Deep Learning for Large-Scale MIMO:

An Intelligent Wireless Communications Approach

by

Muhammad Alrabeiah

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved July 2021 by the
Graduate Supervisory Committee:

Ahmed Alkhateeb, Chair
Pavan Turaga
Gautam Dasarathy
Cihan Tepedelenlioglu

ARIZONA STATE UNIVERSITY

August 2021

ABSTRACT

Recent years have seen machine learning makes growing presence in several areas in wireless communications, and specifically in large-scale Multiple-Input Multiple-Output (MIMO) systems. This comes as a result of its ability to offer innovative solutions to some of the most daunting problems that haunt current and future large-scale MIMO systems, such as downlink channel-training and sensitivity to line-of-sight (LOS) blockages to name two examples. Machine learning, in general, provides wireless systems with data-driven capabilities, with which they could realize much needed agility for decision-making and adaptability to their surroundings. Bearing the potential of machine learning in mind, this dissertation takes a close look at what deep learning can bring to the table of large-scale MIMO systems. It proposes three novel frameworks based on deep learning that tackle challenges rooted in the need to acquire channel state information. Framework 1, namely deterministic channel prediction, recognizes that some channels are easier to acquire than others (e.g., uplink are easier to acquire than downlink), and, as such, it learns a function that predicts some channels (target channels) from others (observed channels). Framework 2, namely statistical channel prediction, aims to do the same thing as Framework 1, but it takes a more statistical approach; it learns a large-scale statistic for target channels (i.e., per-user channel covariance) from observed channels. Differently from frameworks 1 and 2, framework 3, namely vision-aided wireless communications, presents an unorthodox perspective on dealing with large-scale MIMO challenges specific to high-frequency communications. It relies on the fact that high-frequency communications are reliant on LOS much like computer vision. Therefore, it recognizes that parallel and utilizes multimodal deep learning to address LOS-related challenges, such as downlink beam training and LOS-link blockages. All three frameworks are studied and discussed using datasets representing various large-scale MIMO settings. Overall, they show promising results that cement the value of machine learning, especially deep learning, to large-scale MIMO systems.

DEDICATION

To my mother, Norah, and my wife, Hafida

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

The major leaps Machine Learning (ML) has been taking in the past two decades have transformed the landscape of Artificial Intelligence (AI). Machines are now capable of performing tasks that require a heightened sense of intelligence, and they do so in a way that in some cases is on par with human ability—[1] and [2] are two examples of that. These leaps in ML have advanced the state-of-the-art on a variety of tasks across the fields of computer vision and natural language processing (major and classical fields for AI research). As a result, one could argue that ML have left a defining mark on both fields, a mark that casts ML in the role of the enabler to many future technologies that rely on computer vision and/or natural language processing. Good examples for that could be seen in technologies like autonomous vehicles [3], and conversational artificial intelligence [4]. The former leverages the breakthroughs ML has made in computer vision (e.g., image classification [5][6][7], and object detection [8][9]) to achieve scene understanding, which is, in turn, vital for a self-driving vehicle to understand its surrounding. The latter, on the other hand, taps into the advances in natural language processing with ML (e.g., speech recognition [10] [11], machine translation [12, 13]), which enable a machine to listen, comprehend, and respond to spoken or written language.

The success ML has enjoyed in computer vision and natural language processing has recently seeped into the field of wireless communications, a field where ML does not traditionally have obvious presence. A major factor in that is the unprecedented challenges facing modern and future wireless communication systems and networks [14]. Those challenges, to a large extent, stem from a growing need for agility, reliability, and scalability that allow wireless systems and networks to provide communications at high spectral efficiency, low latency, and high reliability [15, 16]. ML (and AI in general) offers wireless communications a promising way to address those challenges head-on. This is evident in recent work that utilizes ML to tackle variety of problems in wireless communications. Some good examples could found in [17–21]. Collectively, that work has not only consolidated the value of ML to wireless communications, but it has also brought to light an interesting duality; as ML holds tremendous potential in terms of advancing wireless communications, it, as well, stands to be advanced by the types of wireless challenges it addresses. Those challenges are not always similar to what ML has long encountered in traditional fields such as computer vision and natural language processing, and, hence, they could be the seed for novel ML research directions.

This dissertation is built around that duality between ML and wireless communications. By focusing on challenges in large-scale Multiple Inputs Multiple Outputs (MIMO) communications [16, 22], it proposes novel and innovative approaches and frameworks to wireless communications as well as ML. The following three sections aim to set the stage for the discussion on those approaches and frameworks. They provide a brief account on what ML paradigm will be adopted, discuss major large-scale MIMO challenges that will be addressed, and, finally, present the thesis statement of the dissertation.

1.1 Deep Learning

ML has recently achieved major milestones in terms of enabling machines to perform certain intelligent tasks on levels close to or surpassing those of humans [1, 2, 23]. This could be, in large part, attributed to the advances in developing learning algorithms with deep architectures [24]. Such algorithms have not only helped achieve those milestones, but they have also ushered in the era of *deep learning*. Prior to the seminal work in [25–27], the ML community was focused on developing well-engineered *shallow learning* architectures [24], e.g., Support Vector Machines (SVM) and Gaussian Mixture Model (GMM) to

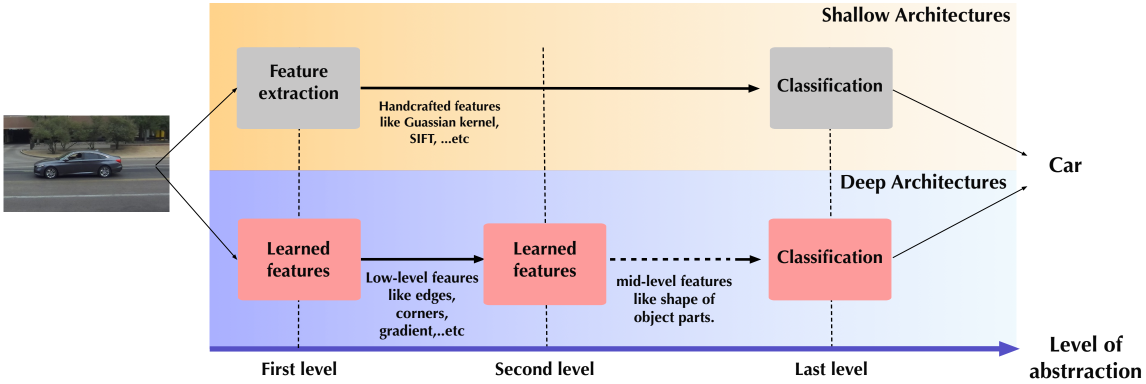


Figure 1.1: An Illustration of Shallow and Deep Architectures from Representation Perspective. Shallow Architectures Extracts Handcrafted Features from the Input and Utilize Them to Predict an Abstract Concept. Deep Architectures, on the Other Hand, Gradually Increase the Level of Abstraction the Features Represent Before They Predict the Final Abstract Concept.

name two. That focus started shifting when several successful approaches had been proposed in [25–27] to train Deep Neural Network (DNN) architectures. They encouraged the study and investigation of deep architecture in variety of problems. As a consequence, AlexNet [28]—a deep Convolutional Neural Network (CNN)—managed to score a record classification error rate on the large-scale image classification challenge ImageNet [29], and this could be seen as the real inflection point at which the ML research pivoted from shallow to deep learning.

1.1.1 Deep Versus Shallow Learning

Learning with deep architecture is fundamentally different form that of shallow ones [30]. Figure 1.1 presents a schematic illustration of that using a classification example from computer vision. Given a classification task with an input sensory data like as RGB images and a target abstract concept like the class “car”, deep learning attempts to breakdown the relation between the input and the target into multi-level hierarchical representation. Each layer of the deep architecture attempts to learn a level of representation from its predecessor. As Figure 1.1 shows, the first level may extract close-to-perception features like edges and corners from the input image. Those features get gradually transformed into new features with a higher level of abstraction as they go through higher layers. For instance, the second layer may transform the edges and corners extracted by the first layer into features describing certain parts of the object of interest, the car. All those transformations are learned throughout the training process using large amounts of data points (examples). In contrast, shallow architectures follow a different approach. They are usually composed of one (or in some cases two) layer(s) of *engineered* feature extraction[1]. An engineered extraction layer is designed such that it identifies specific patterns in the input using some well-designed rules (e.g., like Gaussian kernels [31] and Scale-Invariant Feature Transform (SIFT) [32]). The output here is a set of *handcrafted* features, which are used in conjunction with a powerful classifier to predict the abstract concept.

The layered hierarchical learning paradigm deep architectures follow poses some advantages compared to learning with shallow architectures. This has been argued in many theoretical and empirical studies in the literature [25, 26, 30, 33, 34]. One of the most elegant and intuitive, albeit informal, of those arguments is that presented in [30]. It is stands on three main pillars:

- **Pillar 1:** The smoothness prior [30, 35, 36], common in shallow learning, does not hold for many learning tasks that have an underlying highly-varying function[2].

[1] It is worth mentioning here that this is note a hard fact but a loose description of shallow architectures. For instance, a neural network with one hidden layer is a shallow architecture where the hidden layer features are not quite engineered, but learned via training.

[2] A simple definition to highly-varying functions such as that presented in [30] is adopted here: "a function is highly-varying when a piecewise approximation (e.g., piecewise-constant or piecewise-linear) of that function would require a large number of pieces."

- **Pillar 2:** An effective way to learn highly-varying functions requires the learning of a feature representation where smoothness is maintained between the features and the target.

- **Pillar 3:** Functions that could be represented with depth-k learning algorithm (or a DNN) require an exponential number of computation units (e.g., neurons in DNNs) to be represented with depth-(k-1) algorithms [30, 33].

The first pillar simply states that given some definition of proximity (e.g., Euclidean distance) a function governing a real learning task $f_{\text{task}}(.)$ does not satisfy $f_{\text{task}}(x_1) \approx f_{\text{task}}(x_2)$ when two points x_1 and x_2 are in the proximity of one another ($x_1 \approx x_2$). This suggests that in order to learn $f_{\text{task}}(.)$ using a shallow architecture, one might need at least as many training examples as there are variations (regions where $x_1 \approx x_2$ and $f_{\text{task}}(x_1) \approx f_{\text{task}}(x_2)$ holds) in $f_{\text{task}}(.)$, which could be quite a lot for highly-varying functions. However, an interesting approach around the smoothness issue is presented in **Pillar 2**. It suggests transforming the input into a space where a sense of smoothness could be attained, i.e., identifying $\phi(.)$ such that when $\phi(x_1) \approx \phi(x_2)$, then $g(\phi(x_1)) \approx g(\phi(x_2))$ where $f_{\text{task}} = (g \circ \phi)(.)$. Such transformation is not quite straightforward to engineer [30], and, hence, finding the appropriate transformation could be part of the learning process. This last statement takes the discussion into a crossroads; if a transformation needs to be learned, should it be learned using a shallow or deep architecture? The final and third pillar **Pillar 3** provides an interesting answer to that. On the surface, learning $\phi(.)$ could be done by both architectures, yet deep learning presents a significantly better approach to do that in terms of computational resources. For example, if the question is whether to go with a shallow single-hidden-layer or deep multi-layer neural network, the answer is a deep network because, as per **Pillar 3**, it will require less number of neurons (less computational units).

1.2 Large-Scale MIMO

An intrinsic and defining feature of modern and future wireless communication networks and systems (5G and 6G) is the heterogeneity of their services [14, 15]. This is a direct consequence of the diversity of the applications those networks and systems support, ranging from Virtual Reality (VR) and Augmented Reality (AR) to Autonomous vehicles and the Internet of Things (IoT). The modern view of such diverse pool of services categorize them into three broad categories: (i) Enhanced Mobile Broadband (eMBB), (ii) Machine-Type Communications (MTC), and (iii) Ultra-Reliable Low-Latency Communications (URLLC) [15, 37]. This view is based on three fundamental wireless requirements that are spectral efficiency, reliability, and latency, the mixing of which with varying degrees of impact produces those categories. Meeting the requirements of all three categories of services is a major task modern and future wireless networks and systems has to perform, and large-scale MIMO communication is one key player in doing so [16, 22, 38].

