 2, pp. 1124–1148, 2018, doi: 10.1109/COMST.2017.2785181.

[8] A. Grenier, E. S. Lohan, A. Ometov, and J. Nurmi, "A Survey on Low-Power GNSS," *IEEE Commun. Surv. Tutor.*, vol. 25, no. 3, pp. 1482–1509, Jan. 2023, doi: 10.1109/comst.2023.3265841.

[9] A. Hu, T. Lv, H. Gao, Z. Zhang, and S. Yang, "An ESPRIT-Based Approach for 2-D Localization of Incoherently Distributed Sources in Massive MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 996–1011, Oct. 2014, doi: 10.1109/JSTSP.2014.2313409.

[10] X. Zeng, F. Zhang, B. Wang, and K. J. R. Liu, "Massive MIMO for High-Accuracy Target Localization and Tracking," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10131–10145, Jun. 2021, doi: 10.1109/jiot.2021.3050720.

[11] X. Li, E. Leitinger, M. Oskarsson, K. Åström, and F. Tufvesson, "Massive MIMO-Based Localization and Mapping Exploiting Phase Information of Multipath Components," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4254–4267, Jun. 2019, doi: 10.1109/twc.2019.2922264.

[12] L. Lian, A. Liu, and V. K. N. Lau, "User Location Tracking in Massive MIMO Systems via Dynamic Variational Bayesian Inference," *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5628–5642, Nov. 2019, doi: 10.1109/tsp.2019.2943226.

[13] N. Garcia, H. Wymeersch, E. G. Larsson, A. M. Haimovich, and M. Coulon, "Direct Localization for Massive MIMO," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2475–2487, May 2017, doi: 10.1109/tsp.2017.2666779.

[14] X. Cai, W. Fan, X. Yin, and G. F. Pedersen, "Trajectory-Aided Maximum-Likelihood Algorithm for Channel Parameter Estimation in Ultrawideband Large-Scale Arrays," *IEEE Transactions on Antennas and Propagation*, vol. 68, no. 10, pp. 7131–7143, Oct. 2020, doi: 10.1109/tap.2020.2996774.

[15] X. Cai and W. Fan, "A Complexity-Efficient High Resolution Propagation Parameter Estimation Algorithm for Ultra-Wideband Large-Scale Uniform Circular Array," *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5862–5874, Aug. 2019, doi: 10.1109/tcomm.2019.2916700.

[16] R. Surya, E. Hossain, and V. K. Bhargava, "Machine Learning Methods for RSS-Based User Positioning in Distributed Massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8402–8417, Dec. 2018, doi: 10.1109/twc.2018.2876832.

[17] X. Sun, C. Wu, X. Gao, and G. Y. Li, "Fingerprint-Based Localization for Massive MIMO-OFDM System With Deep Convolutional Neural Networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10846–10857, Nov. 2019, doi: 10.1109/tvt.2019.2939209.

[18] X. Guo and N. Ansari, "Localization by Fusing a Group of Fingerprints via Multiple Antennas in Indoor Environment," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 9904–9915, Nov. 2017, doi: 10.1109/tvt.2017.2731874.

[19] G. Tian, I. Yaman, M. Sandra, X. Cai, L. Liu, and Fredrik Tufvesson, "Deep-learning based high-precision localization with massive MIMO," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 19–33, Jan. 2024, doi: 10.1109/tmlcn.2023.3334712.

[20] J. Vieira, E. Leitinger, M. Sarajlic, X. Li and F. Tufvesson, "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, Canada, 2017.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)., Red Hook, NY, USA.

[22] E. Gonultas, E. Lei, J. Langerman, H. Huang, and C. Studer, "CSI-Based Multi-Antenna and Multi-Point Indoor Positioning Using Probability Fusion," *IEEE*

Trans. Wireless Commun., vol. 21, no. 4, pp. 2162–2176, Apr. 2022, doi: 10.1109/twc.2021.3109789.

[23] Sibren De Bast, E. Vinogradov, and S. Pollin, "Expert-Knowledge-Based Data-Driven Approach for Distributed Localization in Cell-Free Massive MIMO Networks," *IEEE Access*, vol. 10, pp. 56427–56439, Jan. 2022, doi: 10.1109/access.2022.3177837.

[24] R. Bharadwaj, A. Alomainy, and S. K. Koul, "Experimental Investigation of Body-Centric Indoor Localization Using Compact Wearable Antennas and Machine Learning Algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 2, pp. 1344–1354, Feb. 2022, doi: 10.1109/tap.2021.3111308.

[25] R. Surya, E. Hossain, and V. K. Bhargava, "Low-Dimensionality of Noise-Free RSS and Its Application in Distributed Massive MIMO," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 486–489, Aug. 2018, doi: 10.1109/lwc.2017.2787764.

[26] A. Salihu, S. Schwarz and M. Rupp, "Attention Aided CSI Wireless Localization," 2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC), Oulu, Finland, 2022.

[27] M. Stahlke, T. Feigl, S. Kram, B. M. Eskofier and C. Mutschler, "Uncertainty-based Fingerprinting Model Selection for Radio Localization," 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nuremberg, Germany, 2023.

[28] A. Ráth, D. Pjanić, B. Bernhardsson and F. Tufvesson, "ML-Enabled Outdoor User Positioning in 5G NR Systems via Uplink SRS Channel Estimates," IEEE International Conference on Communications (ICC), Rome, Italy, 2023.

[29] R. Whiton, J. Chen, and F. Tufvesson, "Wiometrics: Comparative Performance of Artificial Neural Networks for Wireless Navigation," *IEEE Trans. Veh. Technol.*, pp. 1–16, Jan. 2024, doi: 10.1109/tvt.2024.3396286.

[30] S. F. Bhat, I. Alhashim and P. Wonka, "AdaBins: Depth Estimation Using Adaptive Bins," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021.

[31] Z. Xiong, A. Jonnarth, A. Eldesokey, J. Johnander, B. Wandt, and P.-E. Forssen, "Hinge-Wasserstein: Estimating multimodal aleatoric uncertainty in regression tasks," in 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop, to appear, 2024.

[32] S. K. Lind, Z. Xiong, P. Forssén, V. Krüger, Uncertainty quantification metrics for deep regression, *Pattern Recognition Letters*, 2024, doi:10.1016/j.patrec.2024.09.011.

[33] E. Ilg, et al., "Uncertainty estimates and multi-hypotheses networks for optical flow," in 2018 European Conference on Computer Vision (ECCV), Munich, Germany, 2018.

[34] C. L. Canonne, G. Kamath, and T. Steinke, “The discrete Gaussian for differential privacy,” in Proceedings of the 34th International Conference on Neural Information Processing Systems (Neurips), Vancouver, Canada, 2020.

[35] S. M. Kay, Fundamentals of statistical signal processing: estimation theory. USA: Prentice-Hall, Inc., 1993.

[36] 3GPP, “Base station (BS) radio transmission and reception,” 3rd Generation Partnership Project, Technical Specification (TS) 38.104, 2023, version 15.19.0.

[37] Y. Gal and Z. Ghahramani. ”Dropout as a Bayesian approximation: representing model uncertainty in deep learning,” in Proceedings of the 33rd International Conference on Machine Learning (ICML), New York, USA, 2016.