UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Beam Tracking Using Deep Learning Applied to 6G MIMO

Ailton Pinto de Oliveira

DM: 31/2024

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2024

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Ailton Pinto de Oliveira

Beam Tracking Using Deep Learning Applied to 6G MIMO

DM: 31/2024

UFPA / ITEC / PPGEE

Campus Universitário do Guamá

Belém-Pará-Brasil

2024

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

“ BEAM TRACKING USING DEEP LEARNING APPLIED TO 6G MIMO”

AUTOR: **AILTON PINTO DE OLIVEIRA**

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES.

APROVADA EM: 16/12/2024

BANCA EXAMINADORA:

Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior
(Orientador – PPGEE/ITEC/UFPA)

Prof. Dr. Leonardo Lira Ramalho
(Avaliador Interno – PPGEE/ITEC/UFPA)

Prof. Dr. Diego de Azevedo Gomes
(Avaliador Externo – UNIFESSPA)

VISTO:

Prof. Dr. Diego Lisboa Cardoso

(Coordenador do PPGEE/ITEC/UFPA)

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

P659b Pinto de Oliveira, Ailton.
Beam Tracking Using Deep Learning Applied to 6G MIMO / Ailton Pinto de Oliveira. — 2024.
62 f. : il. color.

Orientador(a): Prof. Dr. Aldebaro Barreto da Rocha Klautau Junior
Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2024.

1. Beam Tracking. 2. mmWave. 3. deep learning. 4. 6G. 5. V2I. I. Título.

CDD 621.382

Beam Tracking Using Deep Learning Applied to 6G MIMO

Dissertação de mestrado submetida à avaliação da banca examinadora aprovada pelo colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará e julgada adequada para obtenção do Grau de Mestre em Engenharia Elétrica na área de Telecomunicações.

Aprovada em _____ / _____ / _____

____

Prof. Dr. Aldebaro Barreto da Rocha Klautau Junior

ORIENTADOR

____

Prof. Dr. Leonardo Lira Ramalho

MEMBRO DA BANCA EXAMINADORA

____

Prof. Dr. Diego de Azevedo Gomes

MEMBRO DA BANCA EXAMINADORA

____

Prof. Dr. Diego Lisboa Cardoso

COORDENADOR DA PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Acknowledgments

First, I would like to thank my advisor, Prof. Aldebaro Klautau, for the intellectual guidance during my during my journey at UFPA, and for the knowledge provided during the interactions over the past years, which I consider to be essential elements of my professional and personal evolution.

Secondly, I wish to thank Prof. Leonardo Ramalho, and Prof. Diego Gomes for serving as readers and examiners of this work.

I am grateful for all my colleagues from LASSE, who have provided knowledge and unforgettable moments over the last few years. I would like to extend special thanks to Lucas Damasceno, a companion who faced many challenges with me and with whom I forged a strong bond of brotherhood. I also thank Savio Bastos for his assistance with simulation tasks. A very special thanks goes to Daniel Takashi for his invaluable help; without his support, this work would not have been possible.

I am also grateful to my parents, Armando Oliveira e Maria de Belem for taking care of me. I am also profoundly thankful for my wife Joely Magno, for her love and support throughout these eight years and for believing in my potential when no one else did.

Most important, I am grateful for my son Rafael, the reason for all my efforts, the owner of all my love, and the one for whom I will always fight. I dedicate this work especially to you.

Last but not least, I want to thank me for believing in me, I want to thank me for doing all this hard work. I wanna thank me for having no days off. I wanna thank me for never quitting. I wanna thank me for always being a giver and trying to give more than I receive. I wanna thank me for trying to do more right than wrong.

Ailton Oliveira
December 2024

All we have to decide is what to do with the time that is given to us.
J. R. R. Tolkien

List of Acronyms

3GPP	Third Generation Partnership Project
5G	5th Generation
6G	6th Generation
AoA	Angle of Arrival
AoD	Angle of Departure
B5G	Beyond 5G
BS	Base Station
CNN	Convolutional Neural Network
CSI	Channel State Information
DL	Downlink
GNSS	Global Navigation Satellite System
HDF5	Hierarchical Data Format - Version 5
IBC	Image-Based Coding
I2I	Infrastructure-to-Infrastructure
LOS	Line of Sight
LMS	Least Mean Squares
LSTM	Long Short-Term Memory
MAFD	Mean Absolute First Difference
MIMO	Multiple-Input Multiple-Output
mmWave	Millimeter Wave
MPC	Multipath Components
MSE	Mean Squared Error
NLMS	Normalized Least Mean Squares
NLOS	Non Line of Sight
OSM	OpenStreetMap

PHY Physical

RAN-1 Radio Access Network - 1

RF Radio Frequency

RNN Recurrent Neural Network

RSRP Reference Signal Received Power

RT Ray-Tracing

Rx Receiver

SNR Signal-to-Noise Ratio

SUMO Simulator for Urban Mobility

THz Terahertz

TR Throughput Ratio

Tx Transmitter

UE User Equipment

ULA Uniform Linear Array

V2I Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle

WI Wireless Insite

WLAN Wireless Local Area Network

List of Figures

List of Tables

Contents

Abstract

This work explores the application of machine learning to enhance beam tracking in 6G MIMO Vehicle-to-Infrastructure (V2I) communications. Beam tracking, essential for sustaining reliable mmWave connections, remains challenging due to the high mobility of vehicular environments and the significant overhead associated with millimeter wave MIMO beamforming. While beam selection has been extensively studied, ML-based beam tracking is relatively underexplored, largely due to the scarcity of comprehensive datasets. To bridge this gap, this study introduces a novel public multimodal dataset, designed in accordance with 3GPP requirements, which combines wireless channel data with multimodal sensor information. This dataset supports the evaluation of advanced data fusion algorithms specifically tailored to V2I scenarios. Furthermore, a custom recurrent neural network (RNN) architecture is proposed as a robust solution for effective beam tracking, leveraging temporal and multimodal data to address the challenges of dynamic vehicular communications.

***Keywords* —** Beam Tracking, mmWave, deep learning, 6G, V2I.

Resumo

Este trabalho explora a aplicação de aprendizado de máquina para aprimorar o rastreamento de feixes em comunicações 6G MIMO Vehicle-to-Infrastructure (V2I). O rastreamento de feixes, essencial para sustentar conexões mmWave confiáveis, continua desafiador devido à alta mobilidade dos ambientes veiculares e à sobrecarga significativa associada à formação de feixes MIMO de ondas milimétricas. Embora a seleção de feixes tenha sido amplamente estudada, o rastreamento de feixes baseado em ML é relativamente pouco explorado, em grande parte devido à escassez de conjuntos de dados abrangentes. Para preencher essa lacuna, este estudo apresenta um novo conjunto de dados multimodais públicos, projetado de acordo com os requisitos do 3GPP, que combina dados de canal sem fio com informações de sensores multimodais. Este conjunto de dados oferece suporte à avaliação de algoritmos avançados de fusão de dados especificamente adaptados para cenários V2I. Além disso, uma arquitetura de rede neural recorrente (RNN) personalizada é proposta como uma solução robusta para rastreamento de feixes eficaz, aproveitando dados temporais e multimodais para abordar os desafios das comunicações veiculares dinâmicas.

***Palavras-chave* —** Beam Tracking, mmWave, deep learning, 6G, V2I.

Chapter 1

Introduction

The 5th Generation (5G) wireless networks is already a reality, the rapid advancement of communication technologies toward 6th Generation (6G) networks aim to meet increasingly demanding requirements, such as ultra-high data rates and low latency [1]. Achieving these goals necessitates the use of wider bandwidths, which are challenging to allocate in the already congested sub-6 GHz spectrum. To address this, Millimeter Wave (mmWave) frequencies, such as 28 GHz and 60 GHz, have been reserved as key bands for mobile communications [2]. The abundant spectrum at mmWave enables reduced symbol time, lower latency, and increased throughput, making it a critical enabler for advanced wireless systems. However, mmWave bands also exhibit higher path attenuation compared to sub-6 GHz frequencies, posing significant challenges for reliable communication.

To overcome the increased attenuation at mmWave frequencies, Multiple-Input Multiple-Output (MIMO) technology has become a cornerstone of 5G and 6G networks [3]. Beamforming, a critical component of massive MIMO, enhances the directionality of electromagnetic waves, mitigating the effects of high path loss [4]. Despite its benefits, beamforming introduces its own challenges, particularly the need for precise alignment between narrow beams at the transmitter and receiver. This requirement for accurate beam tracking underpins many of the ongoing research efforts in mmWave communications [5].

These requirements are especially critical in scenarios such as Vehicle-to-Infrastructure (V2I) communication, where precise beam alignment is essential to maintain robust links under dynamic conditions. In order to provide seamless high-quality services, beam management, which collectively encompasses initial beam training/alignment, monitoring and tracking, as well as recovery from beam failures, is crucial for 6G mmWave and Terahertz (THz) communications [6]. Recent Third Generation Partnership Project (3GPP) initiatives have focused on exploring advanced methods, including machine learning, to enhance beam management processes, aiming to address the challenges posed by dynamic wireless environments and the stringent requirements of 6G networks [7].

Beam tracking is the process of continuously aligning the transmission and reception beams to maximize signal quality, is a fundamental challenge in this context. This dissertation is motivated by the challenge of explore the potential of deep learning techniques to improve the efficiency and reliability of beam tracking in mmWave MIMO systems in dynamic V2I scenarios. The proposed solution aims to strike a balance between computational efficiency and prediction accuracy

1.1 State of the Art Review

Beam tracking in mmWave communication has emerged as a critical area of research, addressing the challenges posed by wireless communications in high mobility and dynamic environments. This section reviews notable advancements, highlighting key contributions and methodologies from recent studies.

Beam management encompasses beam alignment, selection, tracking, and failure recovery, all of which are essential for robust mmWave communication. Challenges such as high path loss, narrow beamwidth [6]. Traditional exhaustive beam search methods, while accurate, impose significant overhead in the communications networks, motivating the exploration of varied algorithms solutions [8, 9], including machine learning-based solutions [10, 11, 12, 13].

Prior works have investigated non-machine learning algorithms for beam tracking, focusing on leveraging mathematical optimization and filtering techniques to address the challenges of mmWave communication. Shaham et al. [9] proposed a Extended Kalman filter-based approach for beam tracking in V2I scenarios, using position, velocity, and channel coefficient as state variables. Their method aims to reduce computational cost, similar to the approach proposed in this work. However, unlike this work, Shaham et al.'s approach relies on the assumption of constant velocity, which simplifies the mobility model. In contrast, the datasets used in this research incorporate more realistic and dynamic mobility patterns.

Asi et al. [8] compare the performance of Least Mean Squares (LMS) algorithm and Normalized Least Mean Squares (NLMS) for tracking the channel and designing the beam. However, their analysis does not include a benchmark with other advanced tracking algorithms, which limits the scope of the evaluation. Their results show that the the LMS implementation required a larger step size to achieve faster system convergence and stability. In contrast, the NLMS algorithm achieved faster convergence with a more stable learning process. While the paper acknowledges this and presents results for various step sizes, it does not propose a systematic method for optimal step size selection, which remains a challenge in adaptive filtering.

Yi et al. [5] proposed a multi-resolution codebook approach aimed at minimizing the search space by leveraging spatial and temporal dependencies. Their method employs a recursive strategy that begins with an exhaustive beam search to identify the sector in which the User Equipment (UE) is located. Using this information, distance and location are estimated, allowing beams to be generated around the predicted direction. The recursion is repeated, effectively using multiple codebook levels to identify a more precise direction.

Machine learning has emerged as a powerful tool for addressing the challenges of beam tracking in mmWave communication, with several studies proposing innovative models that leverage the predictive capabilities of deep learning. Zhao et al. [14] proposed an Long Short-Term Memory (LSTM)-based model that utilizes sub-6 GHz Channel State Information (CSI) to control multiple mmWave Base Station (BS), and predict the wide beam measurements. The work employs a standard LSTM approach to co-located sites, and use heterogeneous networks for BS's separated in different places. while the work was able to achieve relevant results with the standard LSTM model, the heterogeneous networks scenario proved to be a challenge for the proposed method. However, to the LSTM had a good performance, was necessary to utilize 16 time steps of input sequence, which is 4 times more than our work propose. Similarly, Lim et al. [10] developed a LSTM-based model combined with a sequential Bayesian filter to model the temporal evolution of Angle of Arrival (AoA) and Angle of Departure (AoD). The mode use as input a sequence of the previous estimated channels acquired before the transmission period begins. However, the reliance on sequential Bayesian estimation can increases computational complexity depending on the context.

Zhong et al. [15] proposed an Image-Based Coding (IBC) method that incorporates information such as vehicle locations and sizes into a Convolutional Neural Network (CNN). By utilizing environmental images, their approach is suitable in highly dynamic vehicular communication scenarios. However, the method is a site-specific application, as the trained model depends on the specific spatial structure of the scenario where it was trained. This raises concerns about the model's generalization capacity when deployed in unseen environments, highlighting a potential limitation of the method.

Oliveira et al. [16] extended this concept by demonstrating the effectiveness of multimodal data fusion, combining inputs from LIDAR data, GNSS positioning, and historical beam selections. This fusion strategy significantly enhanced prediction accuracy and showcased the potential of integrating diverse data sources for robust beam tracking in V2I scenarios. However this approach demands a large volume of data, and also require further studies on the generalization of the model.

These studies highlight the versatility and effectiveness of machine learning and traditional algorithms for beam tracking, particularly in adapting to complex and dynamic environments. However, challenges remain in balancing computational efficiency with prediction accuracy, as well as ensuring generalization across diverse scenarios and datasets. These gaps highlight the potential for further research about the subject.

1.2 Research Contributions

This thesis concentrates on presents contributions to the field of beam tracking in 5G/6G MIMO V2I communications, addressing key challenges in dynamic vehicular environments. The main contributions are outlined as follows:

1. **Proposed Deep Learning Model for Beam Tracking:** The research introduces a custom Recurrent Neural Network (RNN) architecture, specifically utilizing LSTM layers, tailored for sequential beam tracking in dynamic vehicular environments. The model is designed to process historical beam data, leveraging temporal patterns to predict future beam behavior. By addressing challenges such as fluctuating channel conditions and limited historical data, the proposed model achieves a balance between accuracy and computational efficiency, making it suitable for real-time beam tracking in V2I scenarios.
2. **Development of a Public Multimodal Dataset:** A novel dataset is introduced, adhering to 3GPP requirements, which integrates wireless channel data with multimodal sensor

information such as LiDAR and GNSS positioning. This dataset is specifically designed for evaluating V2I beam tracking scenarios.

3. **Evaluation of Beam Tracking Strategies:** Extensive experiments are conducted across two V2I scenarios, including urban canyon and residential settings, to evaluate the proposed beam tracking model. The results are compared against a CNN baseline model,and benchmarks the proposed approach against state-of-the-art methods from the literature, highlighting its ability to achieve higher accuracy and reduced latency in beam predictions. These evaluations not only underscore the robustness of the proposed method but also provide critical insights into its adaptability to varying environmental dynamics and communication constraints.

4. **Insights into Communication Dynamics:** The study emphasizes the importance of communication dynamics, such as beam sweeping intervals, coherence time, and hierarchical beamforming, in optimizing beam tracking strategies. These insights provide a foundation for reducing measurement overhead without compromising link reliability.

These contributions collectively advance the state of the art in ML-based beam tracking for mmWave MIMO systems, offering valuable tools and methodologies for future research and development in V2I communications.

1.3 Dissertation Outline

The following summarizes the organization of this work and its main contributions

Chapter 2: Fundamentals of Beam-Tracking and MIMO Processing This chapter covers the theoretical foundations of beam tracking and MIMO processing, including antenna arrays, beamforming techniques (analog, digital, hybrid), and MIMO channel propagation with multi-path components and RSRP metric. It also introduces the functioning of the LSTM layers, that will be a core tool in the proposed beam tracking.

Chapter 3: V2I Datasets Generation and Their 3GPP Aspects This chapter focuses on the generation of V2I datasets, specifically through Raymobtime methodology, and their alignment with 3GPP standards. It also covers the simulator configuration, wireless channel setup, and post-processing of simulation results.

Chapter 4: System and Architecture Overview This chapter presents the proposed beam tracking approach, detailing the time series structure modeling and the RNN architecture.