The fundamental idea behind large-scale MIMO communications is the scaling-up of the number of antennas at both transmitters and receivers. This allows wireless networks to harvest the gains of spatial multiplexing and beamforming [16, 39–41] across a wide range of frequency bands, from sub-6 GHz to Millimeter Wave (mmWave) and Sub-Terahertz (sTHz). Both gains have direct impact on all three service categories, albeit with variable degrees. The major beneficiary of large-scale MIMO is eMBB services; applications in that category are majorly rate-centric (or informally rate hungry) [14], and large-scale MIMO promises significant boosts to spectral efficiency. Hence, eMBB would get the most "bang for the buck." The next beneficiary of large-scale MIMO is the MTC category of services. Applications in that category are characterized by the need for massive and reliable connectivity. Large-scale MIMO helps meet that need by expanding the number of multiplexing dimensions to incorporate spatial multiplexing, which means the wireless network

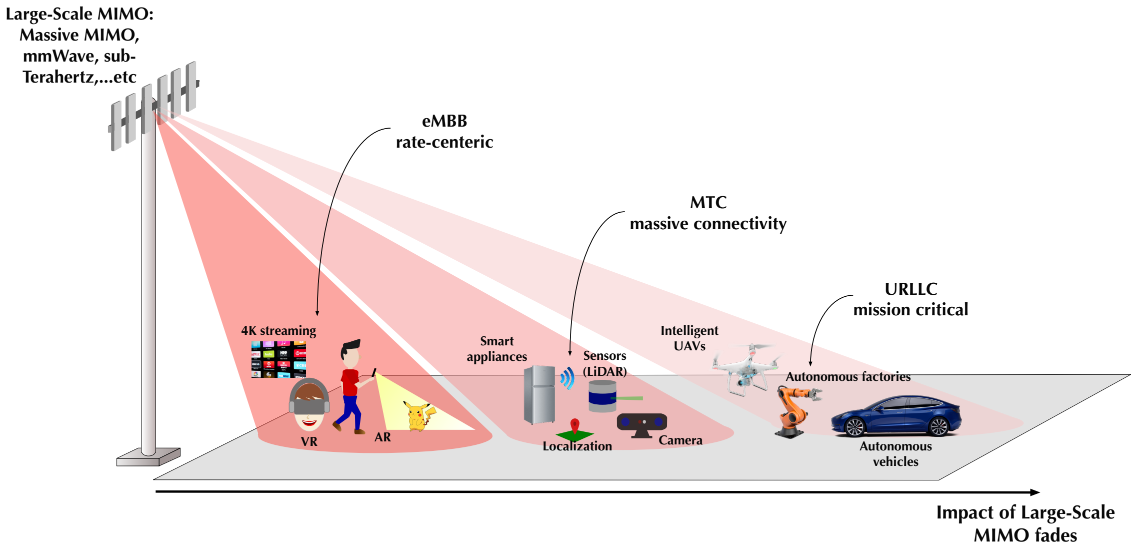


Figure 1.2: The Impact of Large-scale MIMO on Different Modern and Future Wireless Services. The Mostly Impacted Is eMBB as It Immediately Relies on Spectral Efficiency. MTC Benefits from the Spatial Multiplexing Gain Provided by Large-scale MIMO, and URLLC Benefits from the Short Wireless Frames Allowed by the Improved Spectral Efficiency.

has more resources to service the large number of machines. The final beneficiary from large-scale MIMO is URLLC. Applications in that category require, as the name suggests, ultra reliability and low latency. This could be, to some extent of course, met by large-scale MIMO through the increased spectral efficiency and additional spatial multiplexing; the former allow the transmission of shorter wireless frames while the latter offers the wireless network a wealth of radio resources for rapid allocation. The impact of large-scale MIMO on all three categories is summarized in Figure 1.2.

1.2.1 Challenges to Large-Scale MIMO

In the shadows of the beautiful facade that is large-scale MIMO, however, lurk a couple of critical challenges [16, 40] that are, in some sense, at odds with its advantages. Those challenges could be broadly clustered into two categories: (i) **channel-related challenges**, e.g., downlink channel training in FDD massive MIMO and downlink beam training in mmWave and sTHz MIMO; and (ii) **LOS-related challenges**, e.g., mmWave and sTHz Line-of-Sight (LOS) link blockage and user hand-off.

Big picture

The roots of channel-related challenges, in general, lie in the increasing number of antennas. Despite its benefits, scaling up the number of antennas brings about increasing burden on the physical layer of the wireless network. More specifically, whether a large-scale MIMO is operating in sub-6 GHz, mmWave, or sTHz frequency bands, the large number of antennas mandates expensive channel/beam training [18, 40, 42]—at least from a traditional signal-processing perspective. This training burden is critical to acquire wireless channel information (mostly in sub-6 GHz) or to select the best beamforming vector (mostly in mmWave and sTHz). Both spectral efficiency and reliability in wireless networks are impacted by that burden; the relatively long training could result in channel aging [43] where the estimated channel is no longer viable. The result of that aging issue is a degradation in the data-rate and also the reliability of the wireless service, especially for highly-mobile users. Furthermore, training a massive number of antennas occupies a relatively big chunk of the wireless time frame, and, as such, it hinders the response to service requests causing pervasive latency issues.

The second category of challenges revolves around the reliance on LOS communications. Large-scale MIMO systems in high-frequency bands such as mmWave and sTHz [42] rely on directive beam patterns to achieve their performance gains. Those frequency bands are rich in terms of available bandwidth, yet their signal propagation is characterized by two issues: (i) poor penetration, and (ii) high power-loss due to scattering [44]. Both issues make high-frequency large-scale MIMO systems in desperate need for directive beamforming to boost up the received Signal-to-Noise Ratio (SNR), which, subsequently, leaves those systems susceptible to LOS blockages. The reliance on LOS and susceptibility to blockages threaten the ability of large-scale MIMO to meet the stringent reliability and latency requirements of URLLC and MTC services [18, 45]; any object (especially dynamic ones) could constitute a LOS blockage that, so far, can only be combated by performing user hand-off between serving basestations (or access points). This hand-off comes with its own signaling overhead [19, 45] that could degrade both reliability and latency.

Challenges of interest

In massive and high-frequency large-scale MIMO systems, some challenges are more prominent than others based on the attention they draw from the wireless communication community. A good example could be found in FDD massive MIMO challenges, which have seen an uptick in interest recently compared to those of TDD massive MIMO [46–50]. Such unbalanced attention is not only motivated by sheer curiosity but by some practical consideration. For example, the interest in FDD MIMO over TDD is driven by a suite of practical reasons such as: spectrum regulations [46], the ecosystem of modern and legacy cellular-communications [50], and expensive calibration issues in TDD systems [48]. This dissertation recognizes that unbalanced attention, and it focuses on challenges that reflect the interest in the wireless communication community. More specifically, it identifies

1. Downlink channel training in FDD massive MIMO systems,
2. Channel feedback in TDD cell-free massive MIMO systems,
3. Downlink beam training in high-frequency large-scale MIMO systems, and
4. LOS-link blockages in high-frequency large-scale MIMO systems,

as the challenges of interest.

The choice of those four challenges is mainly motivated by practical considerations. The first two are a good embodiment of challenges rooted in the need to acquire or share channel information in sub-6 GHz massive MIMO systems [18, 47, 51]. That need scales up with the number of deployed antennas, and it makes channel training verging on the impossible [47]. On the other hand, the third and fourth challenges on the list are specific to high-frequency large-scale MIMO systems [18, 40]. Due to poor signal penetration and high power loss [16], communication in frequency bands such as mmWave and sTHz requires directive and LOS beams. Maintaining such beams is a bottleneck for high-frequency large-scale MIMO systems as wireless environments are inherently dynamic—moving transmitters, receivers, and scatterers—and beam training is commonly associated with hefty overhead [18].

1.2.2 Classical Solutions and Approaches

The challenges discussed in Section 1.2.1 are well acknowledge in the wireless communication community, and as such, much research has been devoted to dealing with them. Based on the paradigm followed to address large-scale MIMO challenges, that research branches out into one of two directions, ML-oriented and signal-processing-oriented. The latter represent the classical direction as signal processing has long been a close companion to wireless communications. On the other hand, the ML-oriented direction is more modern, for ML has only recently started gaining momentum and making impact on large-scale MIMO challenge.

Since the discussion throughout this dissertation is focused on ML and its role in large-scale MIMO, this section will provide a succinct overview of the major classical approaches proposed to tackle large-scale MIMO challenges. The overview is meant to offer a top-view of the landscape of the classical directions, highlighting the differences between approaches and their shortcomings that prompted the interest in ML. The discussion is divided into two subsections. The first briefly surveys solutions and approaches addressing challenges related to channel/beam training while the second takes a closer look at the challenge LOS-link blcokage.

Downlink channel and beam training

The approaches addressing downlink channel/beam training could be clustered into three distinct categories: (i) Covariance-based downlink channel training, (ii) Compressed downlink channel training, and (iii) Downlink channel extrapolation. The premise of each category and its shortcomings are discussed in the following three paragraphs.

Covariance-based downlink channel training recognizes that the propagation between a massive MIMO basestation and a user is characterized by local scattering around the user (one ring model [52]), which is captured by a low-rank downlink covariance [47]. Therefore, proposed algorithms that fall under this category target estimating the downlink covariance and utilizing it to reduce downlink training overhead (example [49]). Those algorithms are quite promising as they alleviate some of the channel training overhead. However, the achieved overhead reduction is not sufficient, and the reasons for that are twofold. First, the number of downlink training pilots is still large for practical massive MIMO system operation. For instance, the algorithm in [49] reduces channel training overhead in a system with 128 antennas to approximately 40 pilots instead of 128 (using per-antenna training). Second, the covariance estimation itself adds a new overhead, although not as large as that of downlink training. Referring back to [49], the proposed algorithm requires approximately 1000 uplink channels to estimate one downlink covariance.

Compressed downlink channel training relies on a global assumption that the massive MIMO channel has a sparse structure [53]. Proposed algorithms (e.g., [53–55]) utilize the sparsity to design compressed downlink pilots and uses compressive sensing to estimate the downlink channel from the users' feedbacks. Overall, those algorithms can reduce the training overhead, yet the number of pilots falls almost half-way between per-antenna channel training and covariance-based channel training. This has been demonstrated in the comparison presented in [49]. It shows that for a system of 128 antennas, approximately 60-80 pilots are required to achieve reasonable performance, and 80-100 pilots to match the performance of the covariance-based algorithm in [49].

The last category views the problem from a different lens than those of the first and second categories. It focuses on channel parameters estimation under frequency-invariant assumption [56]. It basically targets estimating the per-path parameters of the uplink channels and re-purposing them to estimate the downlink channels. The approach is very interesting, especially given that the reconstruction results are quite good when the uplink and downlink frequencies are narrowly separated. For instance, [56] shows that for a system with 16 antennas operating at 10 dB SNR, the NMSE of the reconstructed downlink channels is between 0.01 and 0.03 for an uplink-downlink separation of 10 to 20 MHz. However, this dependency on the uplink-downlink separation is one of the two main shortcomings of this category. With the move to high-frequency communications (mmWave and/or sTHz), wireless systems will be operating in multiple bands that will definitely have wide separation. This renders the extrapolation approaches in effective. The second reason is closely tied with the first. It is rooted in the assumption that channel parameters are frequency-invariant. This is approximately satisfied when the uplink-downlink separation is narrow, but it rapidly breaks down as the separation widens.

LOS-link blockage

The problem of *LOS link blockage* has long been acknowledged as a critical challenge to high-frequency large-scale MIMO networks [16, 42, 57, 58]. In those networks, the quality of service highly deteriorates with link blockages. Therefore, solutions centered around multi-connectivity are a major avenue to handle that problem [57]. For instance,

[58] proposes a multi-cell measurement reporting system to keep track of the link quality between a mmWave user and multiple basestations. All basestation in that system feed their measurements to a central unit that takes care of cell selection and scheduling. This system is further studied and tested in [57] under realistic dynamic scenarios. A slightly different look on multi-connectivity is presented in [59, 60]. In [59], the authors propose a few approaches for multi-connectivity, all of which focus on utilizing low-frequency bands (sub-6 GHz) to support the mmWave network. [60], on the other hand, develops a multi-connectivity algorithm that does not only factor in network reliability but also latency. Collectivity, the work on multi-connectivity has its promise and elegance, yet it is lacking on two important fronts. First, it is inherently wasteful in terms of resource utilization; multiple basestations schedule resources for one user as a precaution for probable LOS blockages. The other is its reactive nature; the majority of the multi-connectivity algorithms are designed to react to link blockages, not to anticipate them.

1.3 Deep Learning for Large-Scale MIMO

Resorting to deep learning to address many of the challenges facing large-scale MIMO is not only a trending approach, but it is one that has a lot of potential [18, 37, 46]. Deep learning offers a form of adaptability to wireless communications that cannot be achieved by traditional signal processing. Therefore, much research has been poured into identifying what challenges could be addressed with deep learning and what impact deep learning might have on wireless communications and vice versa. In an effort to further that research, this dissertation delves deeper into the role of deep learning in large-scale MIMO and proposes four intelligent frameworks that address many challenges at the level of the physical and network layers of modern and future wireless networks. In order to set the stage for the discussion on those frameworks and there impact on both wireless communications and ML, the following two subsections will provide a concise and high-level overview of the current literature of deep learning in large-scale MIMO and present the thesis statement of this dissertation.

1.3.1 Literature Review

Deep learning has been utilized to address variety of challenges in large-scale MIMO communications. They range from addressing classical problems like channel/beam training [18, 20, 61] and LOS/Non-LOS (NLOS) identification [62] all the way to presenting new capabilities such as large intelligent surfaces [63], proactive blockage prediction [19, 64], and proactive user hand-off [19, 45]. The result of that is a continuously growing body of research that does not only advance wireless communications but also ML. Owing to the volume of that research and its diversity in terms of topics, this subsection is structured such that it provides a high-level description of what has been done. This is aimed to lay the groundwork for more pointy literature reviews in each of the chapters of this dissertation, each of which will summarize the recent work relevant to the problem and solution presented in the chapter.

The approach of choice for the high-level literature review is based on the data-modalities used for the learning task. More specifically, the work on deep learning for large-scale MIMO is divided into two broad and sometime overlapping categories, namely learning form wireless data alone and learning from extra-sensor data. The former provides an umbrella for all deep learning solutions and approaches that are developed on data obtained form the wireless network, i.e., algorithms learning from data such as wireless channels, beamforming vectors, received power,..etc. The latter, on the other hand, represents what could be loosely referred to as the *forward-thinking* category; it groups those deep learning approaches and solutions that introduce extra sources of sensory data such as, but not limited to, LiDAR sensors, RGB cameras, Global Positioning System (GPS) to the wireless network, which could be seen as a form of infrastructural change in wireless networks. An overview of both categories is provided below:

- **Wireless Data:** The deep learning solutions and approaches in this category address variety of large-scale MIMO challenges relying mainly on the data a wireless network can provide. Good examples could be found in [18, 62, 63, 65]. They address different problems using deep learning, e.g., coverage and spectral efficiency [63], link reliability for highly-mobile users [18], LOS/NLOS link identification [62], and finally adaptive beamforming codebooks [65]. However, they all utilize data obtained from the wireless system such as sampled wireless channels, received signal strength, and user-fed SNR. A common denominator to most of the work in this category is the reliance on unimodal ML algorithms, which is reasonable since many of those data types exhibit a strong sense of statistical dependence, i.e., for a certain operating frequency, the received power is strongly dependent on the wireless channel.

- **Extra-Sensor Data:** In contrast to the previous category, this one has the deep learning solutions and approaches that require some infrastructural change like installing cameras at basestations or access points—more on this will be discussed later in this dissertation. This category is a little slimmer than the previous one in term of amount of research. The reasons could be two fold. First, studying the value of extra sensory data requires large-scale MIMO systems equipped with sensors such as RGB cameras, depth images, LiDAR point-cloud data, positioning data, ...etc, which is not readily available. The other reason is the unknown relation between the multi-sensor data and large-scale MIMO, which needs some research per se. However, initial work like, and not limited to, [66, 67] has shown some interesting results and use cases. Position data in [66] have been used to tackle the beam training in mmWave networks while depth images in [67] have been shown to be a promising tool to deal with link blockages. A growing trend in this category is the development of multimodal machine learning algorithms [19]. This could be seen as a natural consequence to equipping wireless networks with one or more new sensors. This provides extra information that may not be available using wireless equipment alone.

The above high-level literature overview may not be detailed enough to do the work on deep learning for large-scale MIMO justice; however, it should be good enough to equip the reader with enough information to digest the thesis statement of this dissertation. As mentioned earlier in this subsection, more detailed literature reviews will be presented in the coming chapters based on the addressed large-scale MIMO challenge and the proposed deep learning framework.

1.3.2 Thesis Statement

The work in this dissertation is founded on the duality between ML and wireless communications that is pointed out at the beginning of this chapter. More precisely, it aims to explore the role that ML can play in advancing wireless communications and vice versa. This is done via addressing various large-scale MIMO challenges with deep learning. The addressed challenges are closely tied to modern and future trends in the wireless communications industry (such as the 5G and 6G technologies), and they are addressed using state-of-the-art algorithms in the field of ML, especially deep learning. The main outcomes of this dissertation are three novel deep learning frameworks that tackle overlapping challenges in the physical and network layers of a large-scale MIMO wireless network. A summary visualization of those outcomes is depicted in Figure 1.3, and they are further detailed in the following few subsections.

Framework 1–Deterministic Channel Prediction:

Acquiring wireless channels (channel training) in large-scale MIMO is a challenging task that requires a relatively hefty training overhead. This framework provides an innovative

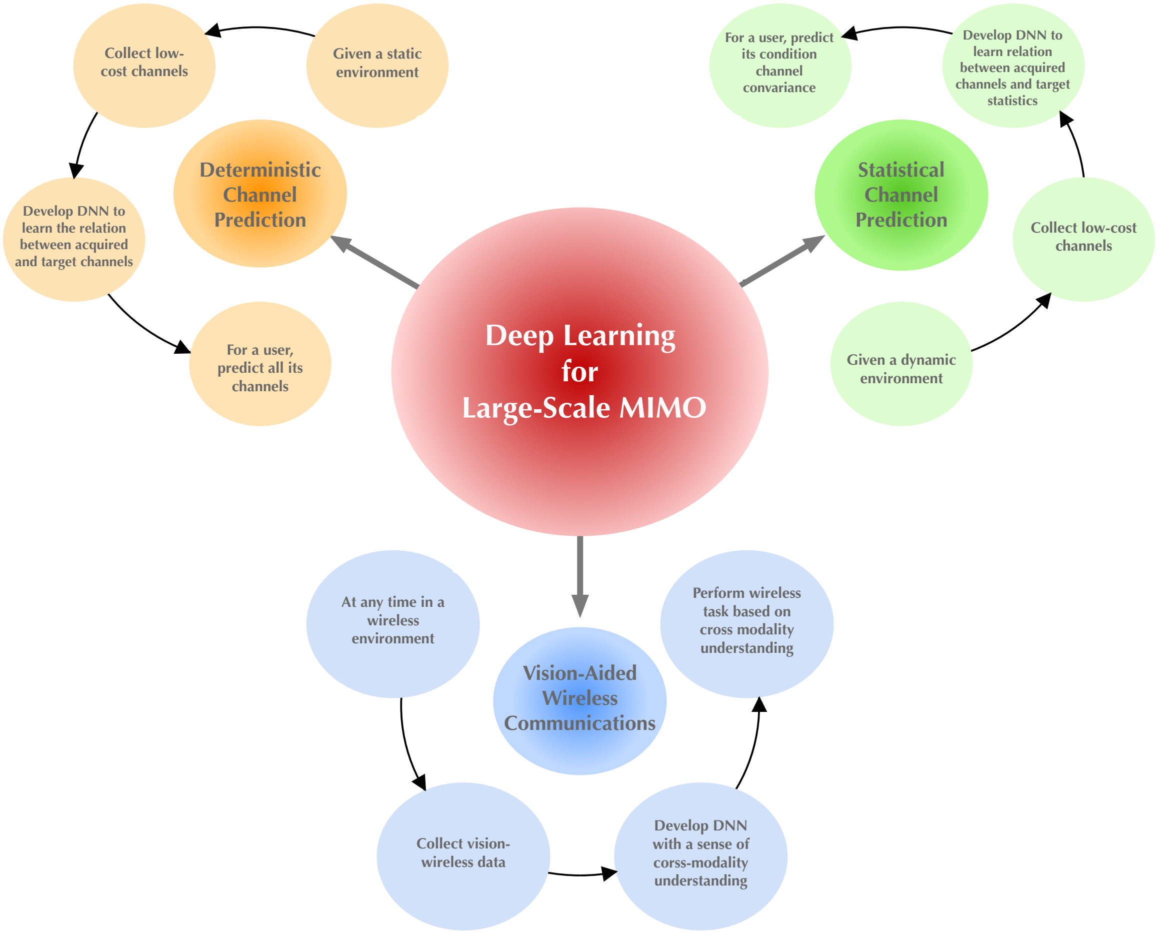

Collect low-cost channels
Given a static environment
Deterministic Channel Prediction
Develop DNN to learn the relation between acquired and target channels
For a user, predict all its channels
For a user, predict its condition channel convariance
Develop DNN to learn relation between acquired channels and target statistics
Statistical Channel Prediction
Collect low-cost channels
Given a dynamic environment
Deep Learning for Large-Scale MIMO
At any time in a wireless environment
Perform wireless task based on cross modality understanding
Vision-Aided Wireless Communications
Collect vision-wireless data
Develop DNN with a sense of corss-modality understanding

Figure 1.3: Summary of The Outcomes of This Dissertation. Three Major Frameworks Providing Novel Approaches to Handling Various Large-Scale MIMO Challenges with Different Learning Paradigms That Are Centered Around Deep Learning.

way to tackle that challenge. It starts by noting three important facts: (i) some wireless channels are easier to acquire than others, e.g., uplink channels in Frequency Division Duplex (FDD) massive MIMO systems are easier to acquire than downlink channels; (ii) the channels are a deterministic and complex function in the wireless environment geometry; and (iii) that function is time-invariant when the environment is stationary. Then, it utilizes those facts to prove the existence of a function that can *predict* some target channels (hard to acquire) given some observed channels (easy to acquire). The framework resorts to deep learning (DNNs in particular) to try and learn that prediction function. Once it is trained, the DNN is plugged into the large-scale MIMO system to reduce the channel training overhead.

Framework 2–Statistical Channel Prediction:

A glaring drawback in the deterministic channel prediction framework is in its target itself; learning to approximate a very complex vector-valued high-dimensional function is not an easy task, especially given the fact that it is also time varying. The statistical channel prediction framework is an attempt to tackle that challenge. It proposes to learn a statistical quantity that characterizes the behavior of the target channels and provide a robust way to handle channel prediction. Form a ML perspective, it focuses on learning a function that predicts a *conditional covariance* for the target channels given those that could be acquired. Learning such covariance is not only challenging from a wireless perspective, but it also poses some challenges to the learning approach and choice of DNN, as will be discussed in Chapter 4.

Framework 3–Vision-Aided Wireless Communications:

This framework introduces computer vision to wireless communications, and especially large-scale MIMO. It is founded on an interesting parallel between high-frequency (mmWave and sTHz) wireless communications and computer vision; both are LOS dependent. This parallel is very interesting as it allows wireless communications to tap into the wealth of information residing in visual data (after all, a picture is worth thousand words) as well as taking advantage of major advances in computer vision. By doing so, a large-scale MIMO wireless network could develop a sense of awareness about its surrounding that could help tackle a multitude of challenges across its physical and network layers. The way to developing that sense lies in developing cross-modality understanding, i.e., how the wireless and visual data are related to the task of interest, which, in turn, calls for multimodal learning. It is important to emphasize here that wireless and visual data are not usual companions, like audio and visual data, and, therefore, developing cross-modality understanding might as well pose an interesting challenge from a ML perspective.

The three frameworks are detailed, discussed, and empirically evaluated across the chapters of this dissertation. They are divided into two major directions based on the number of data modality they require. The first two frameworks are based on the idea that given some easily-acquired channels, a DNN architecture is developed to learn a prediction function that either predicts the target channels or predicts a covariance matrix for the target channels. Therefore, the two are, by nature, *unimodal learning* frameworks, i.e., they only need wireless channels data from which they learn and predict. On the contrary, the third framework is a *multimodal learning* framework, bimodal in the least. This is due to the fact that it leverages visual data as a secondary source of information about the wireless environment (where the large-scale MIMO network operates). This necessitates a DNN that learns from multimodal data to perform a wireless-related task. This clear difference in the learning paradigm (unimodal vs multimodal) shapes the structure of this dissertation, which will be further explained in the following subsection.

1.4 Dissertation Organization

This dissertation is organized into two major parts named unimodal learning and multimodal learning. These two parts embody the learning nature of the three frameworks. The first part on unimodal learning is a composite of three publications, each of which represents a chapter (Chapters 2, 3, and 4). They cover the deterministic channel prediction framework, namely **Framework 1**, a case study for that framework in a modern 5G wireless network, and the statistical channel prediction framework, namely **Framework 2**. The second part of this dissertation covers the Vision-Aided Wireless Communications

(ViWiComm) framework, namely **Framework 3**. It does so in two publications spanning two separate chapters, Chapters 5 and 6. Chapter 5 introduces the ViWiComm framework and shows its potential for high-frequency large-scale MIMO. This is done by considering the problems of beam and blockage prediction in simplified mmWave communication settings. Chapter 6, on the other hand, takes the discussion to a more realistic and practical wireless settings, where there are multiple possible objects responsible for the radio signal. It proposes the novel and fundamental task of transmitter identification in wireless environments. Finally, this dissertation is wrapped up in Chapter 7, where a summary of the three frameworks is presented along with some main takeaways.

As the chapters of this dissertation represent separate publications, they are structured to be self-contained and share a similar introduction section (similar in style). Before going further, the following few notes need to be pointed out:

- **Scope and contribution:** the first section in every chapter is concerned with the scope of the publication and its main contributions. It replaces the introduction of the paper and aims to set the stage for the discussion in the chapter; it introduces a high-level description of the problem, a summary of how it is tackled, and a summary of the main results. It also attempts to connect some of the aspects or ideas in the chapter with those presented in other chapters.

- **System and channel models:** It is a common practice in wireless communications to define system and channel models for under which a certain problem is addressed. Since this dissertation deals with multiple challenges in large-scale MIMO, no unified system or channel models are adopted, but each chapter will have a section where the two are defined.

- **Symbols and mathematical definitions:** Referring again to the different challenges addressed in every chapter, especially across **Parts** 1 and 2, each chapter defines its own mathematical elements, i.e., the used symbols and mathematical definitions are specific to the chapter itself.

Part I

UNIMODAL LEARNING

Chapter 2

MIMO CHANNEL MAPPING WITH DEEP LEARNING: BREAKING SPACE AND FREQUENCY BARRIERS

2.1 Scope and Contribution

Scope

This paper aims to further the efforts on utilizing machine learning, and specifically deep learning, to address the challenges of channel acquisition in large-scale MIMO; it starts by posing the following important question:

> **Q.1:** *If the channels between a user and a certain set of antennas at one frequency band is known, is it possible to use machine learning to map the known channels to those channels at a different set of antennas and at a different frequency band?*

Then, it provides an answer to that question in the form of a novel framework, dubbed the *deterministic channel-prediction framework*. This framework shows that under certain conditions, a channel-mapping function between two sets of antennas at the same or different frequency bands exists, and it shows that DNNs could be used to learn it.