Chapter 5: Simulation and Results The chapter evaluates the proposed beam tracking method through simulations. It analyzes wireless channel dynamics, beam index variations, and system performance under different conditions, including Line of Sight (LOS)/Non Line of Sight (NLOS), and discusses the impact of reducing the number of measurements on the system's accuracy and efficiency.

Chapter 6: Conclusion The final chapter summarizes the key findings of this dissertation and discusses the implications of the research. It also outlines possible directions for future work in beam tracking and MIMO processing, as well as related topics for further investigation. The chapter concludes with a list of the published articles that resulted from this research.

Chapter 2

Fundamentals of Beam-Tracking and MIMO Processing

MIMO in wireless communication systems allows leverage multipath scattering between Transmitter (Tx) and Receiver (Rx) to substantially increase the spectral efficiency [17] through the use of multiple antennas arrange in a antenna array at both the transmitting and receiving ends. The use of MIMO also can increase data rates through multiplexing or exploiting channel diversity. MIMO systems have already been standardized in both Wireless Local Area Network (WLAN), under IEEE 802.11ax [18], and in mobile communications through IEEE 802.16e/m [19] and 3GPP standards for LTE and LTE-Advanced. These implementations operate primarily in sub-6 GHz frequencies and typically support a limited number of antennas [20].

Beamforming is a technique to modify the radiation pattern of an antenna array, making it more directive, forming one or multiples beams that could be changes in direction [21]. To maximize the Signal-to-Noise Ratio (SNR), beamforming technology modifies the beam by controlling the power and phase of each element of the antenna array to control the direction of the wave-front.

Since the physical layer technology has already approached the Shannon capacity limits, further improvements in wireless communication systems as MIMO technology and beamforming have become critical enablers of high-capacity and reliable data transmission, particularly in the context of 5G and the upcoming 6G networks [22]. These technologies significantly enhance the spectral efficiency and coverage by exploiting spatial diversity and directionality.

Beam management is the process responsible for ensuring optimal communication between the Tx and Rx by dynamically adjusting the directionality of beams. This process can be divided into two key methods: beam selection and beam tracking [23]. Beam selection involves identifying the best beam pair between the transmitter and receiver to maximize signal quality. As the environment changes, especially in mobile scenarios, beam tracking consist of continuously monitoring and updating the beam pair, maintaining a robust connection despite user movement, obstacles, or other interference.

2.1 Antenna array

Antenna arrays consist of multiple antennas working together to improve performance in terms of gain, directivity, and beamforming capabilities. By carefully configuring the placement and excitation of these antennas, array configurations can be designed to optimize signal reception and transmission in various scenarios. The following are three most common types of antenna arrays [24], along with their characteristics:

- Uniform Linear Array (ULA): A ULA comprises multiple antenna elements arranged in a straight line, spaced uniformly apart. The performance of a ULA is characterized by its ability to steer beams in the azimuth plane by adjusting the phase differences between the signals fed to the individual elements. Figure 2.1a shows the schematic representation of a ULA.
- Uniform Planar Array (UPA): Uniform Planar Array (UPA): A UPA consists of antenna elements arranged in a two-dimensional grid pattern. This configuration allows for control over both azimuth and elevation angles, facilitating three-dimensional beamforming capabilities. By manipulating the excitation of the elements in both dimensions, a UPA can create highly directive beams and improve spatial coverage. Figure 2.1b illustrates the layout of a UPA.
- Uniform Circular Array (UCA): A UCA features antenna elements arranged in a circular configuration, enabling omnidirectional radiation patterns in the azimuth plane while allowing for control over elevation. The UCA can provide improved spatial resolution and diversity compared to linear arrays. This configuration is particularly useful in applications such as direction of arrival (DoA) estimation and mobile communications. See Figure 2.1c for a visual representation of a UCA.

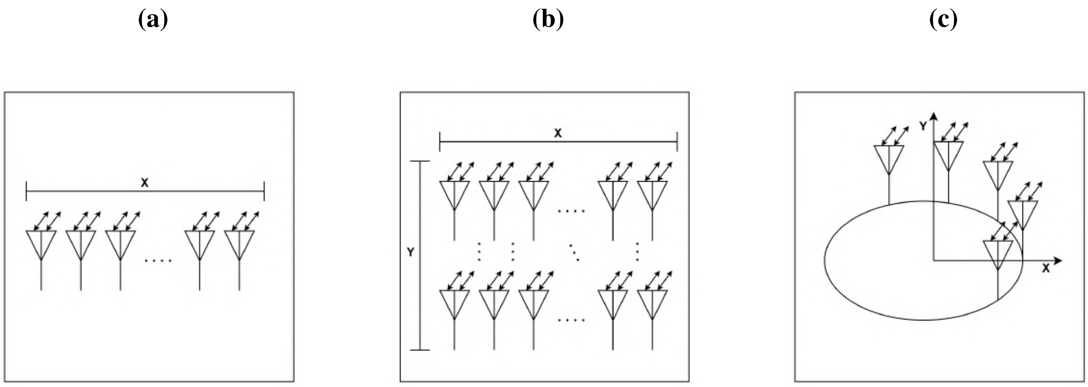

Figure 2.1: Schematics of uniform antenna arrays: (a) Uniform Linear Array (ULA), (b) Uniform Planar Array (UPA), and (c) Uniform Circular Array (UCA).

In the following section, all equations are derived under the assumption of a Uniform Linear Array (ULA), to simplify the analysis and improve conceptual clarity.

2.1.1 Phased Array

Phased array antennas are a type of antenna array that can electronically steer the direction of their radiation pattern without physically moving the antenna. This capability is achieved by adjusting the phase of the signals fed to each individual antenna element, allowing for dynamic control of the beam direction and shape. The ability to rapidly change the beam direction makes phased array antennas particularly useful in applications such as radar, telecommunications, and satellite communications [24].

The key to the operation of phased array antennas lies in the concept of phase shifts. By varying the phase of the signals transmitted or received by each antenna element, the resulting array factor can be manipulated to form a desired radiation pattern. By applying these phase shifts to the individual elements, the antenna can constructively and destructively interfere to form a main lobe in the desired direction while suppressing side-lobes.

Phased array antennas offer several advantages over traditional antennas, including:

- Electronic Beam Steering: The ability to change the direction of the beam electronically allows for rapid adjustments to the antenna's coverage area without mechanical movement.
- Multi-Functionality: Phased arrays can support multiple simultaneous beams, enabling

multiple users or services to be accommodated at the same time.

- Adaptive Beamforming: These antennas can dynamically adjust their patterns in response to changing conditions, such as interference or user movement.
- Compact Design: Phased array antennas can be designed to be smaller and more compact compared to conventional antennas with similar performance characteristics.

Overall, phased array antennas represent a significant advancement in antenna technology, enabling more flexible, efficient, and high-performance communication systems.

2.2 MIMO and Beamforming

The beamforming is performed through the interaction of the signals radiated by each antenna element of the antenna array to modify, through constructive and destructive interference, the radiation pattern. Changing the gain and phase of the signals transmitted in each element of the antenna array makes it possible to change the direction and shape of the array's radiation pattern. the beam widths can be manipulated through a pre-designed codebook, to direct the magnitude and phase of individual antenna signals in an array [25].

In phased antenna arrays, the performance of beamforming is determined by the collective behavior of individual antenna elements, which are arranged in a specific geometry. To understand how beamforming works, it is essential to define some core concepts:

2.2.1 Steering Vector

The *steering vector* is a mathematical representation of how signals from each antenna element are combined to steer the beam in a specific direction. By adjusting the steering vector, the system can steer the beam dynamically, ensuring the signal is directed toward the intended user or target area [26]. For an array of N antenna elements, the steering vector defines the phase shift applied to each element to ensure constructive interference of the transmitted or received signals in the target direction. Mathematically, the steering vector $\mathbf{v}(\phi)$ for a given angle ϕ (desired angle of arrival or departure) can be written as:

$$\mathbf{v}(\phi) = \left[e^{jk\cdot\mathbf{r}_1}, e^{jk\cdot\mathbf{r}_2}, \ldots, e^{jk\cdot\mathbf{r}_N}\right], \tag{2.1}$$

where $\mathbf{r}_n$ is the position of the n-th antenna element, and $\mathbf{k}$ is the phase-shift corresponding to the direction of propagation, that can be defined by:

$$k = 2\pi \cdot \frac{d}{\lambda} \cdot \cos(\phi), \tag{2.2}$$

wheres d is the distance between adjacent antennas elements, λ is the wavelength of the signal, - d is the distance between adjacent antenna elements, and $\frac{d}{\lambda}$ is the factor to scale the the phase shift between antennas by the normalized antenna spacing.

2.2.2 Array Factor

The *array factor* describes the overall radiation pattern of the antenna array, considering the combined effect of individual antenna elements. The steering vector, which applies phase shifts to each element, helps direct the beam in a specific direction, and the array factor reflects how effectively the array forms the beam in that desired direction [27]. In simple terms, the array factor defines how the signals from multiple antenna elements combine in different directions.

For an N-element uniform linear array ULA, the array factor $AF(\theta)$ is expressed as:

$$AF(\theta) = \sum_{n=1}^{N} e^{j(n-1)k}, \tag{2.3}$$

where θ is the direction of the signal (either incoming or outgoing) relative to the array's normal. It's used to evaluate how signals from different elements of the array interfere, either constructively or destructively, depending on the direction of observation. k is the same defined in equation 2.2, but with ϕ replaced by θ.

The array factor provides insights into the beam's directivity and gain, as well as how the beams are shaped based on the geometry of the antenna array.

2.2.3 Array Pattern

The *array pattern*, is the spatial distribution of the radiated power from the antenna array. It combines the array factor with the individual antenna element pattern, illustrating the directional properties of the antenna system [27]. The array pattern describes how energy is radiated in different directions, showing the main lobe (the primary direction of energy transmission) as well as side lobes and nulls (regions of reduced or zero radiation).

Mathematically, the total array pattern $P(\theta)$ is given by:

$$P(\theta) = AF(\theta) \cdot E(\theta), \tag{2.4}$$

where $E(\theta)$ is the radiation pattern of a single antenna element, describing how an individual element radiates energy in space.

2.2.4 Analog Beamforming

Analog beamforming utilizes phase shifters to control the signal phase of each antenna element, allowing the system to direct the beam toward a specific direction [28]. This approach typically uses the same signal to each antenna using a Radio Frequency (RF) chain, to making it more energy-efficient and cost-effective, especially in high-frequency applications like mmWave communications. However, the main limitation of analog beamforming is its inability to handle multiple beams simultaneously, restricting the system to single-stream transmission, which reduces its flexibility in complex multi-user environments.

2.2.5 Digital Beamforming

Digital beamforming, on the other hand, offers greater flexibility by using separate RF chains for each antenna element, enabling the system to form and process multiple beams concurrently [29]. This technique allows for precise control over both amplitude and phase at each antenna, supporting advanced functions like spatial multiplexing and beam steering. While digital beamforming can significantly enhance system capacity and performance, especially in multi-user scenarios, it requires higher computational power and energy consumption due to the multiple RF chains and signal processing demands.

2.2.6 Hybrid Beamforming

Hybrid beamforming combines the advantages of both analog and digital approaches to strike a balance between performance and cost. It uses a reduced number of RF chains while still leveraging phase shifters for analog beamforming, enabling partial digital control over the beams. This allows hybrid beamforming to support multiple beams and spatial multiplexing while keeping power consumption and hardware complexity lower than fully digital systems. Hybrid beamforming is particularly suited for millimeter-wave communications, where the trade-off between performance and efficiency is crucial.

2.3 MIMO propagation characteristics with MPC

MIMO systems rely on the transmission and reception of multiple signals across different paths between the transmitter and receiver, where the transmitted signal reflects, diffracts, or scatters off obstacles in the environment, resulting into Multipath Components (MPC) reaching the receiver [30]. This phenomenon is represented in ray-tracing models, which accurately simulate the paths a signal might take through the environment, considering various physical characteristics [31].

Multipath propagation enables spatial diversity, which is one of the key advantages of MIMO systems. Figure 2.2 illustrates a typical scenario where signals traverse multiple paths due to obstacles in the environment, leading to a combination of delayed, phase-shifted, and attenuated signals at the receiver.

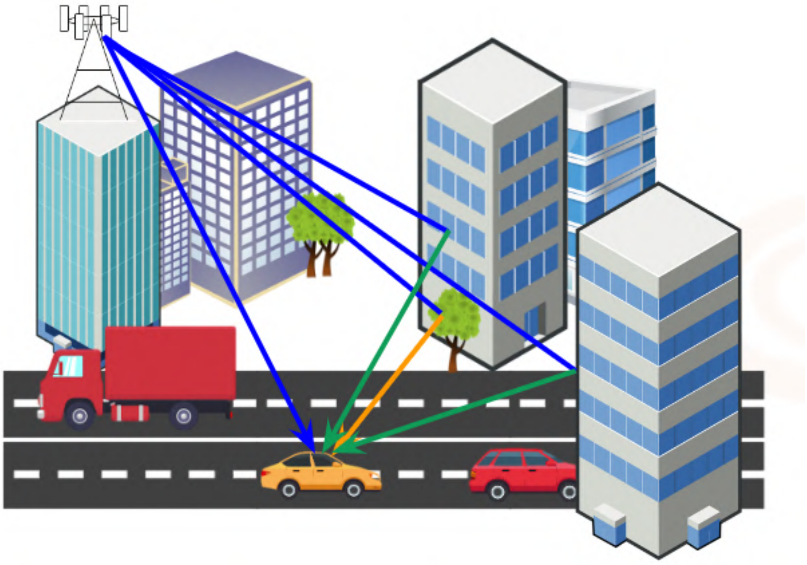

Figure 2.2: Multipath propagation in a MIMO system, where signals take multiple paths to reach the receiver.

Each MPC has its own set of parameters, including amplitude, phase, and delay, contributing to the composite signal that the receiver processes. This spatial diversity is leveraged by MIMO to improve overall system capacity and reliability. Some relevant parameters are:

- **Gain**: The relative amplitude of the received signal, influenced by factors like path loss, shadowing, and distance from the transmitter. Gain can vary significantly across different MPCs due to differences in propagation distance and interaction with obstacles.
- **Phase**: Each MPC arrives at the receiver with a unique phase shift, caused by the difference in travel distance. The constructive or destructive interference between these multipath signals can significantly affect the signal strength.
- **Time of Arrival (ToA)**: The delay with which each MPC arrives at the receiver is a key parameter in determining the temporal spread of the signal. Longer paths result in greater delays, and the spread of arrival times can cause inter-symbol interference (ISI) in high-data-rate systems.
- **Angle of Arrival (AoA)** and **Angle of Departure (AoD)**: These parameters describe the direction from which the signal arrives at the receiver and the direction in which it was transmitted, respectively. These angular characteristics enable MIMO systems to exploit spatial diversity, providing better coverage and increased data rates through beamforming and directional transmission techniques.

Together, these small-scale parameters form the basis for modeling the propagation characteristics of MIMO systems in realistic environments, enabling techniques such as beamforming and CSI acquisition to optimize system performance.

2.4 MIMO Channel Model

A MIMO channel with N_{tx} antenna elements at the transmitter and N_{rx} elements at the receiver can be described by its time-domain response, represented as a matrix $\mathbf{H}(\tau, t)$. This matrix encapsulates the impulse responses between each pair of transmit and receive antennas, accounting for the multipath components that arrive with different delays. The time-varying channel matrix is given by:

$$\mathbf{H}(\tau, t) = \begin{bmatrix} h_{1,1}(\tau,t) & h_{1,2}(\tau,t) & \dots & h_{1,N_{\mathrm{tx}}}(\tau,t) \\ h_{2,1}(\tau,t) & h_{2,2}(\tau,t) & \dots & h_{2,N_{\mathrm{tx}}}(\tau,t) \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_{\mathrm{rx}},1}(\tau,t) & h_{N_{\mathrm{rx}},2}(\tau,t) & \dots & h_{N_{\mathrm{rx}},N_{\mathrm{tx}}}(\tau,t) \end{bmatrix}, \tag{2.5}$$

where $h_{i,j}(\tau, t)$ is the time-varying channel impulse response between the i-th receive antenna and the j-th transmit antenna. This response represents the effect of the channel on an impulse transmitted at time $t - \tau$ and received at time t [32].