Contributions

By focusing on answering Q.1, this paper presents a comprehensive treatment to the large-scale MIMO challenges stemming from the need for channel acquisition. It does so by proposing the deterministic channel prediction framework. The framework establishes an argument through which Q.1 is answered, and it taps into the learning capability of DNNs to show how the challenges could be addressed. The main contributions of the paper could be summarized as follows:

- It lays some theoretical groundwork for answering question Q.1 using the channel bijectiveness argument, which is the foundation of the deterministic prediction framework. The argument revolves around the assumption that every user position in the wireless environment has unique channels at a set of antennas and a frequency band. The argument shows how that assumption results in the existence of *a mapping* from the observed channels to some unknown channels at another set of antennas and another frequency band.

- Based on the existence of a mapping function, the paper proposes to utilize the expressive and universal representational power of DNNs [30, 34] to learn that function. It proposes a DNN architecture that is based on feedforward networks. The architecture takes advantage of residual learning to allow the efficient development of a deep network [6].

- Using the DeepMIMO data-generation framework [68], the proposed deterministic prediction framework and the DNN architecture are studied on three different datasets representing three different communication scenarios. These scenarios are for co-located and distributed MIMO systems, and as such, the evaluation on the three datasets provides important insights into the advantages and limitations of the proposed framework and architecture.

2.2 Related Work

Channel acquisition is a fundamental problem in large-scale MIMO, and as such, a lot of research has gone into investigating solutions and approaches to combat its challenges. Overall, that research could loosely be grouped into two categories. The first category focuses on using signal processing to reduce the channel acquisition overhead [69–72]. In [69], the parameters of the uplink channels, such as the angles of arrival and path delays were estimated and used to construct the downlink channels at an adjacent frequency band. This frequency extrapolation concept was further studied in [70] where lower bounds on the mean squared error of the extrapolated channels were derived. On a relevant line of work, [71, 72] proposed to leverage the channel knowledge at one frequency band (sub-6 GHz) to reduce the training overhead associated with design the mmWave beamforming vectors, leveraging the spatial correlation between the two frequency bands. A common feature of all the work in [69–72] is the requirement to first estimate some spatial knowledge (such as the angles of arrival) about the channels in one frequency band and then leverage this knowledge in the other frequency band. This channel parameter estimation process, however, is fundamentally limited by the system and hardware capability in resolving these channel parameters, which highly affects the quality of the extrapolated channels.

The second category of research is centered around the use of machine learning to overcome various channel acquisition challenges [18, 20, 21, 63]. In [20], the authors address the problem of channel feedback in FDD massive MIMO systems using deep learning. They develop a DNN architecture that learns to encode (compress) and decode (reconstruct) the estimated downlink channels. Despite its elegance, this work is limited to the feedback problem and does not address the more prevalent issue of downlink channel training. The work in [21], [18], and [63] takes a step towards addressing that issue. [21] proposes the use of generative adversarial networks [73] to learn the mmWave channel covariance at a basestation and repurpose that covariance for the precoder design. On the other hand, [18, 63] opts to using feedforward neural networks [73] to learn a mapping from the uplink channel to the best mmWave beamforming or reflection vector. All those papers provide interesting treatments of the downlink training issue, but they all fall short in providing a holistic view on the channel acquisition problem; they utilize specific properties that are

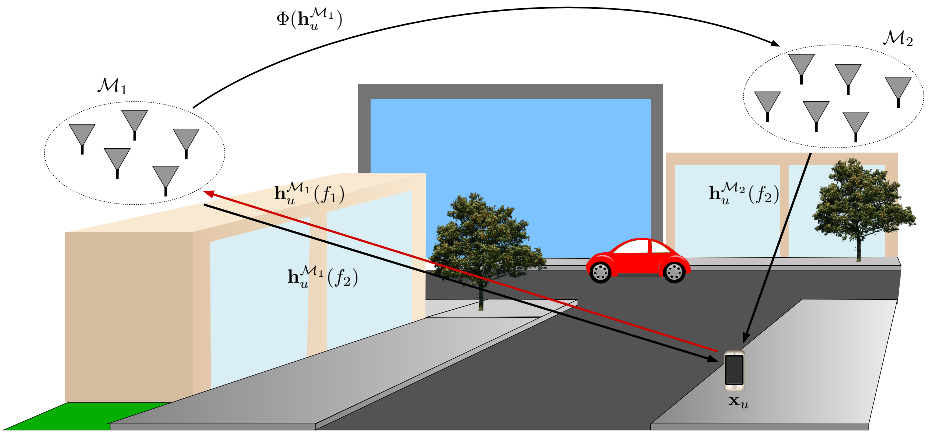

$\Phi(\mathbf{h}_u^{\mathcal{M}_1})$
$\mathcal{M}_1$
$\mathcal{M}_2$
$\mathbf{h}_u^{\mathcal{M}_1}(f_1)$
$\mathbf{h}_u^{\mathcal{M}_2}(f_2)$
$\mathbf{h}_u^{\mathcal{M}_1}(f_2)$
$\mathbf{x}_u$

Figure 2.1: A Massive MIMO System Illustrating The Channel Prediction Idea. For a User u, The Channels $\mathbf{h}_u^{\mathcal{M}_1}(f_1)$ at Antenna Set $\mathcal{M}_1$ and Frequency Band f_1 Could Be Mapped to $\mathbf{h}_u^{\mathcal{M}_1}(f_2)$ at The Same Set of Antennas $\mathcal{M}_1$ but Different Frequency Band f_2 or $\mathbf{h}_u^{\mathcal{M}_2}(f_2)$ at Another Set $\mathcal{M}_2$ and Frequency Band f_2.

inherent to either mmWave systems or large intelligent surfaces, i.e., the sparsity of the mmWave channel covariance [21], the existence of a predefined beamforming codebook [18], and the deployment of reflective (passive) elements in the large surface antenna array [63].

2.3 System and Channel Models

This section presents the system and channel models used throughout this paper. Both systems are chosen to be as comprehensive as possible to capture the generality of the proposed framework.

System model: Consider the general system model in Fig. 2.1 where one user at position $\mathbf{x}_u$ can communicate with one of two candidate sets of antennas, namely $\mathcal{M}_1$ and $\mathcal{M}_2$, over one of two frequency bands f_1 and f_2. This system does not impose any constraints on the relation between the two antenna sets, $\mathcal{M}_1$ and $\mathcal{M}_2$, nor on the frequencies f_1 and f_2; therefore, it captures several special cases such as (i) the case when some antennas are common between the two antenna sets, or (ii) when the two antenna sets use the same frequency $f_1 = f_2$. This allows a well-rounded study and analysis of the proposed framework. It also helps draw important insights for both co-located and distributed (cell-free) massive MIMO systems and for both Time Division Duplex (TDD) and FDD system operation modes, as will be discuss in Section 2.4.2.

Channel Model: Let $h_u^m(f_1)$ denote the channel from user u to antenna m in the antenna set $\mathcal{M}_1$ at the frequency f_1. Assume that this propagation channel consists of L paths. Each path ℓ has a distance d_ℓ from the user to antenna m, a delay τ_ℓ, and a complex gain $\alpha_\ell = |\alpha_\ell| e^{j\phi_\ell}$. The channel $h_u^m(f_1)$ can then be written as

$$h_u^m(f_1) = \sum_{\ell=1}^{L} |\alpha_\ell| e^{j\phi_\ell} e^{-j2\pi f_1 \tau_\ell}. \tag{2.1}$$

Note that the magnitude of the path gain $|\alpha_\ell|$ of path ℓ depends on (i) the distance d_ℓ that this path travels from the user to the scatterer(s) ending at the receiver, (ii) the frequency f_1, (iii) the transmitter and receiver antenna gains, and (iv) the cross-section and dielectric properties of the scatterer(s). The phase ϕ_ℓ also depends on the scatterer(s) materials and wave incident/impinging angles at the scatterer(s). Finally, the delay $\tau_\ell = \frac{d_\ell}{c}$, where c is the speed of light. By reciprocity, we consider $h_u^m(f_1)$ as also the downlink channel from antenna m to user u. Now, we define the $|\mathcal{M}_1| \times 1$ channel vector $\mathbf{h}_u^{\mathcal{M}_1}(f_1) = [h_u^1(f_1), ..., h_u^{|\mathcal{M}_1|}(f_1)]^T$ as the channel vector from user u to the antennas in set $\mathcal{M}_1$. Similarly, we define the channel vector $\mathbf{h}_u^{\mathcal{M}_2}(f_2)$ for the channel between user u and the antennas in set $\mathcal{M}_2$.

2.4 Deterministic Channel Prediction

In this paper, the objective is to provide an answer to Q.1 and show how it could be used to tackle channel-related challenges in large-scale MIMO. This section will first put Q.1 in formal terms, and, then, it will proceed to provide the main argument that underlies the channel prediction framework.

2.4.1 Problem Definition

Q.1 poses an interesting and also fundamental question to large-scale MIMO system; if a mapping between two sets of antennas (whether operating at the same or different frequency bands) exists and could be characterized or modeled, this means that a MIMO system only needs to estimate the channels at one set of antennas and use them to directly predict the channels at all the other antennas for the same or different frequency band. This can dramatically reduce the strain channel-acquisition imposes on large-scale MIMO systems. With this motivation in mind, the problem in Q.1 is formally divided into two main parts. The first one formulates the mapping function and its existence while the second part investigates its modeling. Let $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$ represents the channel mapping function defined as

$$\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}:\left\{\mathbf{h}_{u,\mathcal{M}_1}(f_1)\right\}\rightarrow\left\{\mathbf{h}_{u,\mathcal{M}_2}(f_2)\right\}, \tag{2.2}$$

then, the two parts of the channel mapping problem are stated as follows:

Part 1: *Does the mapping* $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$ *exist?*

Part 2: *If* $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$ *exists, how to model it?*

The rest of this section investigates the existence of the channel mapping function and discusses the benefits and practical implications of the deterministic channel prediction framework. Finally, Section 2.5 will describes how deep learning can address the question on how to model the mapping function $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$.

2.4.2 Existence of Mapping Function

Consider the system and channel models in Section 2.3. Addressing the question in **Part 1** starts by investigating the existence of the position to channel and channel to position mapping functions, and the results derived from examining the two functions will establish a proposition for the existence of $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$.

Existence of position to channel mapping: Consider the channel model in (2.1), where the channel from user u at position $\mathbf{x}_u$ to antenna m is completely defined by the parameters $|\alpha_\ell|, \phi_\ell, \tau_\ell$ of each path and the frequency f_1. Note that these parameters, $|\alpha_\ell|, \phi_\ell, \tau_\ell$, are functions of the environment geometry, scatterer materials, the frequency f_1, in addition to the antenna and user positions, as explained in the discussion after (2.1). Therefore, for a given static communication environment (including the geometry, materials, antenna positions, etc.), there exists a deterministic mapping function from the position $\mathbf{x}_u$ to the channel $h_u^m(f_1)$ at every antenna element m [74]. More formally, if $\{\mathbf{x}_u\}$ represents the set of all candidate user positions, with the sets $\left\{\mathbf{h}_u^{\mathcal{M}_1}(f_1)\right\}$ and $\left\{\mathbf{h}_u^{\mathcal{M}_2}(f_2)\right\}$ assembling the corresponding channels at antenna sets $\mathcal{M}_1$ and $\mathcal{M}_2$, then the position-to-channel mapping functions $\mathbf{g}_{\mathcal{M}_1,f_1}(.)$ and $\mathbf{g}_{\mathcal{M}_2,f_2}(.)$ are defined as

$$\mathbf{g}_{\mathcal{M}_1,f_1} : \{\mathbf{x}_u\} \rightarrow \left\{\mathbf{h}_u^{\mathcal{M}_1}(f_1)\right\}, \tag{2.3}$$

$$\mathbf{g}_{\mathcal{M}_2,f_2} : \{\mathbf{x}_u\} \rightarrow \left\{\mathbf{h}_u^{\mathcal{M}_2}(f_2)\right\}. \tag{2.4}$$

These deterministic mapping functions for a given communication environment can be numerically computed (or approximated) using ray-tracing simulations. However, it is important to emphasize here that while the existence of these position-to-channel mapping functions is assumed, the developed framework and deep learning algorithm will not require the knowledge of those mapping functions, as will be explained shortly in this section and in Section 2.5.

Existence of channel to position mapping: The next step is to investigate the existence of the mapping from the channel vector $\mathbf{h}_u^{\mathcal{M}_1}(f_1)$ of the antenna set $\mathcal{M}_1$ to the user position. For that, the following assumption is adopted.

Assumption 1 *The position-to-channel mapping function,* $\mathbf{g}_{\mathcal{M}_1,f_1} : \{\mathbf{x}_u\} \rightarrow \left\{\mathbf{h}_u^{\mathcal{M}_1}(f_1)\right\}$, *is bijective.*

This assumption means that every user position in the candidate set $\{\mathbf{x}_u\}$ has a unique channel vector $\mathbf{h}_u^{\mathcal{M}_1}(f_1)$. It is important to note here that the bijectiveness of this mapping, $\mathbf{g}_{\mathcal{M}_1,f_1}$, depends on several factors including (i) the number and positions of the antennas in the set $\mathcal{M}_1$, (ii) the set of candidate user locations, and (iii) the geometry and materials of the surrounding environment. While it is hard to guarantee the bijectiveness of $\mathbf{g}_{\mathcal{M}_1,f_1}(.)$, this mapping is actually bijective with high probability in many practical wireless communication scenarios [74].

Using the above, define the channel-to-position mapping function $\mathbf{g}^{-1}_{\mathcal{M}_1,f_1}(.)$ as the inverse of the mapping $\mathbf{g}_{\mathcal{M}_1,f_1}(.)$, i.e.,

$$\mathbf{g}^{-1}_{\mathcal{M}_1,f_1} : \left\{\mathbf{h}_u^{\mathcal{M}_1}(f_1)\right\} \rightarrow \{\mathbf{x}_u\} \tag{2.5}$$

Under Assumption 1, such inverse mapping, $\mathbf{g}^{-1}_{\mathcal{M}_1,f_1}(.)$, exists. In fact, it is widely adopted in the wireless positioning and fingerprinting literature [74, 75].

Existence of channel to channel mapping: Geared with the discussion on the existence of the position-to-channel and channel-to-position mapping functions, it is now time to address the main question of **Part 1**, the existence of the channel-to-channel mapping function $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$. This is done through the following proposition.

Proposition 1 *For a given communication environment, and if assumption 1 is satisfied, then there exists a channel-to-channel mapping function,* $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$*, that equals*

$$\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2} = \mathbf{g}_{\mathcal{M}_2,f_2} \circ \mathbf{g}^{-1}_{\mathcal{M}_1,f_1} : \left\{\mathbf{h}_u^{\mathcal{M}_1}(f_1)\right\} \rightarrow \left\{\mathbf{h}_u^{\mathcal{M}_2}(f_2)\right\} \tag{2.6}$$

Proof: The proof follows from (i) the existence of the mapping $\mathbf{g}^{-1}_{\mathcal{M}_1,f_1}(.)$ under assumption 1, (ii) the existence of the mapping $\mathbf{g}_{\mathcal{M}_2,f_2}(.)$ for any given environment, and (iii) the existence of the composite function $\mathbf{g}_{\mathcal{M}_2,f_2} \circ \mathbf{g}^{-1}_{\mathcal{M}_1,f_1}(.)$ since the domain of $\mathbf{g}_{\mathcal{M}_2,f_2}(.)$ is the same as the co-domain of $\mathbf{g}^{-1}_{\mathcal{M}_1,f_1}(.)$, and they both equal to $\{\mathbf{x}_u\}$. □ The existence of the channel-to-channel mapping function in **Proposition 2** establishes the main and core argument of the proposed deterministic channel-prediction framework. It has several important implications that will be discussed in the next subsection. In addition, it motivates further research on how to characterize that mapping function, giving rise to the question in **Part 2**.

2.4.3 Implications of The Proposed Framework and Practical considerations:

Consider a communication setup with a basestation employing multiple antennas (co-located or distributed), Proposition 2 means that once a subset of these antennas is identified such that it satisfies the bijectiveness condition in Assumption 1, then there exists a way (mapping function) that can map the channels at this set of antennas to the channels at all other antennas, even if they are communicating at a different frequency. This result yields interesting gains for both co-located and distributed massive MIMO systems, some of which is discussed in more details below.

- **FDD Co-located and Distributed Massive MIMO:** The general setup adopted in this section and illustrated in Fig. 2.1 reduces to the special case of FDD massive MIMO systems when $\mathcal{M}_1 \subseteq \mathcal{M}_2$ and when f_1 and f_2 represent the uplink and downlink frequencies. In this case, Proposition 2 implies that only a subset $\mathcal{M}_1$ of the basestation antennas need to be trained in the uplink. The uplink channels at these antennas can be directly mapped to the downlink channels at *all* the antennas, which significantly reduces the training and feedback overhead in these systems. Such gains will be illustrated in Section 2.7. It is worth noting here that this result maps the channels at both space and frequency. Therefore, it includes the following two special cases when only mapping in space or frequency is applied.

  1. *Mapping channels in space* is when $\mathcal{M}_1 \subseteq \mathcal{M}_2$ and $f_1 = f_2$ (representing

the downlink frequency). In this case, Proposition 2 means that only a few antennas need to be trained in the downlink and the rest can be constructed by channel prediction. For example, consider a basestation with 100 antennas. If 5 antennas are enough to satisfy the bijectiveness condition in Assumption 1, then only 5 antennas could be downlink trained and fed back instead of the 100 antennas, which is a missive reduction in the training/feedback overhead.

2. *Mapping channels in frequency* is when $\mathcal{M}_1 = \mathcal{M}_2$ and f_1, f_2, respectively, represent the uplink and downlink channels. In this case, Proposition 2 means that the uplink channels can be directly mapped to the downlink channels which completely eliminates the downlink training/feedback overhead.

- **TDD Distributed (Cell-free) Massive MIMO:** In TDD cell-free massive MIMO systems, the distributed antenna terminals estimate the uplink channels and use it for the downlink transmission. To avoid the need for forwarding all the uplink channels from the terminals to the central processing unit, the initial proposals for cell-free massive MIMO systems adopted conjugate beamforming where every terminal independently designs its downlink precoding weight. If feeding forward all the channels to the central processing is feasible, then several gains can be achieved, such as the ability to adopt more sophisticated precoding strategies and advanced user scheduling among others. Feeding forward all the channels to the central processing unit, however, is associated with high overhead that can limit the scalability of cell-free massive MIMO systems. Interestingly, Proposition 2 suggests that only a subset $\mathcal{M}_1 \subseteq \mathcal{M}_2$ of these antennas need to forward their channels to the central unit which can map them to the channels at all the other antennas. This has the potential of significantly reducing the channel feed-forward overhead associated with cell-free massive MIMO systems, rendering those systems more scalable.

It is also worth mentioning that the channel mapping result in Proposition 2 can also have several interesting applications in mmWave systems, such as using the channels collected at a few distributed antennas to predict the best beam at an antenna array [18], or using sub-6GHz channels to predict the mmWave blockages and beamforming vectors, as will be shown in Chapter 3.

How accurate the bijectiveness assumption is? Bijectiveness is not a stranger to some relevant research directions such as fingerprint positioning, where there is a growing body of work adopting the bijectiveness assumption. The fundamental idea behind fingerprinting is to construct a training set of wireless channels obtained from different user positions in a wireless environment, and utilize it to identify a new user's position from its estimated channels at several antennas [75]. Therefore, fingerprinting in essence aims to learn a positioning function that maps those estimated channels to a user's position, which is fundamentally based on the bijectiveness assumption. Publications like [74] explicitly argues that the position-to-channel function is bijective, and, hence, the positioning function exists. Other publications on fingerprinting like [75–77] do not explicitly rely on the bijectiveness assumption, yet their proposed fingerprinting approaches suggest that the position-to-channel function is bijective to some extent.

Bijectiveness could also be verified empirically; a soft measure for bijectiveness could be defined to reflect the similarity between two sets of channels belonging to two users at different positions (i.e., $\mathbf{x}_u$ and $\mathbf{x}_{u'}$) and observed at a set of antennas $\mathcal{M}_1$. The soft measure is based on Normalized Mean Squared Error (NMSE) and a user-specific channel subspace. Appendix A details the proposed approach to measure bijectiveness and presents two experimental studies for that. In a nutshell, the results of both experiments suggest that massive MIMO channels of any two *well-separated* users are very likely to be bijective, especially when the number of antennas is large. That being said, however, the results do not guarantee bijectiveness or provide clear conditions under which it is maintained.

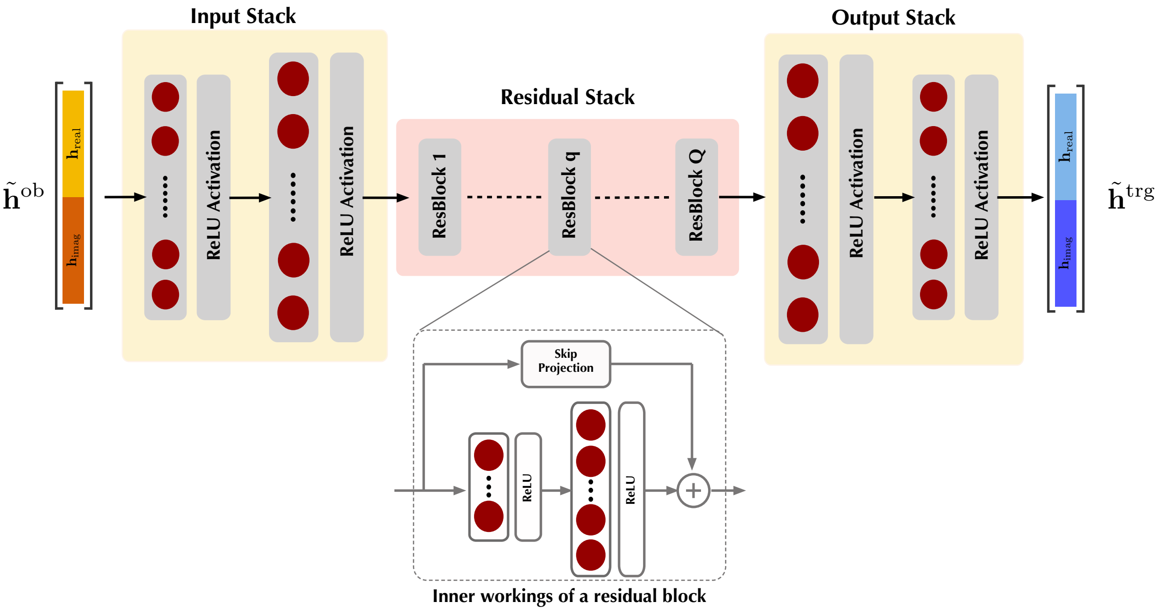


Figure 2.2: A schematic illustrating the architecture of the proposed DNN for channel prediction. It also provides a zoom-in on the architecture of the residual block and the arrangement of the input and output vectors.

Probabilistic Errors: Proposition 2 implies that for a given communication environment and under Assumption 1, there exists a deterministic channel-to-channel mapping function. In other words, given the channels at one antenna set $\mathcal{M}_1$, there is a way to predict exactly the channels at the other antenna set $\mathcal{M}_2$ which could even be a different frequency. In practice, however, there are a few factors that can add some probabilistic error to this channel prediction such as the measurement noise, the limited ADC bandwidth, and the time-varying channel fading. Evaluating the impact of these practical considerations on the channel prediction error is very important, and it is partially touched upon in this work, as will be discussed in Section 2.8.

2.5 Proposed Deep Learning Solution

Section 2.4 presents the core argument of the deterministic channel-prediction framework, which addresses the question in **Part 1** of the problem definition in Section 2.4.1. The framework sketches a road-map for how some known channels could be mapped to another set of unknown channels; it describe the relation between the two sets of channels using a deterministic yet unknown mapping function $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$, and it establishes the necessary condition for that function to exist. Despite the interesting perspective on the relation, the framework does not provide a description of the vessel by which the road-map it lays could be traveled. In particular, it does not say how one could characterize or model the mapping function, which leaves the question in **Part 2** of the problem definition unanswered.

This section is dedicated to address the question of how to characterize or model the mapping function $\mathbf{\Phi}_{\mathcal{M}_1,f_1\rightarrow\mathcal{M}_2,f_2}(.)$. The proposed approach to do that revolves around posing the problem as a machine learning problem. It utilizes the recent advances in machine learning [73] to design a learning algorithm capable of capturing the mapping function using some training dataset. In particular, the learning of the mapping function will be posed as a regression problem, where a DNN is designed to capture the relation between the known channels (henceforth referred to as the *observed channels*) and the unknown channels (henceforth referred to as the *target channels*). This choice of approach is motivated by two main observations: (i) the dependency of the mapping function on many environmentally-specific factors that call for an adaptive and data-driven approach (see Sections 2.3 and 2.4.2), and (ii) the proliferating empirical and theoretical evidence in favor of DNNs as universal function approximators [30, 34, 78]. The rest of this section will present the proposed DNN architecture to learn the channel-mapping function and discuss how this architecture is implemented.

2.5.1 Network Architecture

The proposed architecture is designed to take advantage of two key findings in the field of deep learning, which are the universal approximation property of feedforward (multilayer perceptron) networks [34, 78] and the ability of residual learning to facilitate the training of deep architectures [6]. Since the objective is to learn the *unknown* channel-mapping function with high fidelity, feedforward networks present an appealing choice to take on that task; long ago, they have been shown to have a large capacity to approximate functions by merely implementing a single hidden layer with adjustable number of neurons (adjustable breadth) [78]. More recently, accumulating evidence has been arguing the importance of increasing the depth of a network as opposed to its breadth [6, 30, 33], for it yields multiple benefits in terms of practicality and expressibility. Learning deep architecture, however, has its own challenges. Right off the bat, the performance of deep feedforward architectures degrades as more layers are stacked[1], indicating a heightened level of training difficulty. Residual learning has been proposed [6] to remedy such problem. It has been shown to enable the learning of very deep architectures that have defined the state-of-the-art on many machine learning tasks, e.g., image classification [6] and speech recognition [10, 11].