In a MIMO system, considering a transmitted signal $x_j(t)$ from the j-th element of the transmit array, the signal received at the i-th element of the receiver array (with N_{rx} elements) can be expressed as:

$$y_i(t) = \sum_{j=1}^{N_{\mathrm{tx}}} h_{i,j}(\tau, t) * x_j(t) + n_i(t), \quad i = 1, 2, \dots, N_{\mathrm{rx}}, \tag{2.6}$$

where $n_i(t)$ represents the Gaussian noise at the i-th receiver. This summation captures the combined effect of signals arriving from different transmit antennas, each modulated by its corresponding channel response.

2.4.1 Geometric Channel Model with MPC

For the analysis in this work, we consider the channel as narrowband, meaning the bandwidth is small enough that the channel's frequency response can be treated as constant over the band. To model the propagation characteristics, we employ a geometric channel model based on MPC [20]. In this model, the received signal is composed of L multipath components, each with a specific angle of arrival (AoA) and angle of departure (AoD), along with a complex gain α_ℓ.

The *narrowband* MIMO channel matrix can be described as:

$$\mathbf{H} = \sqrt{N_{tx} N_{rx}} \sum_{\ell=1}^{L} \alpha_\ell \mathbf{a}_r(\phi_\ell^A, \theta_\ell^A) \mathbf{a}_t^*(\phi_\ell^D, \theta_\ell^D), \tag{2.7}$$

where L is the number of MPCs, α_ℓ is the complex gain for the ℓ-th path, $\mathbf{a}_t(\phi_\ell^D, \theta_\ell^D)$ is the transmit array response vector, which depends on the azimuth ϕ_ℓ^D and elevation θ_ℓ^D angles of departure, and $\mathbf{a}_r(\phi_\ell^A, \theta_\ell^A)$ is the receive array response vector, which depends on the azimuth ϕ_ℓ^A and elevation θ_ℓ^A angles of arrival.

2.5 Beam Tracking

Beam tracking, an extension of beamforming, is an essential procedure in mmWave and massive MIMO systems in dynamic environments, such as vehicular networks, where the optimal signal path constantly changes due to the mobility of the transmitter, receiver, or surrounding objects [23], as illustrated in Figure 2.3. Effective beam-tracking techniques ensure that reliable communication links remain robust by continuously adjusting the beam direction in real time.

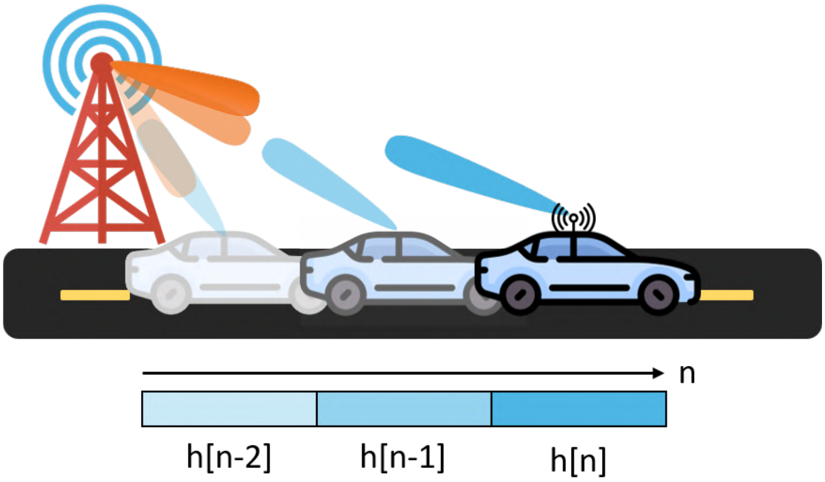

Figure 2.3: Schematic of Beam Tracking in a Vehicular Network.

Beam tracking algorithms typically aim to minimize the overhead the beam search process while ensuring the beam is aligned with the strongest propagation path. Beam tracking techniques in general involves continuously updating the selected beam using the historical data of the previous selected beams, to maintain the alignment as the user moves or as the environment changes. This requires periodic measurements and adjustments to the beam index $\hat{\imath}$ to adapt to the dynamic nature of the wireless channel. One common approach is to use hierarchical search strategies [33], which reduce the number of beam pairs that need to be evaluated, especially in large antenna arrays.

In this work we assume the beamforming used in the proposed beam tracking is defined by:

$$y_i = \mathbf{w}_r^* \mathbf{H} \mathbf{f}_t, \tag{2.8}$$

where $\mathbf{w}_r$ and $\mathbf{f}_t$ are respectively the precoding and combiner vectors used at the BS and UE to perform the beamforming. The vectors $\mathbf{w}_r$ and $\mathbf{f}_t$ are chosen from the codebooks $\mathcal{C}_t =$

$\{\mathbf{w}_1,\ ...,\ \mathbf{w}_{|\mathcal{C}_t|}\}$ and $\mathcal{C}_r = \{\mathbf{f}_1,\ ...,\ \mathbf{f}_{|\mathcal{C}_r|}\}$, where $|\mathcal{C}_t|$ and $|\mathcal{C}_r|$ are drawn from the Discrete Fourier Transform (DFT) 2.9, The size of the codebooks, $|\mathcal{C}_r|$ and $|\mathcal{C}_t|$, corresponds to the number of antenna elements at the receiver (r) and transmitter (t), respectively [34]:

$$\mathbf{F}(n) = \frac{1}{\sqrt{M}} e^{-\frac{j2\pi(n-1)}{M}}, \tag{2.9}$$

where M is the number of antenna elements in the ULA and $n = 1,\ ...,\ N$, with N denoting the number of codewords. In this work, we assume $N = M$.

Thus, the *optimal* beam index $\hat{i}$ is given by

$$\hat{i} = \arg \max_{i \in \{1,\cdots,M\}} |y_i|. \tag{2.10}$$

2.5.1 Evaluation Metrics

The use of RSRP as an essential metric for beam tracking has been proposed in 3GPP discussions regarding AI/ML for beam management for next-generation networks[35]. RSRP measures the received signal strength of the reference signal at the physical layer, providing an indication of the signal quality for a given beam. In beam tracking, RSRP is commonly employed to evaluate the performance of different beams and select the optimal one that maximizes the signal power received at the UE.

The RSRP for a given beam i can be expressed as:

$$\text{RSRP}_i = \frac{1}{N_{\text{RS}}} \sum_{n=1}^{N_{\text{RS}}} P_{\text{RS},i}^{(n)}, \tag{2.11}$$

where N_{RS} is the number of reference signal (RS) resources, and $P_{\text{RS},i}^{(n)}$ is the power of the n-th reference signal for beam i.

By periodically computing RSRP across multiple beam pairs, the system can identify the beam with the highest signal strength and dynamically switch to it, ensuring a stable and efficient communication link, even in scenarios involving user mobility or varying environments.

The Throughput Ratio (TR) is a metric used to evaluate the efficiency of beam-tracking algorithms in selecting optimal beams for data transmission. It quantifies the ratio between the achievable throughput when using the predicted beam and the maximum possible throughput obtained by selecting the best beam. Mathematically, it is defined as:

$$\text{TR} = \frac{\sum_{i=1}^{N} \log_2(1 + y_{\widehat{(i)}})}{\sum_{i=1}^{N} \log_2(1 + y_{\hat{i}})}, \tag{2.12}$$

where N is the number of test examples and $\hat{i}$ is the best beam index, and $\widehat{(i)}$ is the predicted beam index.

2.6 Fundamentals of LSTM Layers

LSTM layers are a type of RNN architecture designed to model sequential data effectively by overcoming the limitations of traditional RNNs. First introduced by Hochreiter and Schmidhuber in 1997 [36], LSTMs are particularly well-suited for tasks involving temporal dependencies, such as time series forecasting, natural language processing.

2.6.1 Structure of an LSTM Cell

An LSTM cell consists of three primary components known as gates: the forget gate, input gate, and output gate. These gates work together to regulate the flow of information through the cell and maintain a stable memory over long sequences:

- **Forget Gate:** Determines which information from the previous cell state should be discarded. It uses a sigmoid activation function to produce a value between 0 and 1, where 0 means "forget completely" and 1 means "retain completely."

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- **Input Gate:** Controls how much new information from the current input should be added to the cell state. It involves a sigmoid function for weighting and a tanh function to create candidate values.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

- **Output Gate:** Decides what part of the cell state should be output as the hidden state for the current timestep. This gate uses a sigmoid function to weight the state.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad h_t = o_t \odot \tanh(C_t)$$

The cell state C_t is updated as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

2.6.2 Advantages of LSTM Layers

The ability to selectively retain and update information gives LSTMs a significant advantage over traditional RNNs, particularly in handling vanishing gradient issues. Key benefits include:

- **Memory Retention:** LSTMs can maintain long-term dependencies by preserving relevant information in the cell state.
- **Effective Training:** By mitigating the vanishing gradient problem, LSTMs facilitate stable and efficient training over long sequences.
- **Versatility:** Their design makes LSTMs applicable to various domains, including speech recognition, machine translation, and dynamic system modeling.

Chapter 3

V2I Datasets Generation and Their Aspects from a 3GPP Perspective

This work employs the Raymobtime methodology to generate wireless channels data in the context of Beyond 5G (B5G). The Raymobtime methodology, initially proposed in [37] and later refined in [38], was developed to address the shortage of 5G mmWave datasets that incorporate time evolution, spatial consistency, and mobility features. These characteristics are achieved through the integration of two software tools. The first tool is Wireless Insite (WI), a Ray-Tracing (RT) simulator from REMCOM [39], which accurately models wireless communication channels by simulating propagation paths. The second tool is the Simulator for Urban Mobility (SUMO), which is free and open source software designed for traffic simulation and is responsible for generating dynamic movement patterns and simulating traffic scenarios.

3.1 Raymobtime Datasets

One of the primary contributions of the Raymobtime project is its commitment to supporting the academic community by not only providing a means to simulate mmWave ray-tracing in scenarios with mobility but also making publicly available a series of datasets with pre-calculated results. These datasets, which integrate channel information from ray-tracing, Global Navigation Satellite System (GNSS), and multimodal data sources, have been widely adopted by researchers to advance the study of communication systems in dynamic environments. For example, [40] utilized the datasets in their work to evaluate the use of federated learning techniques in beam selection applications, demonstrating the applicability of the datasets in modern research topics.

Multiple studies suggest that incorporating external data sources, such as LIDAR sensors and camera images [41, 11], can address critical challenges in 5G applications, including blockage prediction [42]. In this regard, Raymobtime provides a diverse set of multimodal data, integrating information from various sources, as illustrated in Figure 3.1.

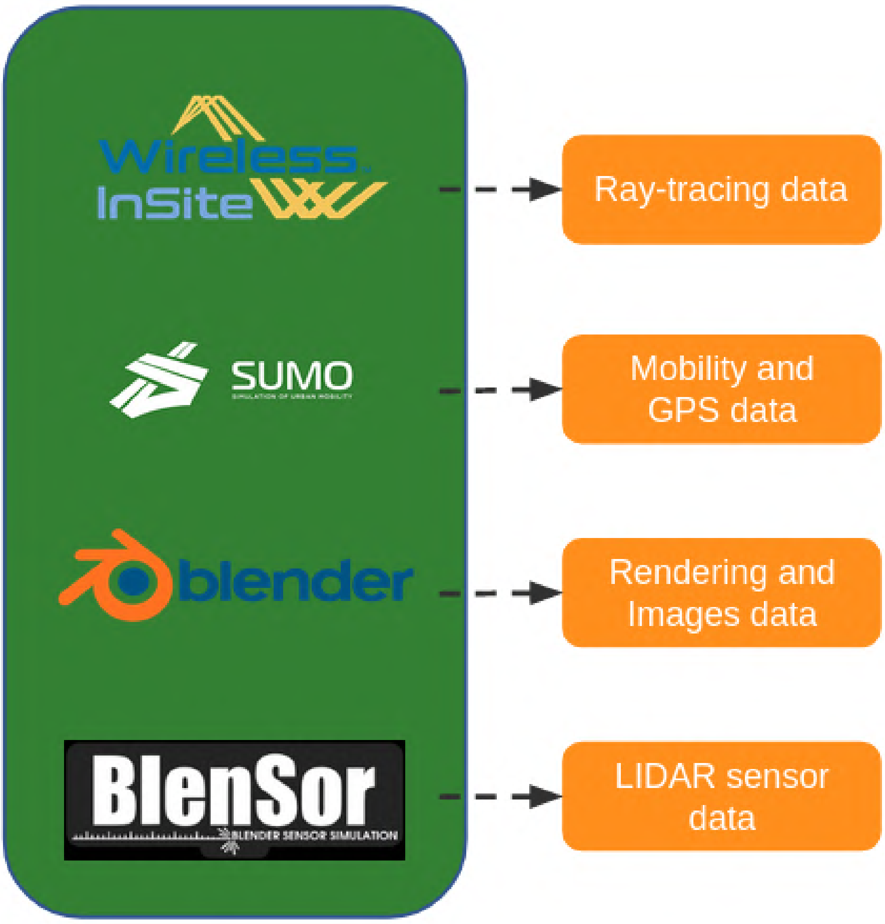

Figure 3.1: Raymobtime multimodal data sources.

The Raymobtime datasets are organized into different episodes, each consisting of multiple scenes. An episode represents a continuous recording of a specific scenario over a period, while a scene captures a momentary snapshot within the episode. This episodic structure allows for detailed analysis of how the wireless environment evolves over time, providing researchers with high-resolution temporal data. However, it is important to remember that episodes should be evaluated independently from each other. The datasets cover a range of scenarios and frequencies, featuring both mobile and fixed receivers, allowing for diverse use cases, such as Infrastructure-to-Infrastructure (I2I) communication for fixed receiver and Vehicle-to-Vehicle (V2V) communication and V2I communication for mobile receivers.

The current public available datasets can be categorized into three types: ray-tracing datasets combined with GNSS information (Table 3.1), multimodal datasets (Table 3.2) that also include LIDAR and camera imagery, and specialized V2V communication datasets (Table 3.3).

Table 3.1: Ray-Tracing + GNSS Datasets

ID	Scenario	Freq. GHz	Rx(Type)	Scene Interval	Episodes Interval	Episodes	Scenes per Episode	Valid Ch.
s000	Rosslyn	60	10 Mobile	100 ms	30 s	116	50	41K
s001	Rosslyn	2.8; 5	10 Fixed	5 ms	37 s	200	10	20K
s002	Rosslyn	2.8; 60	10 Fixed	1 s	3 s	1800	1	18K
s003	Rosslyn	2.8; 5	10 Fixed	1 ms	35 s	200	10	20K
s004	Rosslyn	60	10 Mobile	1 s	30 s	5000	1	35K
s005	Rosslyn	2.8; 5	10 Fixed	10 ms	35 s	125	80	100K
s006	Rosslyn	28; 60	10 Fixed	1 ms	35 s	200	10	20K
s010	Rosslyn	60	10 Mobile	0.5 s	5 s	100	50	30K
s011	Rosslyn	60	10 Mobile	0.5 s	6 s	76	20	13K
s012	Rosslyn	60	10 Fixed	0.5 s	6 s	105	20	21K

Table 3.2: Multimodal Datasets (Ray-Tracing + LIDAR + Camera Images + GNSS)

ID	Scenario	Freq. GHz	Rx(Type)	Scene Interval	Episodes Interval	Episodes	Scenes per Episode	Valid Ch.
s007	Beijing	2.8; 60	10 Mobile	1 s	5 s	50	40	15K
s008	Rosslyn	60	10 Mobile	0.1 s	30 s	2086	1	11K
s009	Rosslyn	60	10 Mobile	0.1 s	30 s	2000	1	10K

Table 3.3: Ray-Tracing Datasets for V2V Communication

ID	Scenario	Freq. GHz	Tx	Rx	Scene Interval	Episodes Interval	Episodes	Scenes per Episode	Valid Ch.
v001	Rosslyn	60	2	5	100 ms	30 s	20	50	8.5K
v002	Rosslyn	60	1	5	0.1 s	0.1 s	2500	1	12.5K

In order to contribute to the evolution of the methodology and provide higher-quality datasets, this work proposes to model new datasets in the V2I scenario, based on specifications obtained from 3GPP documents and discussions. The new datasets will be further detailed in section 3.3.

3.2 Raymobtime Simulator

In addition to offering public datasets, Raymobtime empowers researchers to create customized datasets tailored to their specific needs. The following section outlines a comprehensive guide, from installation to result generation.