Guided with the two finding above, the proposed DNN architecture is designed to comprise three major stacks, the input, residual, and output stacks. Fig. 2.2 displays a schematic of the architecture. The first stack is built with two sequences of full-connected and ReLU activation layers, with the option of implementing dropout in-between when needed. This layer transforms the input channels into a high dimensional vector that is, then, fed to the residual stack. The residual stack consists of Q residual blocks, each of which has two sequences of fully-connected and ReLU layers and a skip-projection layer parallel to the

[1] This degradation is not a result of the famous “vanishing gradient” problem. See [6] for more details.

two sequences, see the example in Fig. 2.2. The skip projection is implemented with a fully-connected layer when the dimensions of the input vector to the residual block and the output vector of the second ReLU of the block do not match. Otherwise, the skip projects is merely an identity projection that passes the input vector to the summation operation. The last stack, i.e., output stack, takes in the output of the last residual block and maps it back to a high-dimensional vector, matching in size to that output by the input stack. This vector is finally projected onto the vector space of the target channel, completing the mapping operation. It is important to note here that the three-stack architecture has been developed via a sequence of experimentation on different datasets. The details on the architecture hyper-parameters, training, and performance are discussed in Sections 6.5 and 2.7.

2.5.2 System Operation

The proposed DNN architecture is designed to be *transparent* to the wireless communication system. This means the architecture (and the channel-mapping capability it provides) is run in the background of the wireless system and is only brought online once it achieves a satisfying mapping performance. Such transparency leads to a system operating in two modes:

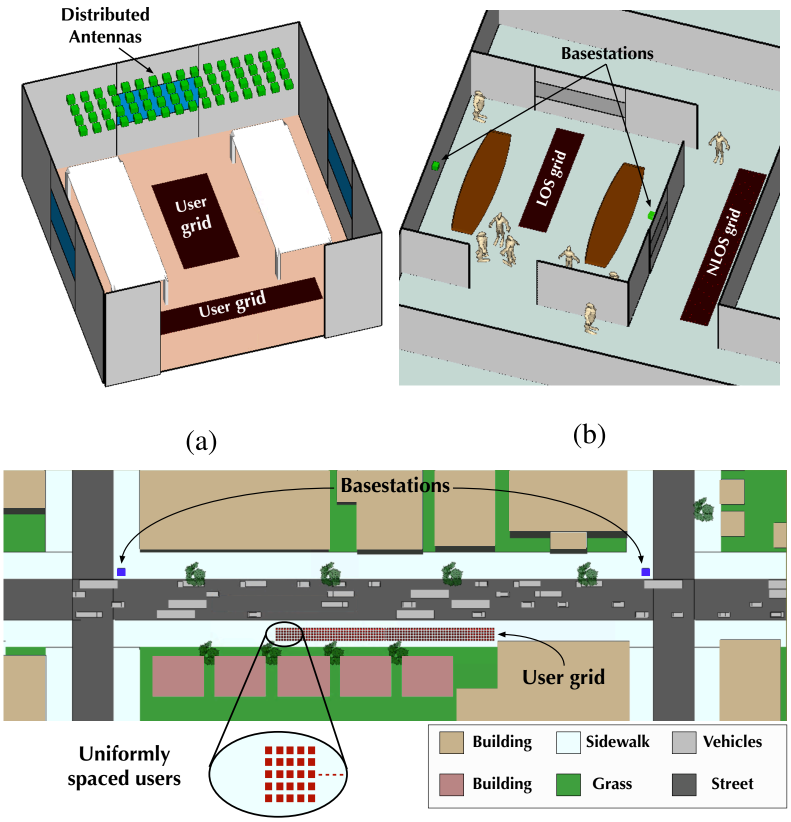
- **Background training:** this mode is where the wireless system operates without the deep learning algorithm using some classical means. During its operation, the system serves the users in the environment and collects observed and target channels. The nature of those channels depends on: (i) type of MIMO system being implemented, and (ii) the overhead that needs to be handled with the deep learning algorithm. For instance, an FDD massive MIMO system may estimate uplink and downlink channels to be used as observed and target channels, respectively. See Section 2.7. The collected channels are used to train and validate the deep architecture in the background.

- **Solution deployment:** this mode kicks in once the deep architecture has reached a satisfying performance defined by some system-performance metric (e.g., certain NMSE level). When the architecture is deemed ready, it is plugged into the part of the system that needs it. For the same FDD system example above, the trained architecture is plugged into the basestation processing unit, so it receives the estimated uplink channels and directly maps them to the downlink channels, eliminating the need to do downlink channel training.

Such system operation is critical for compatibility and adaptability reasons; it shields the wireless system operation from the training of deep learning architecture. This allows the architecture to be trained whenever needed without major interruptions to the wireless system operation. Furthermore, the transparency in the system operation provides an avenue for data collection. As the wireless system needs to estimate channels during its operation in the background training mode, those channels could be accumulated to build a wireless training dataset for the deep learning algorithm. This is expected to provides better adaptability as the data samples are coming directly form the wireless environment where the system operates.

2.6 Experimental Setup

For the sake of evaluating and thoroughly studying the proposed deterministic channel-prediction framework and the deep learning algorithm, three communication scenarios will be considered in this paper. They provide three different massive MIMO communication scenarios, form which three different channel datasets are constructed. These dataset will be used to train and evaluate the proposed solution separately. The following two subsections will give more details on the scenarios, datasets, and algorithm training. Performance evaluation of the algorithm, on the other hand, is presented and discussed in the following sections, Sections 2.7 and 2.8.



(a)

(b)

(c)

Figure 2.3: An Illustration of The Three Evaluation Environments. (a) and (b) Show Perspective Views of Two Stationary Scenarios for Indoor Environments with Different MIMO System Configuration, and (c) Is a Top-view of The Third Scenario Representing an Outdoor Dynamic Environment.

Table 2.1: The Adopted DeepMIMO Dataset Parameters

Parameter	Distributed	Colocated	Dynamic
Scenario name	I1	I3	O1_dyn
Operating frequency (GHz)	2.4 and 2.5	2.4 and 2.5	3.4 and 3.5
Active BS	1 to 64	1 and 2	1 and 2
Active users	1 to 502	1 to 1159	1 to 5
Number of BS antennas in (x, y, x)	(1,1,1)	(1,64,1)	(64,1,1)
System BW (GHz)	0.02	0.02	0.02
Number of OFDM sub-carriers	64	64	64
OFDM sampling factor	1	1	1
OFDM limit	16	16	16
Number of paths	5	15	15
Number of scenes	1	1	518

2.6.1 Evaluation Scenario and Dataset

Three massive MIMO scenarios are considered in this paper. They are all based on the publicly available DeepMIMO data-generation framework [68]. The first two scenarios, namely the distributed and colocated scenarios, represent stationary indoor wireless environments with two different massive MIMO systems and user settings. The distributed scenario, depicted in Fig. 2.3a, represents a distributed MIMO system serving LOS users in a conference room while the colocated scenario, depicted in Fig. 2.3b, has a colocated MIMO system deployed in a similar conference room but serving LOS and NLOS users in the room and its hallways, respectively. The third scenario, namely the dynamic scenario, takes a step towards more realistic wireless environments by considering a massive colocated MIMO system in an outdoor dynamic-scatterer settings, see Fig. 2.3c. The scenario depicts a typical downtown street with its different elements, e.g., trees, skyscrapers, vehicles,...etc, and it has a stationary grid of users, stationary basestations, and various moving scatterers in the form of vehicles. More information on the scenarios could be found in the DeepMIMO website [68].

The DeepMIMO generation script is used to construct three datasets of channel tuples, namely $\mathcal{S}_c = \{s_n\}_{n=1}^{N_c}$, where $c \in \{1, 2, 3\}$ representing, respectively, the distributed, colocated, and dynamic scenarios; s_n is the n-th tuple of channels obtained from the environment; and finally, N_c is the total number of tuples (samples) in the c-th scenario. All three scenarios have MIMO systems implementing Orthogonal Frequency-Division Multiplexing (OFDM) with K-subcarriers, but they differ in basestation configuration and scatterer dynamics. As such, the content of each tuple $s_n \in \mathcal{S}_c$ varies depending on the scenario. The following list details that:

- **Distributed scenario ($c = 1$):** it has a set of 64 distributed antennas ($|\mathcal{M}| = 64$) scattered in the environment, and, hence, a tuple here has two channel matrices representing the channels between all 64 antennas and a user u across all K subcarriers and at two different frequencies f_1 and f_2, namely $s_n = (\mathbf{H}^{\mathcal{M}}(f_1), \mathbf{H}^{\mathcal{M}}(f_2))_u$ where $\mathbf{H}^{\mathcal{M}}(f_1), \mathbf{H}^{\mathcal{M}}(f_2) \in \mathbb{C}^{64\times K}$ are channel matrices at f_1 and f_2. The total number of samples is equal to the total number of users in the environment, i.e., $N_1 = U_1$.

- **Colocated scenario ($c = 2$):** it has two basestations, each of which is equipped with a $|\mathcal{M}_a|$-element ULA with $a \in \{1, 2\}$. A tuple here consists of four different channel matrices that correspond to the channels between the u-th user and both basestations at two frequencies f_1 and f_2. More specifically, $s_n = \Big(\mathbf{H}^{\mathcal{M}_1}(f_1), \mathbf{H}^{\mathcal{M}_1}(f_2), \mathbf{H}^{\mathcal{M}_2}(f_1),$ $\mathbf{H}^{\mathcal{M}_2}(f_2)\Big)_n$, where $\mathbf{H}^{\mathcal{M}_a}(.) \in \mathbb{C}^{|\mathcal{M}_a|\times K}$. The total number of samples is equal to the total number of users in the environment, i.e., $N_2 = U_2$.

- **Dynamic scenario ($c = 3$):** similar to the colocated scenario, this one has two basestations equipped with $|\mathcal{M}_1|$- and $|\mathcal{M}_2|$-element ULAs. However, since this scenario has dynamic scatterers, a new dimension is introduced in the dataset, which is time instance t or as it will be henceforth referred to as *scene*. The scenario is generally composed of T scenes, each of which sees the scatterers assuming different positions in the environment. A tuple in this dataset has four channel matrices corresponding to the channels between the u-th user at the t-th scene and both basestations at two frequencies f_1 and f_2, i.e., $s_n = \left(\mathbf{H}^{\mathcal{M}_1}(f_1), \mathbf{H}^{\mathcal{M}_1}(f_2), \mathbf{H}^{\mathcal{M}_2}(f_1), \mathbf{H}^{\mathcal{M}_2}(f_2)\right)_n$ where $\mathbf{H}^{\mathcal{M}_a}(.) \in \mathbb{C}^{|\mathcal{M}_a| \times K}$. The total number of samples in this dataset depends on both number of users and number of scenes, i.e., $N_3 = U_3 T$.

Table 2.1 summarizes the generation hyper-parameters used for each dataset. For each scenario, they describe the configuration of the generation process, which includes, but not limited to, the number of basestations, number of antennas per basestation, operating frequency, number of subcarriers and so forth. The total number of data samples for the stationary scenarios (distributed and colocated) is $|\mathcal{S}_1| = U_1 \approx 151 \times 10^3$ and $|\mathcal{S}_2| = U_2 \approx 118 \times 10^3$. while the number of samples in the dynamic scenario is $|\mathcal{S}_3| = U_3 T = (405)(518) \approx 210 \times 10^3$. The datasets are all split $70 - 30\%$ to form three training and three validation datasets that will be used in the Sections 2.6.3. 2.7, and 2.8 to train and evaluate the proposed DNN architecture.

2.6.2 Data Pre-processing

To achieve good training and prediction performances, the inputs and targets of a neural network (or any machine learning algorithm for that matter) commonly undergo a preprocessing pipeline [18, 79, 80]. In this paper, the adopted pipeline comprises two main components: (i) data normalization, and (ii) data reshaping. The former separately estimates the average powers of the inputs and outputs from the training dataset, and it uses them to normalize the inputs and output to have unity element-wise average power. Formally, if $\mathbf{H}^{\mathcal{M}_1}(f_1)$ is the input, then it is normalized as follows

$$\mathbf{H}_{\text{norm}}^{\text{ob}} = \frac{1}{\sigma}\mathbf{H}^{\mathcal{M}_1}(f_1), \tag{2.7}$$

where

$$\sigma = \sqrt{\frac{1}{N_c}\sum_{n=1}^{N_c} ||\mathbf{H}_n^{\mathcal{M}_1}(f_1)||_F^2}, \tag{2.8}$$

N_c is the total number of samples in the training dataset of the c-th scenario, and $\mathbf{H}_{\text{norm}}^{\text{ob}}$ is a normalized input channel matrix. The same process could be followed to estimate the average power and normalize the outputs. Through experimentation, this choice of normalization has been found to be very effective for the training of the proposed architecture. The second component, i.e., data reshaping, is a consequence of using fully-connected layers as the building block of the proposed DNN. These layers expect their input to be fed in the form of a high dimensional vector, which necessitates the structuring of the input and output channels in the form of vectors. Furthermore, popular deep learning software-development frameworks, such as PyTorch [81] and TensorFlow [82], only support real-valued computations. Hence, the inputs and targets needs to be converted to that format. The conversion of choice in this paper is to decouple and stack the real and imaginary components into one high dimensional vector, see Fig. 2.2. Formally, this is given by

$$\tilde{\mathbf{h}}^{\text{ob}} = \left[\Re\left\{\text{vec}(\mathbf{H}_{\text{norm}}^{\text{ob}})\right\}, \Im\left\{\text{vec}(\mathbf{H}_{\text{norm}}^{\text{ob}})\right\}\right]^T \tag{2.9}$$

$$\tilde{\mathbf{h}}^{\text{trg}} = \left[\Re\left\{\text{vec}(\mathbf{H}_{\text{norm}}^{\text{trg}})\right\}, \Im\left\{\text{vec}(\mathbf{H}_{\text{norm}}^{\text{trg}})\right\}\right]^T \tag{2.10}$$

where $\mathbf{H}_{\text{norm}}^{\text{trg}}$ is a normalized output channel matrix.

2.6.3 Network Training

The DNN architecture described in Section 2.5.1 is empirically optimized to learn the mapping function using the datasets described above. The experiments are conducted using the deep learning software development framework PyTorch [81] and on a machine with an NVIDIA® RTX 2080 Ti GPU, 128 GB RAM, and 10-core Intel® Xeon®. This optimization process includes: (i) determining the breadth of each full-connected layer, (ii) determining the number of residual blocks, and (iii) identifying the best set of training hyper-parameters. The final architecture is determined through experiments on the task of mapping uplink channels to downlink channels in the distributed and colocated datasets—details on the task and the performance are given in Section 2.7. The architecture that achieved reasonably good performance for that task is found to have 1024 and 4096 neurons for the input stack, 3 residual blocks with 512 and 1024 neurons, and 4096 and 2048 neurons for the output stack. This architecture has a total of 8 layers and $\approx$ 27 million parameters. The training is carried out using min-batches of input-output pairs. For each mini-batch B, a Mean Squared Error (MSE) loss is used as a training metric. This loss is given by

$$\mathcal{L}_{\text{MSE}} = \frac{1}{B}\sum_{b=1}^{B} ||\tilde{\mathbf{h}}^{\text{prd}} - \tilde{\mathbf{h}}^{\text{trg}}||_2^2 \tag{2.11}$$

where $\mathcal{M}$ is either the set of 64 antennas in the distributed scenario or the first $\mathcal{M}_1$ or second $\mathcal{M}_2$ basestations in the colocated and dynamic scenarios; and $\tilde{\mathbf{h}}^{\text{prd}}$ is the predicted channel vector that has the same dimensions as $\tilde{\mathbf{h}}^{\text{trg}}$. The hyper-parameters used to train and evaluate the final architecture are summarized in Table 2.2, and [83] has example implementation scripts for the proposed architecture.

2.7 Performance Evaluation on Stationary Environments

This section will kick off the discussion on the feasibility of the channel-mapping framework and its advantages and shortcomings by examining the performance of the proposed DNN in stationary environments, i.e., environments with stationary scatterers, stationary basestations (or antennas), and multiple users. The discussion here is split into two subsections representing two different case studies. The first one considers the distributed massive MIMO scenario while the second focuses on the colocated massive MIMO sce-

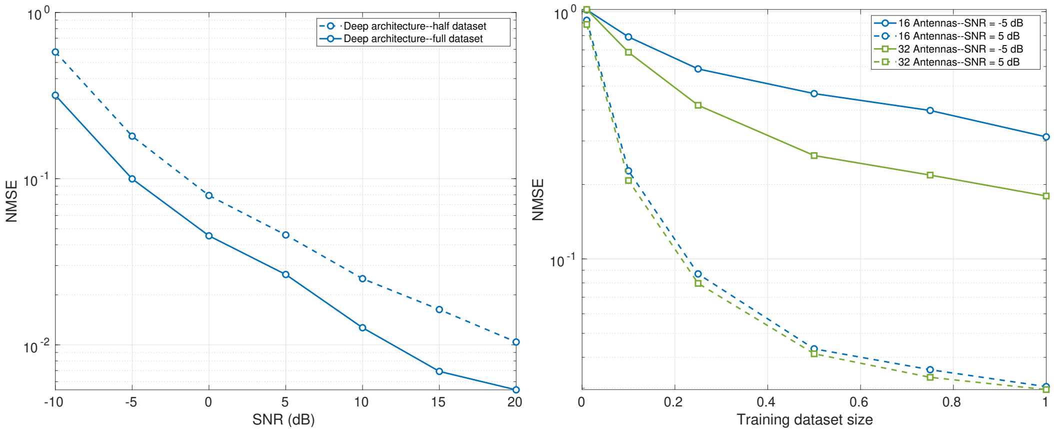
Table 2.2: Training Hyper-parameters of The Final Architecture

Hyper-parameter	Value
Solver	Adam [84]
Learning rate	1×10^{-3}
Number of epochs	350
Rate schedule	0.1 @ epoch 250
Batch size	5000
Dropout	0.5 @ SNR $\leq 5dB)$
Weight decay	1×10^{-5}

nario. It is important to point out here that all the results discussed in this section and the next one are obtained on the validation sets. Consult Section 6.5 for more information.

2.7.1 Case study 1: Distributed MIMO

Using the training dataset of the distributed MIMO scenario, the proposed architecture is trained to learn the following tasks: (i) the mapping from the uplink channels at frequency f_1 to the downlink at frequency f_2, and (ii) the mapping from sampled uplink channels at frequency f_1 to all uplink channels at the same frequency. The first task has obvious advantages for FDD massive MIMO systems; the uplink channels are commonly easily obtained with low channel-training overhead, and mapping those channels to the downlink ones is a desirable property that could reduce the severe downlink channel-training overhead. On the other hand, the second task sheds light on how channel prediction could be of value to TDD massive MIMO. In particular, scaling up the number of antennas (or terminals) in a distributed MIMO system creates a bottleneck for the fronthaul connecting those antennas to the central processing unit. Channel prediction could alleviate that fornthaul strain by



(a)

(b)

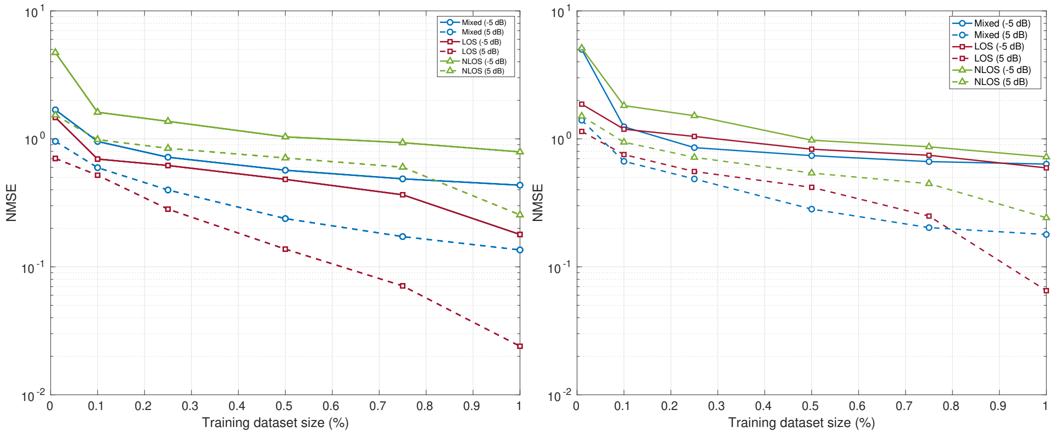
Figure 2.4: The Performance of The Proposed DNN in a Distributed MIMO Setting. (a) Depicts The NMSE Against SNR for Two Choices of Training Set Size. (b) Shows NMSE Versus The Size of The Training Set for Two SNR Levels and Two Choices of Number of Antennas.

mapping a subset of those channels to the rest of them.

The performance of the proposed DNN is illustrated in Fig. 2.4 for both tasks described above. For the first task (FDD distributed system), the NMSE of the mapped channels is calculated for a range of SNR values (from -10 to 20 dBs) and two training set sizes (50% and 100% of $|\mathcal{S}_1|$), and the performance is depicted in Fig. 2.4a. On the surface, the figure shows an expected trend where the NMSE performance improves as both SNR and training set size increase. A deeper look at the figure, however, reveals a more interesting observation, the gap in the NMSE performance between the two curves gradually vanishes as the system moves towards a high-SNR regime. For instance, at -5 dB, going from 50% of the training set to 100% realizes an NMSE improvement of ≈ 0.1, but at 10 dB, that same increase of data points barely secures an improvement of ≈ 0.01. This observation is important from two perspectives. The first is a machine learning perspective, where the observation suggests that less training data points are required when the system operates at high

SNR. This amounts to a reduced data collection burden and a speeded-up training process, both are desired for smooth and robust DNN implementation and communication-system operation. The other perspective concerns the bijectiveness assumption in the channel-mapping framework. Achieving an NMSE gap of 0.01 with half the training set constitutes the first evidence that a certain degree of bijectiveness holds in the environment; increasing the number of data points has little effect on the performance, hinting at another factor that causes the performance saturation.

The above bijectiveness evidence is further explored in the second task, in which not only the feasibility of the assumption is studied but also its implications on the TDD system operation. Fig. 2.4b plots the NMSE performance versus the training set size ($|\mathcal{S}_1|$) for two randomly-sampled antenna subsets (16 and 32 antennas) and two SNR regimes (-5 dB and 5 dB). The obvious observation from the figure is that the SNR regime plays a strong role in learning the mapping function. The proposed DNN seems to struggle in mapping the channels at low SNR across almost all training set sizes. However, what is more interesting is the performance of the DNN in a high-SNR regime (5 dB) with 16 or 32 antennas. The two curves have a small NMSE gap that gradually vanishes with more training data points. For instance, the gap at 10% training set is ≈ 0.02, and it drops to less than 10^{-3} with the whole training set. This is very important for two reasons. First, it strengthens the insight drawn from Fig. 2.4a; bijectiveness could be achieved in a stationary wireless environment, and that does not require the channel knowledge at a large number of antennas, e.g., 25% of the antennas is enough for the distributed scenario. The second reason lies in the implication of this shrinking gap on the system operation. It clearly indicates that leveraging the channel prediction framework could help mitigate the challenges of scaling up the number of antennas in a distributed MIMO system.



(a) (b)

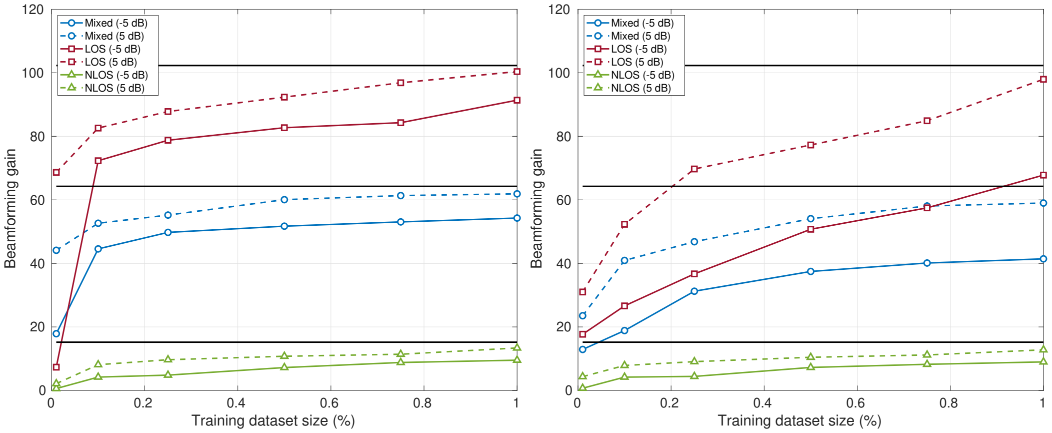
Figure 2.5: The NMSE Versus The Training Set Size for The Proposed DNN in The Colocated Scenario. (a) Presents The Results for Across-Frequency Mapping, and (b) Presents The Results for Across-Space-and-Frequency Mapping.

2.7.2 Case study 2: Colocated MIMO

The proposed DNN architecture is now studied in the colocated MIMO scenario. Similar to the distributed scenario, mapping from the uplink channels to the downlink ones has the same desirable impact of reducing the channel-training overhead. Hence, it is the first learning task considered in this section. However, what could be even more interesting in this communication setting is mapping the uplink channels at one basestation to the downlink channels at another basestation. Although such mapping is implied in *Case study 1* (Section 2.7.1), it has more interesting implications in a colocated MIMO system as it hints at the ability of the channel prediction framework to introduce a sense of inter-cell interference mitigation; for a massive MIMO basestation, the knowledge of the downlink channels at an adjacent basestation could be factored into the precooding process. Therefore, the second learning task in this section will focus on mapping the channels from one basestation to another.

Fig. 2.5 depicts the NMSE of the mapped channels versus the training set size for both tasks and two SNR levels, -5 and 5 dB. Since the scenario has distinct sets of LOS and NLOS users, the figure presents the NMSE performance of the DNN when it is trained on each set separately as well as on the mixed users case. Fig. 2.5a shows the performance for the mapping across frequency alone. It indicates that having LOS connection with a basestation results in a simpler learning problem regardless of the operating SNR. Such trend is expected and not at all surprising, especially given the results in *Case study 1*. What should be highlighted here is the value of the number of training data points in this scenario. Opposite to what has been observed in the distributed scenario, more data points seems to help improve the performance of the DNN in the LOS case and at high SNR (5 dB). This could be attributed to the colocated nature of the MIMO system. It induces higher correlation among the user's channels compared to the channels in the distributed case, which makes the learning task more difficult and in need for more data points. This correlation notion could be affirmed by the results on the second task that are presented in Fig. 2.5b. The figure also displays NMSE versus training set size and is organized in the same way Fig. 2.5a is. The mapping across space and frequency is expected to be more difficult as the two basestations have different views of the environment, and this makes the number of training data points more important. For example, increasing the data points from 50% of the training set to 100% secures, respectively, ≈ 0.3 and ≈ 0.11 improvements on the NMSE performance of the LOS and mixed-user cases.