3.2.1 Configuring Wireless Insite

WI allows for the import of 3D models to compose the simulation scenario and objects. Utilizing this feature, the methodology proposes exporting satellite data to generate a realistic environment based on a real location (see Figure 3.2). The first step is to choose a study area where the signal propagation is to be analyzed. To import the scenario, we suggest using the CadMapper tool, an online resource that transforms data from public sources such as OpenStreetMap, NASA, and USGS into organized CAD files.

To ensure realism, it is important to select well-mapped areas. The CAD file generated by CadMapper contains 3D constructions, roadways, topography, vegetation, and other elements. However, it is crucial to ensure that all these features are in a format compatible with WI. It is recommended use DXF or Collada file formats, although WI supports other formats as well.

Once the scenario is imported correctly, the next step is to configure the basic settings in

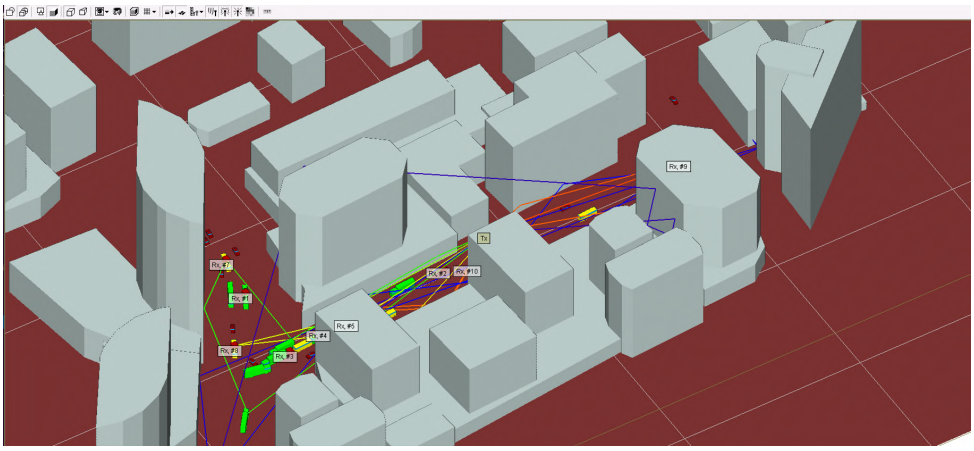

Figure 3.2: Simulation environment in Wireless InSite using 3D models and satellite data for signal propagation analysis.

the RT simulator, such as the positions of antennas, waveforms, frequencies, and other parameters. The Raymobtime MIMO channel is typically simulated during post-processing, as this provides greater flexibility in exploring array dimensions. For this purpose, it is necessary to place the Rx and Tx as isotropic antennas and to include as many antennas as required for the experiment. It is important to note that while the number of antennas can be reduced in later stages, increasing the number after the initial setup is not feasible.

In setting up a simulation with the Raymobtime methodology, several essential files are required to ensure proper configuration and execution. The default simulation name should be "model," and the corresponding files include .txrx, .study.xml, .setup, .city, and .object. These files serve distinct purposes: the .txrx file defines the transmitter and receiver parameters, the .study.xml file contains the study configuration, the .setup file includes the simulation settings, the .city file represents the imported 3D scenario, and the .object file outlines specific objects within the simulation environment. To maintain a baseline configuration, it is necessary to create a copy of each of these files and rename them with the prefix "base" (e.g., base.txrx, base.study.xml). These base files should be placed in the same directory as the original "model" files, ensuring they are readily available for reference or to reset the simulation to its default state.

Remember to adjust the material properties of the scenario according to the carrier frequency. Due to the increased computational cost of adding more realistic 3D objects with many faces, always optimize and review your models to ensure both integrity and good performance.

3.2.2 Setting Up SUMO for Mobility Simulation

To simulate realistic mobility patterns within the study area used in the WI setup, we must configure SUMO appropriately. This process start by importing the geographic data of the study area into SUMO. This data should match the location used in the WI simulation to ensure consistency. Use OpenStreetMap (OSM) data, which provides detailed maps that can be converted into SUMO-readable formats. The `netconvert` tool, provided by SUMO, is used to convert OSM data into a road network file (`.net.xml`).

```
netconvert --osm-files <file.osm> --output-file <file.net.xml>
```

Ensure that the OSM data covers the exact boundaries of your study area for accurate alignment with theWI environment.

After creating the road network, the next step is to generate the traffic routes. While SUMO provides scripts to generate random traffic, it is recommended to create custom scripts to generate routes that accurately represent the desired traffic patterns and vehicle distributions. These script must creates route files (`.rou.xml`) that include a fixed header specifying supported vehicle types and other simulation parameters. Ensure that the header is configured correctly to define the types of vehicles, such as cars, buses, and trucks, reflecting the traffic mix in the study area.

For example, a custom script might generate a route file with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<routes>
    <vTypeDistribution id="typeVehicleDistribution">
        <vType id="Car" departSpeed="max" accel="3" decel="4.5"
        length="4.645" width="1.775" height="1.59"
        maxSpeed="17.88" speedDev="0.1" sigma="0.2"
        minGap="0.3" probability="0.6"/>
        <vType id="Truck" accel="2.0" decel="4" length="12.5"
        width="2.5" height="4.3" maxSpeed="17.88"
        speedDev="0.1" sigma="0.2" minGap="0.3"
        probability="0.2"/>
        <vType id="Bus" accel="2.0" decel="4" length="9"
        width="2.4" height="3.2" maxSpeed="17.88"
        speedDev="0.1" sigma="0.2" minGap="0.3"
        probability="0.2"/>
    </vTypeDistribution>
     <flow id="flowA" color="0,0,1" begin="0" end= "3000"
     probability="0.98" type="typeVehicleDistribution">
        <route edges="laneA"/>
    </flow>
    <flow id="flowB" color="0,1,0"  begin="0" end= "3000"
    probability="0.98" type="typeVehicleDistribution">
        <route edges="laneB"/>
    </flow>
</routes>
```

Once the network and routes are defined, configure the simulation parameters in the SUMO configuration file (`.sumocfg`). This file should include references to the generated network and route files, as well as simulation settings like time steps and output files.

```
<configuration>
    <input>
        <net-file value="file.net.xml"/>
        <route-files value="custom_routes.rou.xml"/>
    </input>
    <time>
        <begin value="0"/>
        <end value="3600"/>
        <step-length value="1.0"/>
    </time>
</configuration>
```

It is possible run the SUMO simulation using either the SUMO GUI or CLI. The GUI is useful for visualizing traffic flow and validating route configurations, while the CLI is ideal for automated and batch processing.

```
sumo-gui -c <study_area.sumocfg>
```

Adjust the route generation scripts and simulation parameters as needed to ensure the simulation accurately reflects the conditions of the study area. This flexibility allows for finetuning traffic scenarios, such as adjusting vehicle types, traffic density, and other simulation factors.

3.2.3 Raymobtime Simulation Setup

To set up a simulation using the Raymobtime framework, some software tools and scripts must be properly configured and executed. This subsection details the necessary steps, from code organization to running the simulations, providing a structured approach to simulate realistic mobility and wireless channel conditions. All related projects for the simulation setup start with the prefix `5gm` on GitHub. The simulation environment explained above is assumed using Linux and Python 3. The following steps outline the installation process:

Quick Installation: To quickly set up the environment, first create and activate a Python virtual environment. Then, install the required Python packages and clone the necessary GitHub repositories:

```
# Clone the simulation repository and install
git clone https://github.com/lasseufpa/5gm-rwi-simulation
cd 5gm-rwi-simulation
# If you have a Python virtual environment, activate it
# Install all required packages
pip install -r requirements.txt
python3 setup.py install
# Clone the data processing repository and install
cd ..
git clone https://github.com/lasseufpa/5gm-data
cd 5gm-data
python3 setup.py install
```

After a successful installation, the directories `5gm-data` and `5gm-rwi-simulation` should be present in the specified location, and several Python packages, including `rwimodeling`, `rwiparsing`, and `rwisimulation`, will be installed.

These three Python packages are fundamental to the simulation setup. The `rwimodeling` package manages the manipulation of 3D objects within the InSite files, enabling modifications and analysis of environmental geometry. This package works closely with `rwiparsing`, which handles the proprietary file formats of InSite, ensuring that data is correctly interpreted and processed. Finally, the `rwisimulation` package orchestrates the entire simulation process, leveraging the functionalities provided by both `rwimodeling` and `rwiparsing` to run the simulations efficiently and accurately.

Ensure that the Wireless InSite is installed in your setup and accessible through your system's PATH environment variable for seamless execution of the ray-tracing simulations.

3.2.4 Running the Simulation

To run the simulation, you first need to configure the `config.py` file, which can be found in the "example" folder of the `5gm-rwi-simulation` repository. This file contains essential parameters that define the paths and settings for the simulation environment. Open `config.py` in a text editor and adjust the following key variables:

`base_insite_project_path` : This variable should point to the folder with the base simulations files created ate subsection 3.2.1.

`results_dir` : Define the directory where all simulation outputs will be stored.

`wibatch_bin` and `calcprop_bin`: These variables should specify the paths to the respective executables for running different simulation modes in WI.

`sumo_bin` : Set this to the directory where SUMO is installed in you machine. For instance: `/usr/bin/sumo`.

`sumo_cfg`: This variable defines the path to the SUMO configuration file, which contains the simulation settings and network routes.

`n_run`: This variable defines an iterator that controls the number of ray-tracing simulations samples to be executed, running from 0 to 99. It sets the maximum number of simulations to 100, iterating one simulation at a time.

`sampling_interval`: This variable sets the time interval between scenes during the simulation, measured in seconds.

`time_of_episode`: This variable defines the total duration of each episode in terms of the number of scenes.

`time_between_episodes`: This variable specifies the time, in seconds, determining the pause between episodes.

You can also customize the simulation's behavior through the following variables, which control different modes:

`use_fixed_receivers`: Set to True use the receivers in the same positions placed in the base simulation file; otherwise, set to False to focus on mobile receivers.

`use_pedestrians`: Set to True to include pedestrians in the simulation, provided your SUMO setup supports them.

`use_vehicles_template`: Set to True to use pre-made 3D vehicle models instead of generic boxes. Ensure the objects folder containing the models is present.

`drone_simulation`: Set to True if only drones should act as receivers in the simulation, provided your SUMO setup supports them.

`use_V2V`: Set to True to enable Vehicle-to-Vehicle (V2V) communication, where both transmitters and receivers are vehicles.

These variables are the most commonly adjusted to tailor the simulation to specific scenarios. However, there are several other configuration options available in the `config.py` file. Users should carefully review these additional variables to ensure that all relevant parameters are set correctly based on their specific simulation requirements. After configuring these variables, you're ready to run the simulation.

Next, execute the simulation script by navigating to the `5gm-rwi-simulation/example` directory and using the following commands:

```
python3 -m rwisimulation.simulation -po
python3 -m rwisimulation.simulation -rj
```

Here's what the individual commands do:

`-po` (Placement Only): This command runs the pre-processing stage of the simulation. It parses the scenario files, sets up the 3D environment models, and configures the SUMO traffic patterns. During this step, it integrates mobility data from SUMO and updates the InSite files with accurate object placements. Additionally, it creates multiple run folders, each representing an individual simulation, contributing to the overall mobility dataset.

`-rj` (Run Job): This command executes the full simulation, using the environment and mobility data set up in the pre-processing phase. It triggers Wireless InSite to simulate wireless propagation for each run folder, calculating the channel characteristics and metrics. If a simulation folder has already been processed, the `-rj` command skips it, ensuring efficient execution by only processing pending runs.

By following these steps, you will be able to configure and execute a comprehensive simulation environment that models complex wireless communication scenarios for research and development purposes.

3.2.5 Post-Processing and Analysis

After completing the simulation runs, each folder will have individual raw RT output data, including the MPC information and channel characteristics. To facilitate the usage of the data, it must be must be organized and converted into more accessible formats for further analysis. The raymobtime results are typically stored in a SQLite database file (`episodedata.db`) and can also be transformed into matrix formats such as Hierarchical Data Format - Version 5 (HDF5), which are convenient for use in Python and MATLAB.

Storing Results in SQLite: The ray-tracing and SUMO simulation outputs can be stored in an SQLite database for easier management and querying. To do this, move the simulation files to the `5gm-data` project directory and use the following commands to insert the raw data into an SQLite database named `episodedata.db`:

```
cd 5gm-data
python3 todb.py rt_results
```

Mobile Simulations: For simulations with mobile receivers, you can convert the `episodedata.db` file into HDF5. Before proceeding, ensure the variables `numScenesPerEpisode` and `fileNamePref` in `convert5gmv1ToChannels.py` are correctly set. Then, run the conversion script to organize the data further:

```
python3 convert5gmv1ToChannels.py
```

Fixed Simulations: For simulations with fixed transmitters and receivers, you can convert the ray-tracing output into formats suitable for Python and MATLAB. Then, run the following command to generate the matrix files:

```
python3 collectFixedChannels.py rt_results
```

V2V Simulations: For vehicle-to-vehicle (V2V) simulations, additional adjustments are required. Ensure that the variables `numScenesPerEpisode`, `numTx`, and `numRx` in the script `collectChannels_from_V2V.py` are properly set according to your scenario. Once configured, run the following command to convert the data into HDF5 files:

```
python3 collectChannels_from_V2V.py rt_results output
```

The primary output format is an HDF5 file containing a 4D structure with the dimensions: number of scenes, transmitter-receiver pairs, rays, and path parameters. These parameters include metrics such as received power, time of arrival, angles of departure and arrival, and line-of-sight flags of each MPC. This ensures that the simulation data can be efficiently analyzed and visualized, supporting further research and development efforts. The structure of the HDF5 output is detailed in Table 3.4, ensuring all key channel characteristics are stored for further analysis.

Table 3.4: Ray-Tracing Data Output Format

Archive Type	**Archive Format**	**Order of Path Parameters**
HDF5	4D structure: [scenes, TX-RX pairs, max rays, parameters]	1. Received power (dBm) 2. Time of arrival (s) 3. Elevation angle of departure (degree) 4. Azimuth angle of departure (degree) 5. Elevation angle of arrival (degree) 6. Azimuth angle of arrival (degree)) 7. LOS flag ('1' = LOS, '0' = NLOS) 8. Ray phase (degree)

3.3 V2I 3GPP-Based Datasets Scenarios

This section outlines the datasets generated based on specifications discussed within the 3GPP Radio Access Network - 1 (RAN-1) study group, which focuses on Physical (PHY) layer. The group is actively engaged in several topics but the data discussed in this work were collected from discussions on AI/ML in beam management topic.

Two primary scenarios have been modeled for this research: Rosslyn and Marselha. The Rosslyn scenario is a well-established example within the Raymobtime framework, recognized for its representation of an urban canyon environment. This scenario effectively captures the complexities of wireless communication in densely built-up areas, making it a valuable benchmark for evaluating beam management algorithms.

In contrast, the Marselha scenario was chosen to illustrate a more residential context. This scenario enables the exploration of beam management performance in environments characterized by lower density and different mobility patterns. By analyzing both urban and residential scenarios, this research aims to provide a comprehensive understanding of how varying environments influence the effectiveness of beam management strategies.

The Table 3.5 presents the details of the 3GPP-based datasets, including key parameters such as frequency, receiver type, and scene intervals. The 3D representations of these scenarios are shown in Figures 3.3 and 3.4. These datasets serve as a foundation for further analysis and experimentation, aligning with the specifications set forth by the 3GPP discussions.

Table 3.5: 3GPP-Based Datasets

ID	**Scenario**	**Freq. GHz**	**Rx(Type)**	**Scene Interval**	**Episodes Interval**	**Episodes**	**Scenes per Episode**	**Valid Ch.**
t005	Marselha	30	5 Mobile	80 ms	30 s	145	40	29K
t006	Rosslyn	30	5 Mobile	80 ms	30 s	145	40	29K

As can be verified in the meeting document R1-2407554 [35], the discussion group which focuses on the evaluation of AI/ML techniques for beam management places primary importance on Downlink (DL) transmit beam prediction, that can be separated into two key scenarios:

- BM-Case1 - Spatial-domain DL Tx beam prediction: This approach predicts Set A beams using measurement results from Set B beams, capitalizing on spatial correlations to enhance beam selection accuracy.
- BM-Case2 - Temporal DL Tx beam prediction: In this case, predictions for Set A beams are derived from historical measurement data of Set B beams, which allows the system to

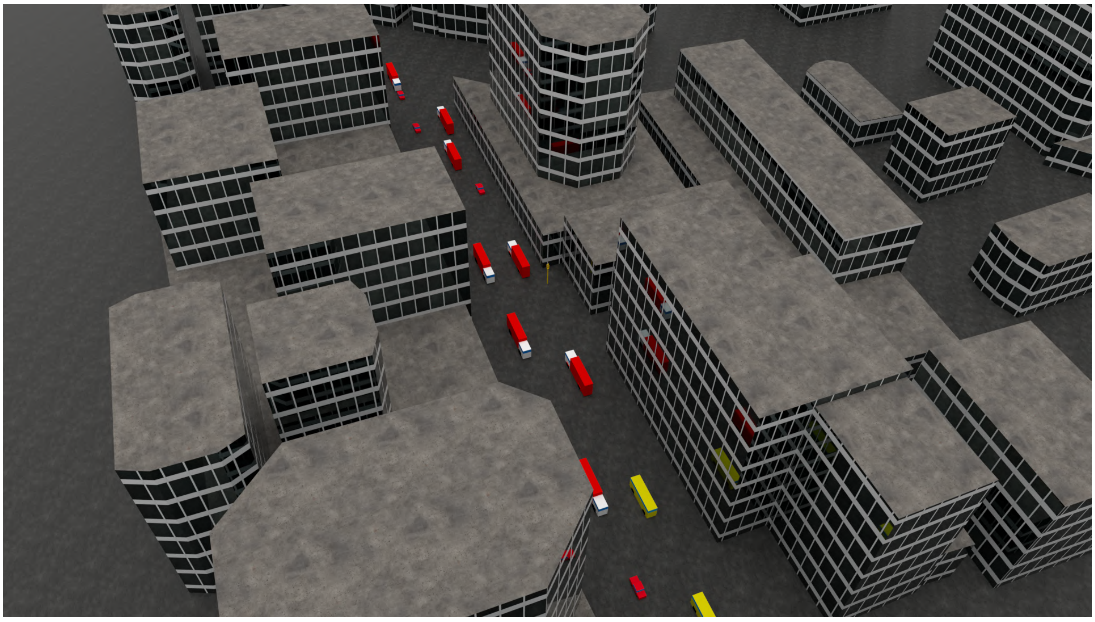

Figure 3.3: Simulated scenario: Rosslyn (in USA) Urban-canyon

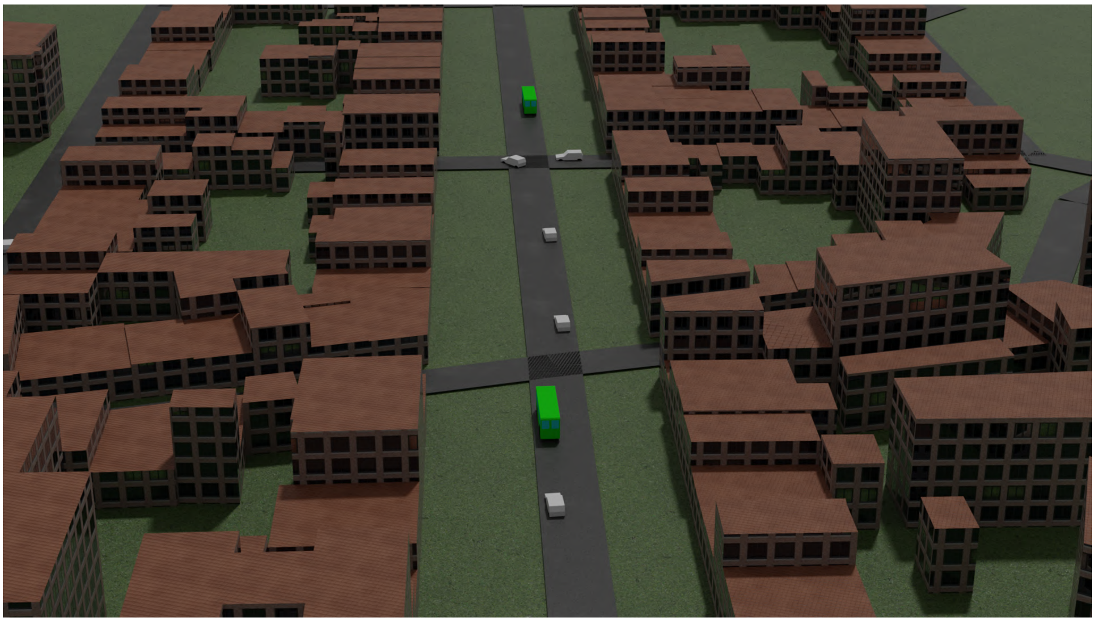

Figure 3.4: Simulated scenario: Marselha (in France) Suburban

consider the time-dependent evolution of the environment for future beam predictions.

The 3GPP has not yet reached a consensus on the definitions of Set A and Set B, leading to scenarios where one may be a subset of the other or where both sets are entirely independent. In this work, we will present a configuration proposal for Set A and Set B in Section 4.

The parameter specifications used in our both simulations are closely aligned with ongoing discussions within the 3GPP RAN1 study group, particularly the insights from meeting document R1-2306199 [7]. These specifications are derived from the scenarios outlined in the meeting discussions, ensuring compliance with emerging standards in beam management for 5G and beyond.

The frequency range is set to 30 GHz, consistent with the mmWave frequency band, which is crucial for high-capacity and high-speed data transmission in vehicular communication scenarios. In the 3GPP discussions, two distinct mobility profiles were considered for evaluation. The first, labeled as Set A, simulates low-mobility conditions with a a UE speed of 3 km/h, while Set B incorporates higher mobility scenarios with speeds of 30 km/h and 60 km/h. These profiles capture a variety of mobility conditions typically encountered in urban and suburban environments. For t003 and t004 datasets only the Set B configuration, which focuses on higher mobility, was adopted to better reflect the challenges of fast-moving vehicular environments.

The base station (BS) transmission power is specified at 40 dBm, ensuring adequate coverage and signal strength across the selected environments. The BS antenna is mounted at a height of 25 meters, with a downtilt angle of 110°, which optimizes beamforming for both near-field and far-field UE locations. The inter-site distance between base stations is set to 200 meters, representing a typical urban deployment scenario.

The UE distribution for beam prediction is outlined in 3GPP specification, distinguishing between the spatial and temporal prediction domains. For spatial domain beam prediction, the UE distribution follows the guidelines in TR 38.901 [43], with 80% of UEs positioned indoors and 20% outdoors. An alternative scenario with 100% outdoor UEs is also considered. For time-domain prediction, all UEs are assumed to be outdoors to account for mobility dynamics. Given that our application focuses on beam tracking, where mobility plays a crucial role, our dataset adopts the 100% outdoor configuration to better reflect the dynamic conditions of the target scenarios.

Chapter 4

System and Architecture Overview

In this chapter, we present the overall system and architecture developed for efficient beam-tracking in V2I communications using mmWave MIMO systems. With the growing demand for high-speed, low-latency communication in next-generation networks, beam-tracking has emerged as a crucial technique to maintain robust connections in dynamic environments. This work addresses the challenges posed by rapid vehicular movement and complex multipath propagation.

The system integrates a new beam-tracking approach with machine learning techniques, specifically leveraging a RNN to enhance the ability to predict and adapt to beam changes in real-time in comparison with previous works.

4.1 Beam-Tracking Proposed Approach

The proposed beam-tracking system is designed to adapt to different dynamic scenarios of vehicular networks by employing deep learning to select the optimal beam in real time. This approach leverages historical beam data as input for accurate predictions.

4.1.1 Input Hierarchical Dimensionality Reduction

To efficiently optimize the input size, the system utilizes a technique similar to hierarchical beamforming [33] to reduces the overhead associated with exhaustive beam searches. Based on the estimate position of the UE obtained by CSI [44] the set of all beams is geometrically reduced to a subset of potential N candidates, referred to as Set A, following the current 3GPP nomenclature defined in [35]. This reduction is achieved by identifying beams that are oriented towards the estimated location of the UE, as illustrated in Figure 4.1. The selection process considers the angular proximity of each beam in Set A, retaining only those that fall within a specified angular sector around the UE's estimated position.

The angular selection process involves computing the relative angle between each beam's direction and the estimated UE position. The angle θ between the BS position P and the UE position Q is calculated as:

$$\theta = \text{arctan2}(y_Q - y_P, x_Q - x_P) \cdot \frac{180}{\pi}, \tag{4.1}$$

where (x_P, y_P) and (x_Q, y_Q) are the coordinates of the BS and UE, respectively. The resulting angle is normalized to lie within the range $[-180°, 180°]$ for consistency.

To form Set A, the system selects the N beams with directions closest to θ, based on their absolute angular differences from θ. This step minimizes redundancy by prioritizing beams most aligned with the estimated UE position, as shown in Figure 4.1.

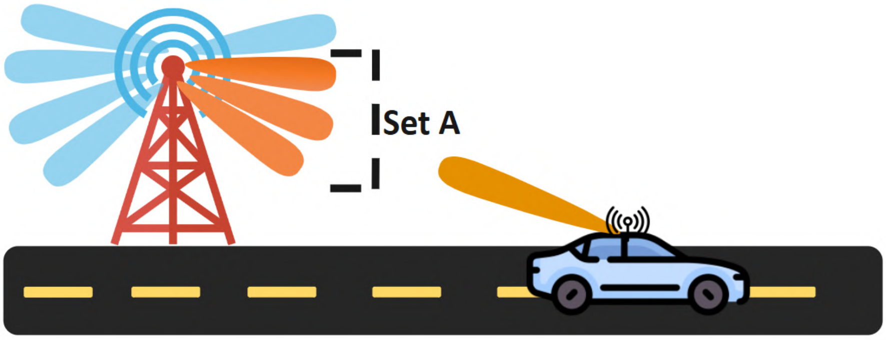

Figure 4.1: Beam set spatial reduction.

Once the N candidates in Set A are determined, the time series input can be build.

4.1.2 Historical Beam Time Series Input

The method utilizes historical beam data, structured as shown in Figure 4.2, to predict the optimal beam. The system adopts a periodic RSRP measurement strategy, where RSRP (measured in Watts) is recorded within an observation window of size 4 (scenes). In this methodology, a beam measurement is interleaved with two predictions. As a result, each input incorporates at least two real RSRP measurements, with the remaining data consisting of predictions. This frequency ensures that the system can respond quickly to changes in the environment.

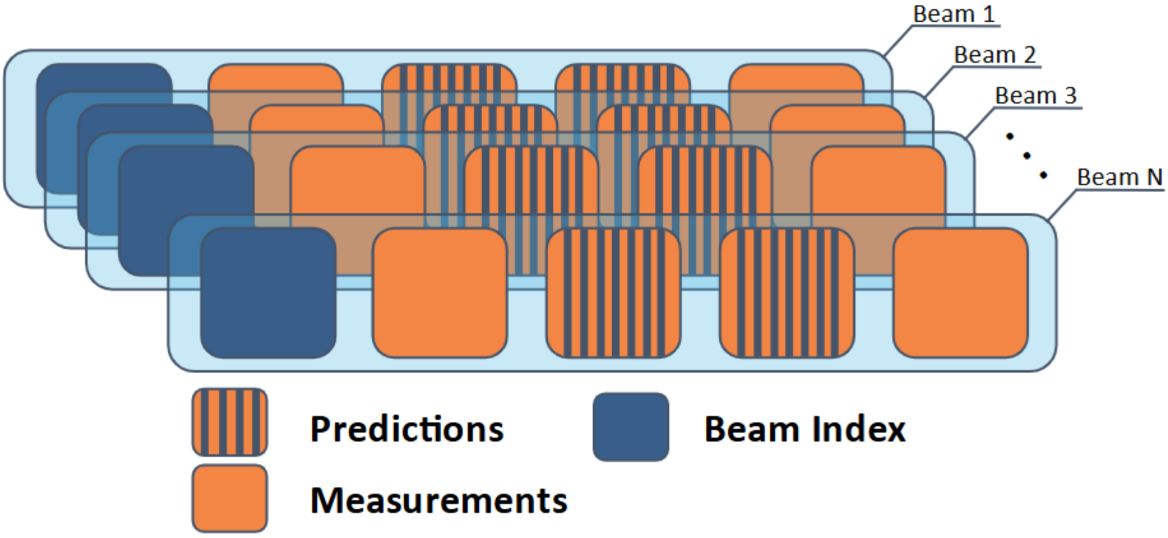

Figure 4.2: Structure of historical beam data as input for the RNN.

The RSRP is used as input to provide a better understanding of the beam's signal strength, allowing the system to track variations over time. Additionally, the beam index is also included in the input. This is crucial because it provides a structured way to handle the beam search process when hierarchical dimensionality reduction is applied, narrowing the scope to the most probably relevant beams. This way the system is able to efficiently manage the complexity of the beam-tracking process while maintaining the temporal context necessary to monitor changes in the communication channel.

In this work beam measurements are taken every 240 ms, while beam predictions are performed more frequently, at intervals of 80 ms between the measurements. Figure 4.3 illustrates an example input considering a Set A with $N = 2$, where the selected beams are the ones indexed as 32 and 33, and their corresponding hypothetical RSRP values. In beam 32, the 0.6 and 0.8 are values derived directly from physical measurements, accounting for the overhead of real-time data acquisition. On the other hand, predicted values, such as 0.8 and 0.7, are estimated by the model.

By relying on predictions to fill in the gaps between measurements, the approach significantly reduces the overhead compared to scenarios where measurements are taken at every timestamp. For example, if beam measurements were performed at the same frequency as the predictions, the overhead would duplicated, depending on the scenario and the number of beams being tracked.

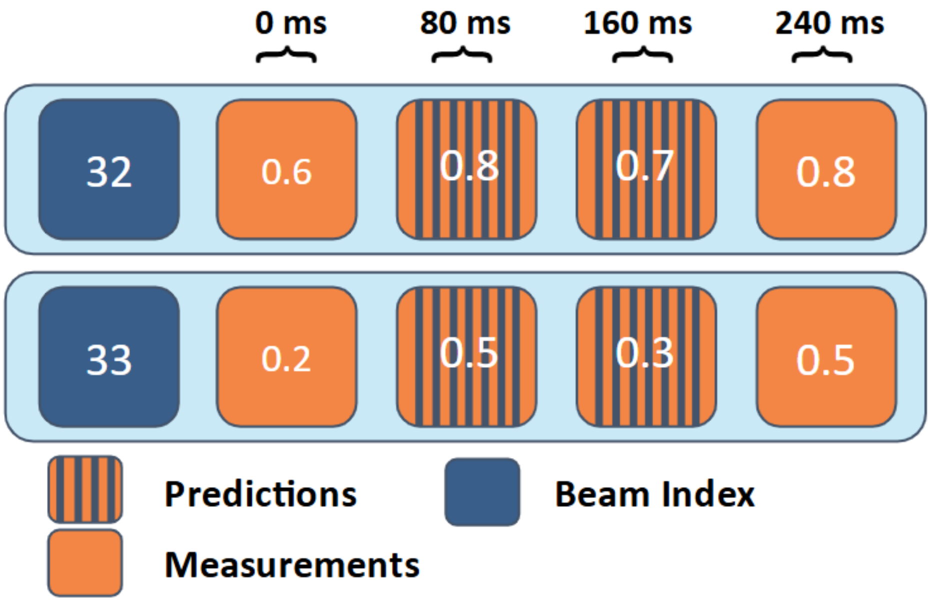

Figure 4.3: Example of historical beam information as input to the RNN, with selected beams and RSRP values.

4.1.3 Recurrent Neural Network Architecture

As shown in Figure 4.4, the proposed tracking method is realized through an RNN model, specifically using LSTM layers, designed to handle the sequential historical beam data, predicting future beam behavior based on past trends.

The architecture leverages multiple LSTM layers to handle the dynamic nature of vehicular scenarios by extracting features from time-series input. Dropout layers are integrated to prevent overfitting, ensuring robust performance across varying conditions. Dense layers further refine the predictions, producing outputs that align with the number of candidate beams.

This hierarchical structure effectively handles sequential data, balancing model complexity and generalization. By leveraging the sequential processing capabilities of LSTM layers with regularization and dense transformations, the architecture addresses the challenges of real-time beam tracking, such as fluctuating channel conditions and limited historical data. This approach balances accuracy and efficiency, making it well-suited for dynamic vehicular environments.

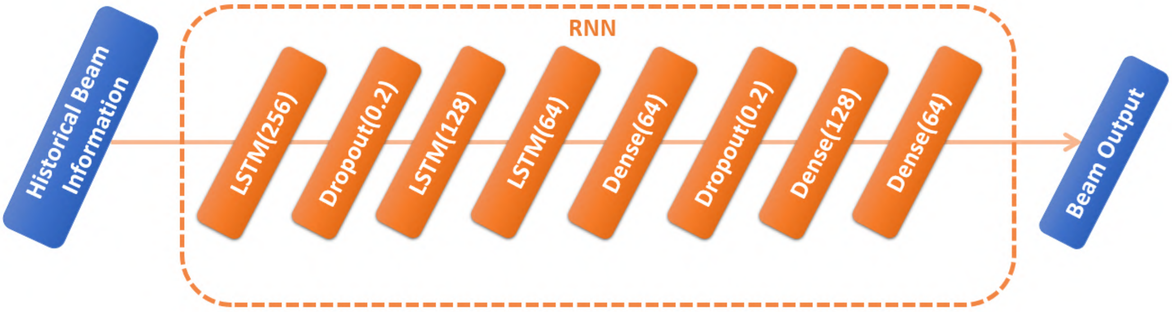

Figure 4.4: Proposed beam tracking RNN architecture.

This architecture will be evaluated on two primary and independent tasks, designed to assess its effectiveness in beam-tracking scenarios:

4.1.3.1 Beam Index Classification

The first task involves classifying the optimal beam index for the next time step. Given the sequential historical beam data, the model predicts the most probable best beam index from the candidate set. The model's performance will be trained and evaluated using the top-K accuracy metric, ensuring its capability to identify the optimal beam under dynamic conditions.

4.1.3.2 Beam RSRP Regression

The second task focuses on predicting the RSRP for each candidate beam. This regression task provides a quantitative measure of signal strength, enabling the system to rank beams based on their predicted RSRP values. The model is trained using Mean Squared Error (MSE) as the loss function.

Additionally, the regression predictions will also be used to evaluate top-K accuracy. The beam index corresponding to the highest predicted RSRP will be extracted and compared to the ground truth to assess the model's ability to identify the optimal beam within the top-ranked candidates.

Hence, it is important to note that, after the model performs *regression*, we will use the outputs to perform top-K *classification*.

Chapter 5

Simulations and Results

This chapter outlines the experimental setup and presents the results of the proposed beam-tracking system. The simulations are conducted to evaluate the system's performance in dynamic vehicular scenarios, focusing on two core tasks: beam index classification and RSRP regression. The evaluation employs key metrics such as Top-K accuracy to assess the model's effectiveness, and Mean Absolute First Difference (MAFD), and beam index switching rate to provide insights into the dataset's characteristics. Additionally, the results are benchmarked against the similar LSTM based approach proposed by Zhao et al. [14], and a LIDAR-based CNN baseline originally proposed in [11], which was later enhanced with residual layers [45] and utilized in [46], resulting in the final architecture define by Suzuki et al. [12] showed in Figure 5.1.

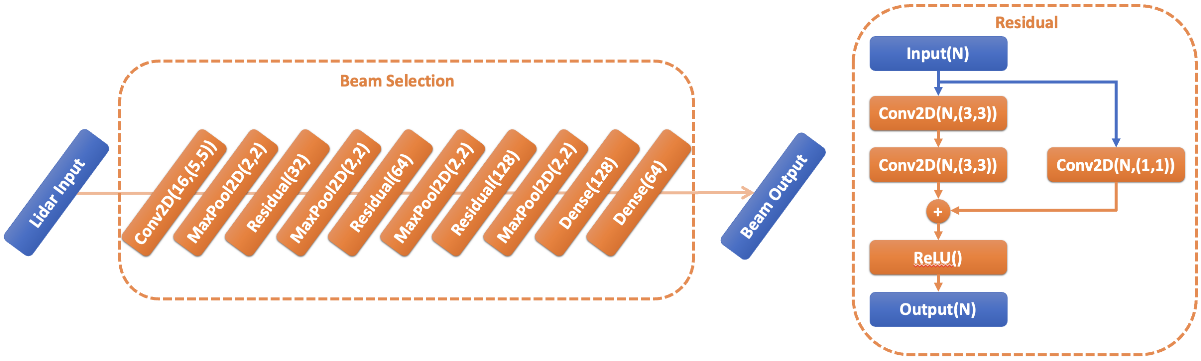

Figure 5.1: CNN baseline architecture.

This comparison underscores the need for a deeper analysis of the vehicular communication environment to fully understand the factors influencing system performance. The following section delves into some characteristics of the V2I wireless channels datasets generated for this research, which play a critical role in beam-tracking accuracy and reliability.

5.1 V2I Wireless Channels Analysis

The analysis of V2I wireless channels is essential to understanding the dynamics of beam-tracking in real-world vehicular scenarios. This section evaluates the characteristics of the datasets used in the simulations, focusing on LOS and NLOS proportions, as well as key metrics such as the MAFD and beam index switching rate.

5.1.1 Line-of-Sight Proportion

The LOS and NLOS distribution provides insights into the propagation conditions of the datasets. Table 5.1 summarizes the LOS and NLOS sample counts for the Marselha and Rosslyn scenarios. Both datasets exhibit a high proportion of LOS samples, reflecting typical V2I environments where clear visibility often dominates, mainly due to the 3GPP specifications [7] about the base-station antenna height (25 meters). However, the presence of NLOS samples highlights the challenges of beam management in obstructed scenarios, emphasizing the importance of robust tracking mechanisms.

Table 5.1: Datasets Line of Sight proportion

Dataset ID	**LOS**	**NLOS**
Marselha - t005	27800	2200
Rosslyn - t006	27899	2101

5.1.2 Beam Index Dynamics

Some metrics are employed to characterize the beam index dynamics in the datasets: MAFD, the beam variance and beam index switching rate. These metrics quantify the temporal variability and stability of beam selection, which are crucial for evaluating the tracking system's ability to adapt to rapid changes in communication channels.

The MAFD makes sense when consecutive beam indices represent neighboring (consecutive) angles, and is defined as:

$$\text{MAFD} = \frac{1}{R}\sum_{r=1}^{R}\frac{\sum_{s=1}^{S-1}\mathbf{D}_{s_r}}{S-1}, \tag{5.1}$$

where R is the number of receivers, S is the number of scenes in the given episode, and $\mathbf{D}_{s_r}$ represents the difference between beam indices $\hat{\imath}_{s_r}$ and $\hat{\imath}_{s_r-1}$ of consecutive scenes, and is given by

$$\mathbf{D}_{s_r} = \min((\hat{\imath}_{s_r-1} - \hat{\imath}_{s_r}) \bmod N_{total}, (\hat{\imath}_{s_r} - \hat{\imath}_{s_r-1}) \bmod N_{total}). \tag{5.2}$$

The *mod* N_{total} operation implements a "wrap around", with N_{total} being the total number of beams. For instance, assuming $N_{total} = 3$, a given receiver $r = 5$, an episodes with $S = 7$ scenes and a sequence of beam indices $\hat{\imath}$ given by

$$1, 1, 0, 2, 1, 2, 0;$$

the respective sequence $\mathbf{D}_{s_5}$ is

$$0, 1, 1, 1, 1, 1.$$

Note the wrap around $N_{total} - 1 = 2$ in this case, such that the last value of $\mathbf{D}_{s_5}$ is 1 (not 2), characterizing a closed loop.

The beam index switching rate is calculated as:

$$\text{Index switching rate} = \frac{1}{R}\sum_{r=1}^{R}\frac{\sum_{s=1}^{S-1}1(\hat{\imath}_{s_r} \neq \hat{\imath}_{s_r-1})}{S-1}, \tag{5.3}$$

this rate indicates how often the selected beam index changes from one time instant to the next.

Table 5.2: Datasets beam change rate.

Dataset ID	MAFD	Variance	Index switching rate
Marselha - t005	0.922	130.24	0.3481
Rosslyn - t006	2.032	310.01	0.3495

Key Observations

The results in Table 5.2 reveal distinct patterns in beam dynamics for the Marselha and Rosslyn datasets:

- The beam index switching rates are comparable for both datasets, suggesting similar number of beam indices changes. However, this rate does not indicate by how much the indices changed. This is something that the variance and the MAFD can inform.
- The Marselha dataset exhibits a lower MAFD, indicating smoother beam transitions, mainly in the shift between LOS channel and NLOS channel, resulting in a more stable channel conditions (in the sense that the best beam index does not vary much).
- The Rosslyn dataset shows a higher MAFD, reflecting more significant beam angular variations, likely due to the urban canyon environment.

These findings highlight the varying demands of different vehicular environments on beam-tracking systems and underline the importance of adaptability in the proposed method.

5.2 Beam Tracking Evaluation

This section presents the performance evaluation of the proposed beam-tracking system. The analysis focuses on comparing the proposed RNN-based model with a CNN-based baseline to assess their effectiveness in dynamic vehicular scenarios. The comparison encompasses various dimensions, including model complexity, input size requirements, and overall performance in beam-tracking tasks.

Table 5.3 summarizes the key specifications of the models used in the evaluation. The RNN-based model leverages its sequential processing capabilities to handle historical beam data, while the CNN baseline processes LIDAR data, which inherently requires larger input sizes. Both models differ slightly in the number of trainable parameters and have a significant difference in memory footprint, reflecting distinct trade-offs in computational and storage requirements.

Table 5.3: Models Specifications

Model	Parameters	Model size	Input mean size
RNN	535,552	2.04 MB	1.25 KB
CNN	514,000	1.96 MB	1048 KB

The input sizes were calculated based on the memory requirements of the arrays used to represent the data. Specifically, the memory size corresponds to the dimensions and datatype of the arrays processed by each model during training. For instance, the RNN processes smaller arrays derived from sequential historical data, while the CNN requires larger arrays to accommodate the spatial dimensions of LIDAR inputs.

Table 5.3 illustrates that the RNN model achieves a comparable number of parameters and model size to the CNN, despite using a significantly smaller input size. This reduction also minimizes computational demands and lowers the memory requirements for processing. The RNN's efficiency stems from its ability to extract meaningful features from sequential historical data, eliminating the need for the large-scale spatial inputs required by the CNN.

5.2.1 Top-*K* Accuracy

Top-*K* Accuracy is a metric used to evaluate a model's performance in multi-class classification or ranking tasks. Unlike standard accuracy, which only considers the top prediction, Top-*K* accuracy checks whether the true label is within the Top-*K* predicted values, ranked by their probabilities or scores. In our model, this metric is applied as follows:

- For **classification tasks**, a prediction is considered correct if the true class label is within the Top-*K* predictions sorted by their associated probabilities.
- For **ranking tasks**, such as beam RSRP prediction, the metric evaluates whether the true beam is included among the Top-*K* candidates ranked by their predicted RSRP values.

The formula for Top-*K* accuracy is:

$$\text{Top-}K\text{ Accuracy} = \frac{\text{Number of samples where the true label is in the top-}K\text{ predictions}}{\text{Total number of samples}}$$

.

Top-*K* accuracy is particularly relevant for beam management systems, as it measures how often the correct beam index or candidate is included among the Top-*K* predictions. By identifying a smaller set of high-probability candidates, this metric can inform strategies that reduce the frequency of exhaustive beam sweeps in dynamic scenarios, potentially minimizing overhead while maintaining system performance.

The subsequent sections provide a detailed comparison of the proposed and baseline models performance across classification and regression tasks using the Top-*K* accuracy

5.3 Performance Analysis and Visualization

The performance of the proposed beam-tracking approach is evaluated using the Top-K accuracy metric across various configurations and datasets. The results are presented in Figures 5.2 to 5.6 and are analyzed below.

5.3.1 Comparison of Models Performances

The performance analysis, visualized in Figures 5.2 through 5.6, highlights key insights into the behavior of the proposed beam-tracking model.

In Marselha-T005 (Figure 5.2), the classification models outperform the regression models. However, the regression models still achieve high absolute accuracy levels, demonstrating their viability for beam tracking. The strong performance of the classification approach in this scenario suggests that it effectively leverages the dataset's topological characteristics. Nevertheless, as evidenced by the results on the Rosslyn-T006 dataset, classification models may face limitations and struggle to maintain consistent performance across varying scenarios.

In T006 (Figure 5.3), the performance gap between classification and regression models becomes more pronounced. The classification models, both RNN and CNN, underperform significantly, likely due to the higher MAFD of the dataset. This increased MAFD introduces greater complexity in identifying the correct beam index, as it disrupts the predictable patterns in the output labels. In contrast, the regression models maintain robust performance across both datasets, showcasing their ability to generalize effectively to more complex and dynamic scenarios. This robustness positions regression models as strong candidates for real-world deployment, where adaptability and consistent performance are critical.

Comparing the baselines with the proposed model in Marselha-T005 (Figure 5.2), the RSRP regression model outperforms the *Zhao et al. LSTM* model by achieving higher Top-K accuracy and compared to the *Suzuki et al. CNN*, the proposed model achieves equivalent performance. However, it achieves this result with a significantly lower computational burden.

In Rosslyn-T006 (Figure 5.3), the RSRP regression model outperforms both baselines. These results highlight the model's ability to adapt to the dynamic propagation conditions present in the Rosslyn-T006 dataset. Below is a concise analysis of the results obtained in both scenarios, highlighting key performance trends and insights.

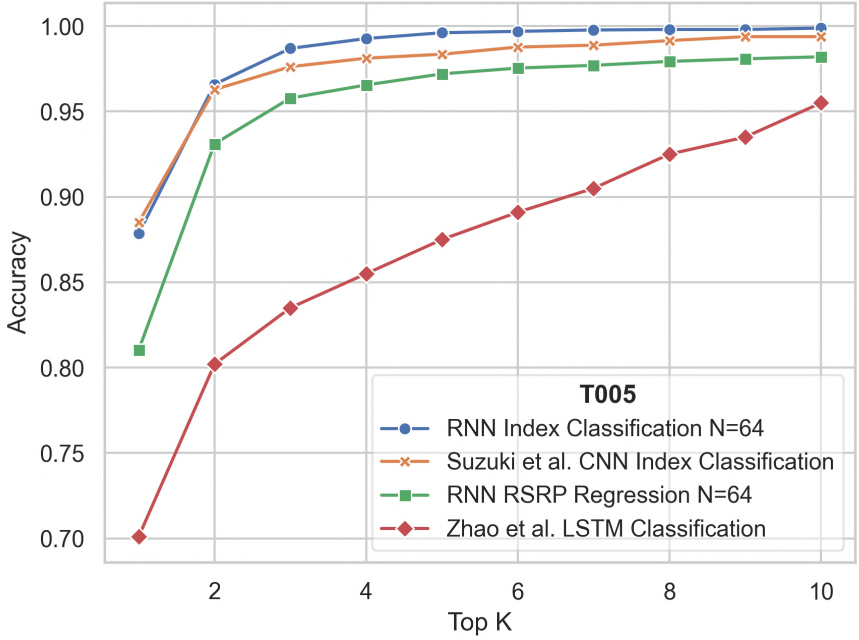

Figure 5.2: T005 beam tracking Top-K accuracy.

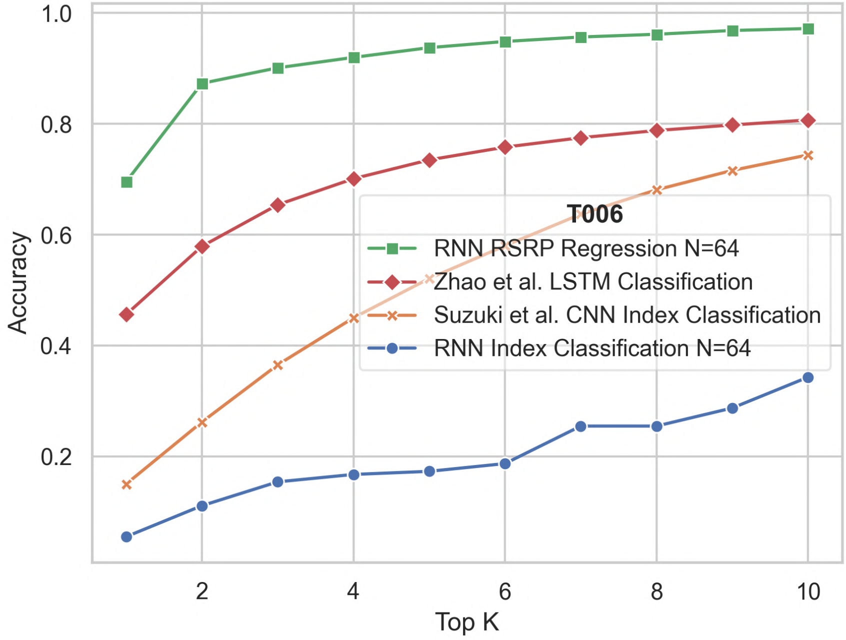

Figure 5.3: T006 beam tracking Top-K accuracy.

Summary of results obtained in Marselha-T005

- **RNN Index Classification** ($N = 64$) outperforms other models, achieving nearly 99.9% accuracy at $K = 10$, demonstrating its superiority in beam index prediction.

- **RNN RSRP Regression** ($N = 64$) exhibits consistent improvement, reaching a maximum accuracy of 98.2% for $K = 10$.
- **Suzuki et al. CNN Index Classification** with LIDAR data, shows high accuracy, reaching 99.4% for $K = 10$, highlighting the benefit of incorporating multimodal data with spatial information.
- **Zhao et al. LSTM Classification** achieves 95.5% at $K = 10$, suggesting that the lower variance and reduced MAFD value may facilitate the beam tracking process.

Summary of results obtained in Rosslyn-T006

- **RNN RSRP Regression** ($N = 64$) shows robust performance, achieving 97.2% at $K = 10$, although slightly lower than T005 due to increased dataset complexity.
- **Zhao et al. LSTM Classification** achieves 80.07% at $K = 10$, which is lower than the result reported in the reference paper. This difference is justifiable, as although the scenario (Rosslyn) is the same, the dataset was modeled differently, leading to variations in the evaluation conditions.
- **Suzuki et al. CNN Index Classification** with LIDAR data, improves progressively, achieving 74.4% at $K = 10$.
- **RNN Index Classification** ($N = 64$) performs poorly, achieving an accuracy of only 34.3% at $K = 10$, likely due to dataset-specific challenges.

5.3.2 Impact of Set A Size

The impact of varying the input size (*N*) on beam tracking accuracy is illustrated in Figures 5.4 and 5.5. As expected, increasing *N* generally improves accuracy across the Top-*k* values, likely due to the larger data context available for decision-making. However, interesting trends emerge when analyzing specific scenarios.

In the Marselha-T005 dataset, an intriguing observation is that the model with *N*=32 outperforms the model with *N*=64, despite using a smaller input size. This result highlights the potential of achieving superior accuracy with reduced computational overhead, which is advantageous for scenarios requiring low-latency or resource-constrained environments. This unexpected behavior may be influenced by the dataset's MAFD, which can create scenarios where an intermediate input size captures the most relevant information without redundancy.

In contrast, for the Rosslyn-T006 dataset, the model with N=64 maintains a slight advantage over N=32, though the difference in accuracy is minimal. This close performance suggests that, for the simulated scenarios, using N=32 remains a viable alternative for scenarios where reducing overhead is critical, offering a good balance between computational cost and accuracy.

A noteworthy observation for both datasets is the behavior of the model with N=5. Due to the limited input size, its accuracy plateaus at Top-k values greater than five, as the reduced input inherently limits its ability to provide diverse predictions. While the performance is understandably lower compared to larger input sizes, N=5 still offers insights for applications where resource constraints outweigh accuracy requirements.

Overall, the results emphasize the importance of tailoring input size to the specific scenario, balancing accuracy and computational efficiency while accounting for dataset characteristics such as MAFD.

A concise evaluation of the outcomes in both scenarios is provided below, showcasing key performance patterns and observations:

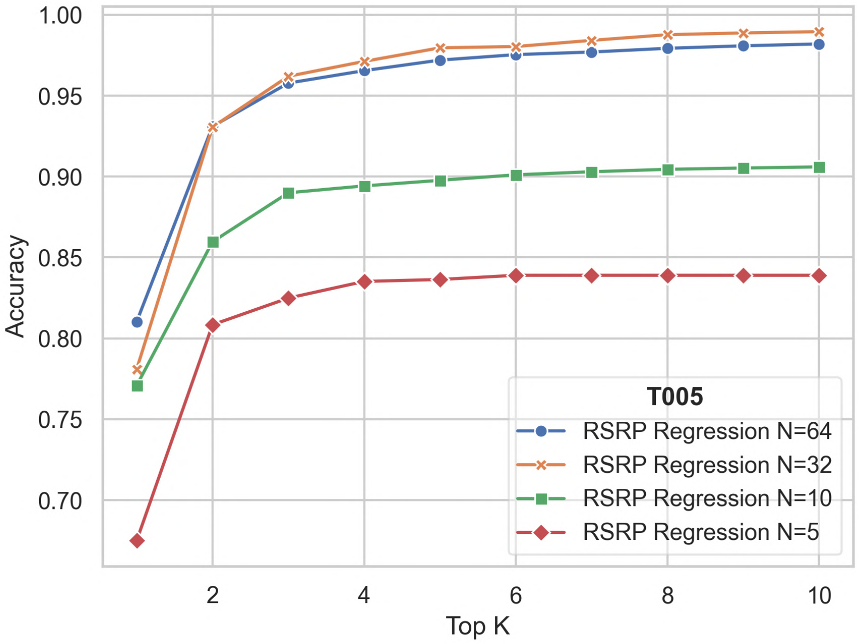

Figure 5.4: T005 beam tracking Top-K accuracy with varying input sizes.

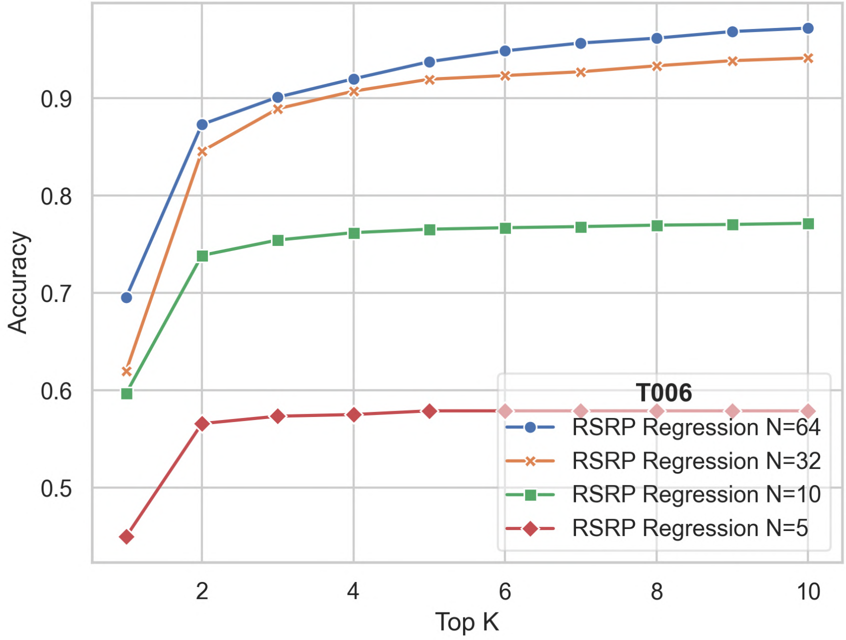

Figure 5.5: T006 beam tracking Top-K accuracy with varying input sizes.

Summary of Results Obtained by Varying Input Size in Marselha-T005

- $N = 32$ achieves the highest accuracy, reaching 99.0% at $K = 10$, surpassing $N = 64$.
- Increasing N to 64 results in slightly lower accuracy at $K = 10$, achieving 98.2%.
- Smaller input sizes ($N = 10$ achieves 90.6% and $N = 5$) lead to noticeable performance drops, with $N = 5$ stabilizing at 83.9% from $K = 5$ on-wards while still maintaining relevant accuracy levels.

Summary of Results Obtained by Varying Input Size in Rosslyn-T006

- $N = 64$ achieves the best accuracy, reaching 97.2% at $K = 10$.
- Reducing N to 32 slightly lowers accuracy to 94.1% at $K = 10$.
- For $N = 10$, the model achieves 77.16% accuracy at $K = 10$. While not as strong as higher input sizes, this result is still reasonable given the reduction in input data.
- $N = 5$ results in the lowest performance, stabilizing at 57.9% for $K \geq 5$.

5.3.3 Performance Under LOS and NLOS Conditions

Figure 5.6 compares the Top-K accuracy of the proposed regression model under Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) conditions. As expected, the accuracy under LOS conditions outperforms NLOS due to the clearer and more direct beam paths available in LOS scenarios. However, it is notable that the NLOS accuracy remains relatively high despite the increased complexity and unpredictability of the environment.

In T005, the LOS condition achieves a peak accuracy of 98.2% at $K = 10$, while the NLOS condition reaches 88.2%. The gap of approximately 10% demonstrates the impact of obstructed paths on the model's performance but also highlights its robustness in less ideal conditions.

In T006, the gap is slightly wider, with LOS achieving 97.2% and NLOS reaching 82.2% at $K = 10$. This is consistent with the more challenging characteristics of T006, as reflected in its higher MAFD.

While the NLOS results are lower across both datasets, they are still promising, particularly given the inherent difficulty of accurately predicting beam paths in such conditions. Future work could explore the effects of training the model with a more balanced dataset containing additional NLOS samples to further enhance generalization in complex environments.

Overall, these results emphasize the strengths of the proposed RSRP regression model, particularly in its resilience to dataset complexities and its potential for reducing overhead while maintaining high accuracy in beam-tracking tasks.

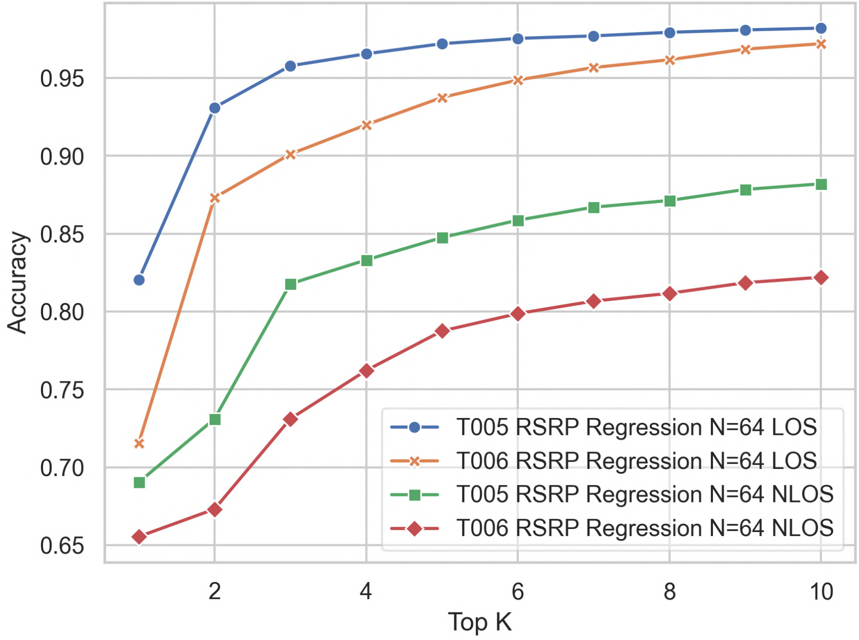

Figure 5.6: Line-of-sight and Non-line-of-sight beam tracking Top-*K* accuracy.

5.3.4 Throughput Ratio

Figure 5.7 illustrates the throughput ratio achieved for Top-*1* accuracy across different beam-tracking methods in two distinct scenarios: Marselha (T005) and Rosslyn (T006). The throughput ratio is a key performance metric that quantifies the fraction of successful beam alignments relative to an ideal baseline, offering insight into the effectiveness of the proposed models in maintaining a stable communication link.

Among the classification models, only Zhao et al. maintained a good generalization capacity across both datasets. The "RSRP Regression" methods, even when reducing the size of Set A by half, achieved comparable and high throughput in both scenarios. Specifically, with $N = 64$, the throughput ratios achieved 0.8739 in Marselha (T005) and 0.8449 in Rosslyn (T006), while for $N = 32$, the throughput ratios were 0.8702 and 0.7663 in Marselha (T005) and Rosslyn (T006), respectively.

These results highlight notable differences in performance between the two datasets, reflecting a similar pattern to that observed in the accuracy results.

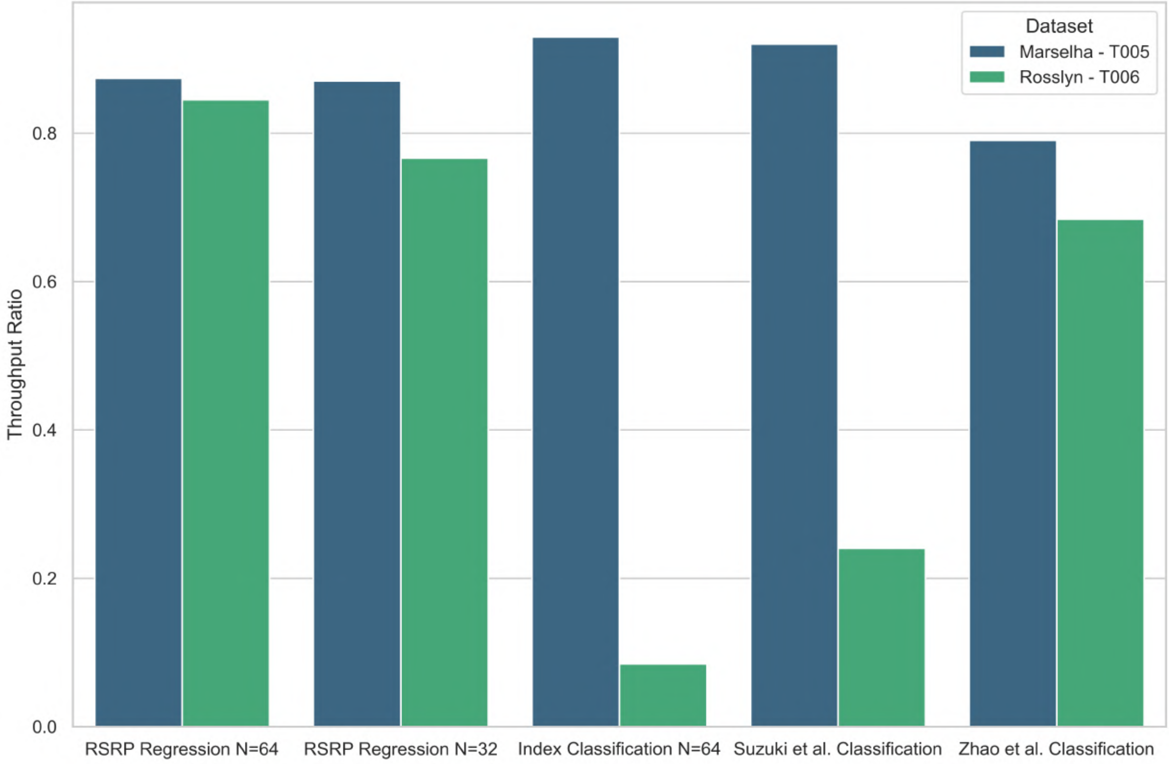

Figure 5.7: Throughput Ratio for Top-*1* accuracy.

5.3.5 Advantages of Reducing the Number of Measurements

Reducing the number of measurements performed in beam tracking not only reduces network overhead but also offers significant advantages in terms of resource efficiency and energy consumption. By evaluating the performance of the proposed model with different measurement setups, we can quantify these benefits while maintaining satisfactory accuracy levels.

In the full measurement case, each episode comprises 40 scenes, and each beam is measured in all 40 scenes, resulting in a total of $40 \times 64 = 2560$ measurements per episode. However, by adopting the proposed methodology, this number is significantly reduced. For example:

- Considering the input using both measurements and predictions this became a reduced measurement scenario, only 14 scenes are used per beam, leading to $14 \times 64 = 896$ measurements—an overall reduction of approximately 65%.
- Further reductions can be achieved by decreasing the input size N. For instance, with $N = 32$, the total number of measurements drops to $14 \times 32 = 448$, representing an 82.5% reduction compared to the full measurement case.

Given that each episode spans 3.12 seconds, the reduced measurement setup also decreases the time required for data acquisition. This reduction is particularly advantageous in scenarios where network resources are limited, as it allows the system more time to focus on transmitting user data, thereby improving overall network efficiency.

Moreover, the proposed reduction methodology preserves the model's performance, as shown in earlier analyses. Even with fewer measurements and smaller input sizes, the regression-based approach achieves competitive accuracy levels, reinforcing the practicality of the method. This trade-off between efficiency and performance highlights the potential for deploying beam-tracking solutions in real-world applications where resource constraints are critical.

Chapter 6

Conclusions

This research presented a first exploration of deep learning-based beam tracking for 5G and 6G mmWave communication, specifically targeting V2I scenarios. By leveraging regression models to predict RSRP values, the proposed methodology demonstrated consistent and robust performance across the investigated environmental conditions, including the more challenging NLOS propagation scenarios.

The results highlighted the model's capacity to achieve high accuracy while reducing the input size and the number of measurements required per episode. These optimizations significantly enhance the system's computational efficiency and decrease network overhead, freeing resources for actual data transmission. Unlike most approaches, which focus on beam selection, this work addresses the more challenging problem of beam tracking. The results represent initial steps in this direction, evaluated on two distinct scenarios, and provide promising evidence of the method's applicability.

In the Marselha-T005 dataset, regression models displayed exceptional performance, achieving accuracy levels as high as 98.2$\%$ for $K = 10$. Classification models excelled further in this specific scenario, with the RNN index classification model achieving nearly 99.9$\%$ accuracy at the same K value. The use of multimodal data, such as LIDAR, proved beneficial for the CNN classification model, which reached up to 99.4$\%$ accuracy. However, For the RNN models, reducing the input size revealed trade-offs between computational efficiency and accuracy, as models with smaller inputs like $N = 5$ or $N = 10$ displayed noticeable performance degradation.

In the Rosslyn-T006 dataset, the RSRP regression models again exhibited robustness, maintaining high accuracy levels despite the increased dataset complexity, which introduced a higher degree of unpredictability in beam index patterns. The RSRP regression model achieved a maximum accuracy of 97.2% for $K = 10$. However, classification models struggled in this scenario, with the RNN classification model reaching only 34.3% accuracy at $K = 10$, likely due to the higher MAFD present in the dataset. These results demonstrate the adaptability of the regression approach to diverse and dynamic scenarios, making it a strong candidates for more researches seeking real-world deployments where generalization and resilience to environmental variability are critical.

Further analyses of varying input sizes reinforced the effectiveness of the regression-based models. For Marselha-T005, an input size of $N = 32$ surprisingly achieved higher accuracy than the largest input size $N = 64$, suggesting a potential for optimizing accuracy while reducing computational overhead. For Rosslyn-T006, the highest accuracy was observed with $N = 64$, while $N = 32$ provided comparable results with only minimal accuracy loss, highlighting its utility in resource-constrained scenarios. Models with $N = 5$ or $N = 10$ exhibited significant accuracy reductions, although they still offered meaningful insights for highly constrained environments.

The proposed beam-tracking model demonstrated resilience across both LOS and NLOS conditions, consistently achieving high accuracy levels. In the Marselha-T005 dataset, the LOS condition reached 98.2% accuracy at $K = 10$, while the NLOS condition achieved 88.2%. The Rosslyn-T006 dataset presented greater challenges, with LOS and NLOS conditions achieving 97.2% and 82.2% accuracy, respectively. For greater reliability of these results, it is necessary to evaluate them in more scenarios. However, these initial results underscore the model's robustness and adaptability to various propagation environments.

Throughput ratio results further validated the proposed method's effectiveness, with the "RSRP Regression" models demonstrating stable throughput ratios in both Marselha and Rosslyn scenarios.

Reducing the number of measurements proved to be a crucial factor in improving network efficiency, particularly in terms of communication dynamics. This approach reduces network overhead by up to 65%, allowing more efficient use of resources by focusing on essential measurements that align with the dynamics of beam tracking, such as beam sweep and channel estimation. While there is a relationship between the reduction in measurements and model accuracy, the primary benefit lies in optimizing the overall system performance, freeing up network resources for data transmission. These optimizations reflect a crucial trade-off between between performance and efficiency.

The insights gained from this research contribute to advancing the state-of-the-art in beam management for next-generation networks and provide a foundation for future investigations.

6.1 Future Works

For future work, several avenues can be further explored to enhance the methodology and broaden its applicability. One promising direction is leveraging transfer learning techniques, which could allow the model to adapt to new scenarios or environments with limited data, significantly reducing the need for retraining and enabling faster deployment in diverse conditions. Additionally, the extension of this work to V2V communication scenarios presents an exciting opportunity to evaluate the method's robustness and adaptability in a wider range of vehicular networks.

Another key avenue lies in evaluate the methodology in more scenarios, since this work was limited to two datasets only. Also is necessary focus in optimizing the training process using datasets with a more balanced LOS and NLOS distribution to assess model performance comprehensively in various conditions. Incorporating real-world measurements for validation and extending the methodology to handle dynamic multi-user scenarios could also broaden its applicability.

Finally, introducing mechanisms to correct beam-tracking failures in real-time may further improve system reliability, particularly in highly dynamic environments.

6.2 Published Articles

This master's research culminated in several contributions to the academic community, resulting in the publication of the following articles:

- Oliveira, A., Suzuki, D., Bastos, S., Correa, I., & Klautau, A. (2024, November). Machine Learning-Based mmWave MIMO Beam Tracking in V2I Scenarios: Algorithms and Datasets. In 2024 IEEE Latin-American Conference on Communications (LATINCOM). IEEE.

- Klautau, A., Correa, I., Bastos, F., Nascimento, I., Borges, J., Oliveira, A., ... & Lins, S. (2023). Integrated simulation of deep learning, computer vision and physical layer of UAV and ground vehicle networks. In Deep Learning and Its Applications for Vehicle Networks (pp. 321-342). CRC Press.
- Bastos, S., Oliveira, A., Suzuki, D., Gonçalves, L., Sousa, I., & Klautau, A. (2023). Generation of 5G/6G Wireless Channels Using Raymobtime with Sionna's Ray-Tracing. XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais.
- Brasil, C., Reis, R., Oliveira, A., Braun, C., Damasceno, L., Correa, I., & Klautau, A. (2023). Automatic Generation of Images Using Unreal Engine for Supervised Learning. XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais.
- Suzuki, D., Oliveira, A., Gonçalves, L., Correa, I., Klautau, A., Lins, S., & Batista, P. (2022, November). Ray-Tracing MIMO Channel Dataset for Machine Learning Applied to V2V Communication. In 2022 IEEE Latin-American Conference on Communications (LATINCOM) (pp. 1-6). IEEE.
- Correa, I., Oliveira, A., Du, B., Nahum, C., Kobuchi, D., Bastos, F., ... & Klautau, A. (2022). Simultaneous beam selection and users scheduling evaluation in a virtual world with reinforcement learning. ITU Journal on Future and Evolving Technologies, 3(2), 202-213.

Bibliography

[1] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, “6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions,” *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[2] W. Roh, J. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, “Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results,” *IEEE Communications Magazine*, vol. 52, pp. 106–113, 2014.

[3] J. Huang, C.-X. Wang, H. Chang, J. Sun, and X. Gao, “Multi-Frequency Multi-Scenario Millimeter Wave MIMO Channel Measurements and Modeling for B5G Wireless Communication Systems,” *IEEE Journal on Selected Areas in Communications*, vol. 38, pp. 2010–2025, 2020.

[4] E. Bjornson, L. Van der Perre, S. Buzzi, and E. G. Larsson, “Massive MIMO in Sub-6 GHz and mmWave: Physical, Practical, and Use-Case Differences,” *IEEE Wireless Communications*, vol. 26, no. 2, pp. 100–108, 2019.

[5] W. Yi, W. Zhiqing, and F. Zhiyong, “Beam training and tracking in mmwave communication: A survey,” *China Communications*, 2024.

[6] Q. Xue, C. Ji, S. Ma, J. Guo, Y. Xu, Q. Chen, and W. Zhang, “A survey of beam management for mmWave and THz communications towards 6G,” *IEEE Communications Surveys & Tutorials*, 2024.

[7] 3GPP, “Feature lead summary #3 evaluation of AI/ML for beam management,” Meeting Document R1-2306199, 3GPP TSG RAN WG1 Meeting #112b-e, e-Meeting, Apr. 2023.

[8] B. A. Asi and F. E. Mohmood, “Beam tracking channel for millimeter-wave communication system using least mean square algorithm,” *Al-Rafidain Engineering Journal (AREJ)*,

vol. 26, no. 2, pp. 118–123, 2021.

[9] S. Shaham, M. Kokshoorn, M. Ding, Z. Lin, and M. Shirvanimoghaddam, "Extended kalman filter beam tracking for millimeter wave vehicular communications," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[10] S. H. Lim, S. Kim, B. Shim, and J. W. Choi, "Deep learning-based beam tracking for millimeter-wave communications under mobility," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7458–7469, 2021.

[11] M. Dias, A. Klautau, N. González-Prelcic, and R. W. Heath, "Position and LIDAR-aided mmWave beam selection using deep learning," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[12] D. Suzuki, A. Oliveira, L. Gonçalves, I. Correa, A. Klautau, S. Lins, and P. Batista, "Ray-Tracing MIMO Channel Dataset for Machine Learning Applied to V2V Communication," in *2022 IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2022, pp. 1–6.

[13] I. Correa, A. Oliveira, B. Du, C. Nahum, D. Kobuchi, F. Bastos, H. Ohzeki, J. Borges, M. Mehta, P. Batista *et al.*, "Simultaneous beam selection and users scheduling evaluation in a virtual world with reinforcement learning," *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 2, pp. 202–213, 2022.

[14] Y. Zhao, X. Zhang, X. Gao, K. Yang, Z. Xiong, and Z. Han, "LSTM-Based Predictive mmWave Beam Tracking via Sub-6 GHz Channels for V2I Communications," *IEEE Transactions on Communications*, 2024.

[15] W. Zhong, L. Zhang, H. Jin, X. Liu, Q. Zhu, Y. He, F. Ali, Z. Lin, K. Mao, and T. S. Durrani, "Image-Based Beam Tracking With Deep Learning for mmWave V2I Communication Systems," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[16] A. Oliveira, D. Suzuki, S. Bastos, I. Correa, and A. Klautau, "Machine learning-based mmwave mimo beam tracking in v2i scenarios: Algorithms and datasets," in *2024 IEEE Latin-American Conference on Communications (LATINCOM)*, 2024, pp. 1–5.

[17] J. R. Hampton, *Introduction to MIMO communications*. Cambridge university press,

2013.

[18] *IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 5: Enhancements for High Efficiency WLAN*, IEEE Std. IEEE 802.11ax-2021, 2021. [Online]. Available: https://standards.ieee.org/standard/802_11ax-2021.html

[19] *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Broadband Wireless Access Systems*, IEEE Std. IEEE 802.16e-2005, IEEE 802.16m-2011, 2005/2011. [Online]. Available: https://standards.ieee.org/standard/802_16-2011.html

[20] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, 2016.

[21] B. Ning, Z. Tian, Z. Chen, C. Han, J. Yuan, and S. Li, "Prospective beamforming technologies for ultra-massive MIMO in terahertz communications: A tutorial," *arXiv preprint arXiv:2107.03032*, 2021.

[22] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "Networks and devices for the 5G era," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 90–96, 2014.

[23] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, 2018.

[24] R. J. Mailloux, *Phased array antenna handbook*. Artech house, 2017.

[25] J. Wang, Z. Lan, C.-S. Sum, C.-W. Pyo, J. Gao, T. Baykas, A. Rahman, R. Funada, F. Kojima, I. Lakkis *et al.*, "Beamforming codebook design and performance evaluation for 60GHz wideband WPANs," in *2009 IEEE 70th Vehicular Technology Conference Fall*. IEEE, 2009, pp. 1–6.

[26] P. J. Bevelacqua, *Antenna arrays: Performance limits and geometry optimization*. Arizona state university, 2008.

[27] S. V. Hum, "Radio and Microwave Wireless Systems," 2017, accessed: 12/10/2024. [Online]. Available: http://www.waves.utoronto.ca/prof/svhum/ece422/notes/15-arrays2.

pdf

[28] E. Ali, M. Ismail, R. Nordin, and N. F. Abdulah, "Beamforming techniques for massive MIMO systems in 5G: overview, classification, and trends for future research," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, pp. 753–772, 2017.

[29] D. d. S. Brilhante, J. C. Manjarres, R. Moreira, L. de Oliveira Veiga, J. F. de Rezende, F. Müller, A. Klautau, L. Leonel Mendes, and F. A. P. de Figueiredo, "A literature survey on AI-aided beamforming and beam management for 5G and 6G systems," *Sensors*, vol. 23, no. 9, p. 4359, 2023.

[30] A. Knopp, R. T. Schwarz, C. A. Hofmann, M. Chouayakh, and B. Lankl, "Measurements on the impact of sparse multipath components on the LOS MIMO channel capacity," in *2007 4th International Symposium on Wireless Communication Systems*. IEEE, 2007, pp. 55–60.

[31] F. Fuschini, E. M. Vitucci, M. Barbiroli, G. Falciasecca, and V. Degli-Esposti, "Ray tracing propagation modeling for future small-cell and indoor applications: A review of current techniques," *Radio Science*, vol. 50, no. 6, pp. 469–485, 2015.

[32] A. Paulraj, D. Gore, R. Nabar, and H. Bolcskei, "An overview of MIMO communications - a key to gigabit wireless," *Proceedings of the IEEE*, vol. 92, no. 2, pp. 198–218, 2004.

[33] G. Stratidakis, G. D. Ntouni, A.-A. A. Boulogeorgos, D. Kritharidis, and A. Alexiou, "A low-overhead hierarchical beam-tracking algorithm for THz wireless systems," in *2020 European Conference on Networks and Communications (EuCNC)*. IEEE, 2020, pp. 74–78.

[34] S. He, J. Wang, Y. Huang, B. Ottersten, and W. Hong, "Codebook-based hybrid precoding for millimeter wave multiuser systems," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5289–5304, 2017.

[35] 3GPP, "FL summary #5 for AI/ML in beam management," Meeting Document R1-2407554, 3GPP TSG RAN WG1 Meeting #118, Maastricht, NL, Apr. 2023.

[36] S. Hochreiter, "Long Short-term Memory," *Neural Computation MIT-Press*, 1997.

[37] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO data for machine learning: Application to beam-selection using deep learning," in *2018 Information Theory and Applications Workshop (ITA)*, 2018, pp. 1–9.

[38] A. Klautau, A. Oliveira, I. Trindade, and W. Alves, “Generating MIMO channels for 6G virtual worlds using ray-tracing simulations,” in *IEEE Statistical Signal Processing Workshop*. IEEE, 2021, pp. 595–599.

[39] Remcom, “Wireless InSite: EM Propagation Software,” https://www.remcom.com/wireless-insite-em-propagation-software, accessed: 2024-04-20.

[40] M. B. Mashhadi, M. Jankowski, T.-Y. Tung, S. Kobus, and D. Gündüz, “Federated mmWave beam selection utilizing LIDAR data,” *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2269–2273, 2021.

[41] A. Nascimento, W. Frazão, A. Oliveira, D. Gomes, and A. Klautau, “Multimodal dataset for machine learning applied to telecommunications,” in *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBRT)*, 2020.

[42] M. Alrabeiah, J. Booth, A. Hredzak, and A. Alkhateeb, “Viwi vision-aided mmWave beam tracking: Dataset, task, and baseline solutions,” *arXiv preprint arXiv:2002.02445*, 2020.

[43] 3GPP, “Technical Specification Group Radio Access Network; Study on channel model for frequencies from 0.5 to 100 GHz (Release 16),” 3rd Generation Partnership Project (3GPP), Tech. Rep. TR 38.901, May 2020, version 16.1.0. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.901/38901-g10.zip

[44] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The next generation wireless access technology*. Academic Press, 2020.

[45] S. Targ, D. Almeida, and K. Lyman, “Resnet in resnet: Generalizing residual architectures,” *arXiv preprint arXiv:1603.08029*, 2016.

[46] B. Salehi, G. Reus-Muns, D. Roy, Z. Wang, T. Jian, J. Dy, S. Ioannidis, and K. Chowdhury, “Deep learning on multimodal sensor data at the wireless edge for vehicular network,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7639–7655, 2022.