To understand the effect of the NMSE results discussed above on the system performance, Fig. 2.6 plots the beamforming gain versus the training set size for both tasks under the same SNR values, assuming conjugate beamforming. The figure also breaks down the performance into three groups of curves, LOS, NLOS, and mixed users. Each group has the upper-bound gain (achieved only with full downlink channel knowledge), the gain at low SNR, and the gain at high SNR. Fig. 2.6a shows the beamforming gain for the first



(a) (b)

Figure 2.6: The Beamforming Gain Versus The Training Set Size for The Proposed DNN in The Colocated Scenario. (a) Presents The Results for Across-Frequency Mapping, and (b) Presents The Results for Across-Space-and-Frequency Mapping.

task. This performance turns the attention to an important finding; although the NMSE established the need to more data points to learn the mapping function, the communication penalty incurred from the relatively inaccurate mapping is somewhat limited especially at high SNR. For example, at 50% training set size, the achieved beamforming gain for mixed-users is, respectively, $\approx 7\%$ and $\approx 14\%$ shy off the upper-bound for SNR values of 5 and -5 dB. With more data points, the improvement in the beamforming gain may seem a little insignificant, but it is important to point out here that such improvement in the NMSE performance will reflect better on the beamforming gain when multi-user settings (multiple users are being served). The beamforming gain trend does not quite extend to the across space-and-frequency mapping task as it is expected to be more complex. Fig. 2.6b shows the gain dropping across different training sizes and SNR values. For instance, the mixed-user case sees the achievable gain degrading by $\approx 14\%$ and $\approx 40\%$ for the 50% training set size and 5 dB SNR. Again similar to the NMSE case, this degradation if mitigated using more data points.

2.8 Performance Evaluation in A Dynamic Environments

The above case studies and discussions are focused on stationary environments, and a natural question at this stage is whether those results could hold up in more realistic wireless communication settings or not. For that, the dataset $\mathcal{S}_3$ from the dynamic scenario (O1_dyn) is used to train the propose architecture. The focus in this case study is on the performance of the deterministic channel prediction framework with respect to environment dynamics. The dynamics in the wireless environment is quantified by the number of scenes considered to construct the dataset $\mathcal{S}_3$—see Section 2.6.1 for more information on the dataset.

Fig. 2.7 depicts the performance of the proposed architecture as the number of scenes increases. It is obtained for the task of predicting the downlink channels from observed uplink channels at the same basestation and at 5 dB SNR. The curves shown reflect the architecture behavior with respect to dynamics. It shows the progress of the training and validation losses—computed using (2.11) for two different number of scenes $T = 120$ and 280. With both choices of number of scenes, the training loss rapidly decreases as the training progresses, yet the validation loss displays clear signs of overfitting. The number of scenes effectively increases the number of data samples available for training, which helps the proposed architecture achieve slightly better loss value (on training and validation) at the beginning of the training. For example, 15% through the training process, the validation loss drops from ≈ 0.25 to ≈ 0.16 as the number of scenes increases from 120 to 280. However, this improvement is superficial as the architecture rapidly undergoes overfitting. The behavior the architecture displays suggests that the channel-to-channel mapping function is quite hard to capture with the dynamics.

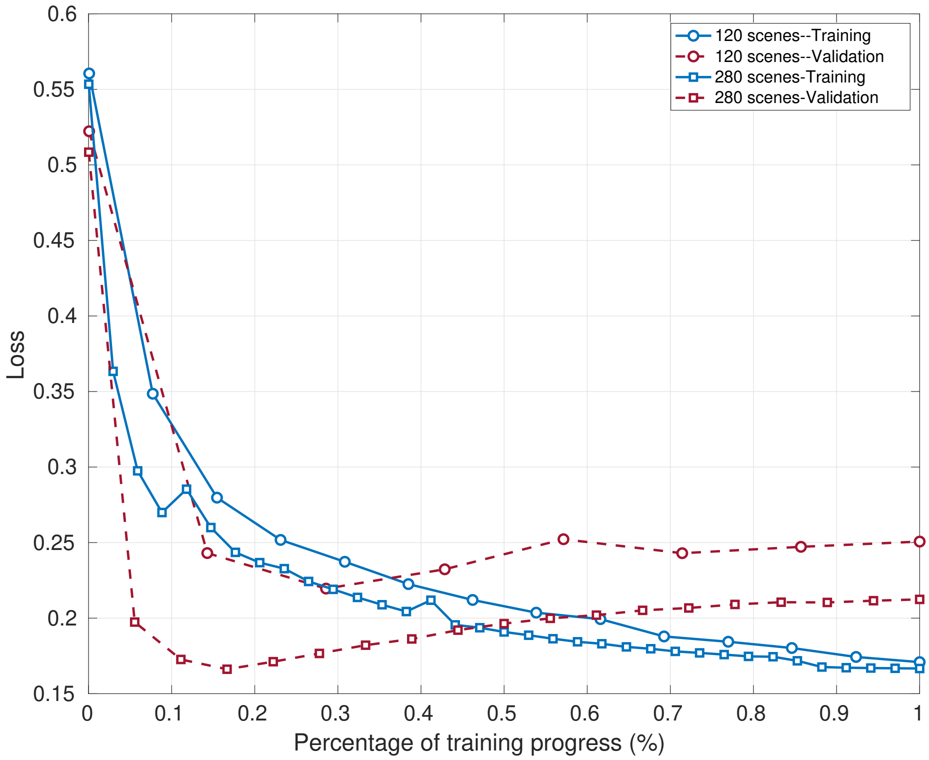


Figure 2.7: Training Progress Versus Training and Validation Losses for Two Different Number of Scenes T. The Training Progress Is Quantified as The Percentage of Completed Iteration, i.e., Ratio of Current Iteration to The Total Number of Iterations.

Chapter 3

DEEP LEARNING FOR MMWAVE BEAM AND BLOCKAGE PREDICTION USING SUB-6GHZ CHANNELS

3.1 Scope and Contributions

Scope

Chapter 2 lays the groundwork for channel prediction under the channel bijectiveness condition, and shows the feasibility of learning the prediction function using DNNs. The case studies discussed in that chapter show some interesting results on channel prediction in stationary environments and between neighboring frequency bands. However, a couple of questions may arise at this point.

> **Q.1:** *Due to the fact that material properties and signal propagation characteristics both change as the spacing between frequency bands increases, e.g., the difference between sub-6 GHz and mmWave bands, could similar results be obtained when the frequency spacing is quite large?* and
>
> **Q.2:** *The need to learn a complex-valued and high-dimensional target function using regression could be seen as a difficult problem. Therefore, could the prediction task be posed in different learning settings? i.e., does it really need to be posed as a regression problem?*

Both questions are interesting to address for two reasons: (i) the fact that modern and large-scale MIMO wireless networks inherently operate with multi-bands [85, 86], like sub-6 GHz and mmWave; and (ii) the vision that AI and specifically ML are going to be integral components to future large-scale MIMO networks [14, 37].

In an effort to address the above questions, this paper presents a third case study for the deterministic channel prediction framework. It considers the task of predicting mmWave channels from knowledge of sub-6 GHz channels. What is interesting in this task is not only its relevance to modern and future large-scale MIMO networks, but also the possibility of posing it as a classification task; mmWave channels in a certain wireless environment could be characterized by one or both of the following: (i) the choice of beamforming codebook, and (ii) the communication link status (i.e., whether the link is LOS or NLOS). Such characterization could be, from a ML perspective, viewed as a discretization of the mmWave channels. Therefore, one might choose to simplify the task of predicting the mmWave channels from their sub-6 GHz counterparts by posing it as classification problem. The objective of the problem is either to predict the label of the best beaforming vector in the codebook or predict if the link is LOS or NLOS. Both objectives have important implication to the wireless network and will be discussed in this paper.

Contributions

This paper considers dual-band systems where the base station and mobile users employ both sub-6 GHz and mmWave transceivers. It develops a theoretical argument for the use of sub-6 GHz channels to directly predict mmWave beamforming vectors and link status (henceforth referred to as predicting blockages), and it shows that deep learning models can be efficiently leveraged to achieve these objectives. The main contributions of this paper can be summarized as follows:

- We prove that for any given environment, there exists a mapping function that can predict the optimal mmWave beam (out of a codebook) directly from the sub-6 GHz channel if certain conditions are satisfied. These mapping functions, however, are hard to characterize analytically which motivated leveraging deep learning.

- Leveraging the universal approximation theory [78], we prove that large enough neural networks can learn how to predict the optimal mmWave beams directly from sub-6 GHz channel vectors with a success probability that can be made arbitrarily close to one.
- We show that a similar result can be established for blockage prediction, and identify the conditions under which the sub-6 GHz channels can be used to predict whether or not the mmWave LOS link is obstructed. We also prove that large enough neural networks can be exploited to learn this blockage prediction with an arbitrarily high success probability.
- We propose a deep neural network model that efficiently uses the sub-6 GHz channels to predict the optimal mmWave beams and blockage status. We also show that the transfer learning could utilized to reduce the learning time overhead.

The proposed deep learning based mmWave beam and blockage prediction solutions were evaluated using the publicly-available dataset DeepMIMO [68]. This dataset generates sub-6 GHz and mmWave channels using the accurate 3D ray-tracing simulator Wireless InSite [87] which incorporates the materials' dielectric properties at the two bands. The simulation results confirm the promising capability of deep learning models in learning how to predict the mmWave beams and blockages using sub-6 GHz channels, as explained in detail in Section 3.8.

3.2 Related Work

Estimating the channels at one frequency band using the channel knowledge at a different frequency band is attracting increasing interest [69–72]. In [69], the parameters of the uplink channels, such as the angles of arrival and path delays were estimated and used to construct the downlink channels at an adjacent frequency band. This frequency extrapolation concept was further studied in [70] where lower bounds on the mean squared error of the extrapolated channels were derived. On a relevant line of work, [71, 72] proposed to leverage the channel knowledge at one frequency band (sub-6 GHz) to reduce the training overhead associated with design the mmWave beamforming vectors, utilizing the spatial correlation between the two frequency bands. A common feature of all the prior work in [69–72] is the requirement to first estimate some spatial knowledge (such as the angles of arrival) about the channels in one frequency band and then leverage this knowledge in the other frequency band. This channel parameter estimation process, however, is fundamentally limited by the system and hardware capability in resolving these channel parameters, which highly affects the quality of the extrapolated channels.

The general idea of using some knowledge about the sub-6 GHz channels to aid the system and network operation at mmWave is motivated by the spatial correlation between the two bands, which has been verified through experimental measurements [88–90]. On the network perspective, [88] proposed a network architecture that leveraged the spatial correlation between sub-6 GHz and mmWave bands for traffic scheduling and training overhead reduction. In [57], a dual connectivity protocol was developed that relies on a local coordinator to hand over the users between the two bands to avoid link failures. Leveraging deep learning, [91, 92] proposed strategies that learn the correlation between the sub-6 GH and mmWave bands and exploit that for selecting the communication band or handing over the users from one band to the other. While the work in [57, 88, 91–93] is relevant, it does not target predicting the mmWave beams or blockages using sub-6 GHz channels, which is the goal of this paper. Other prior work [62, 94] leveraged machine learning to identify current link status, in terms of being LOS or None-LOS (NLOS). This work, however, focuses on sub-6GHz systems and incurs certain limitations on the system operation.

To reduce the mmWave beam training overhead, [90] designed a novel algorithmic framework to leverage the sub-6 GHz spatial information in estimating the candidate mmWave beam directions. The feasibility of this solution was also studied in [90] using a proof-of-concept prototype. This solution, however, was mainly limited to detecting the LOS mmWave direction. In [71], the spatial information from sub-6 GHz was used to guide the compressive sensing based beam selection at mmWave bands and reduce the beam search overhead. With the same goal, [95] proposed an approach that constructs the mmWave channel covariance using the spatial characteristics extracted from the sub-6 GHz band. This mmWave covariance knowledge can then be exploited to reduce the training overhead associated with the design of the analog or hybrid analog/digital precoding matrices.

While the interesting solutions in [71, 90, 95] have the potential of reducing the search space of the mmWave beams, they share the following common limitations. First, the solutions in [71, 90, 95] generally rely on the approach of estimating some spatial parameters, such as the angular characteristics and path gains, at the sub-6 GHz band and then leverage them at mmWave. This makes their performance very sensitive to the parameters estimation error at the low-frequency bands. Also, this approach does not incorporate how the materials' dielectric coefficients, such as the reflection/scattering coefficients, differ in the two bands, which could be critical for the accurate modeling of the mmWave signal propagation. Further, the solutions in [71, 95] still require relatively large beam training overhead at the mmWave band, which scales with the number of antennas. Finally, and to the best of our knowledge, no prior work has provided any theoretical guarantees on using the sub-6 GHz channels to directly find the *optimal* mmWave beams or detect the mmWave blockages.

Notation: The following notation is used throughout this chapter: $\mathbf{A}$ is a matrix, $\mathbf{a}$ is a vector, a is a scalar, $\mathcal{A}$ is a set of scalars, and $\boldsymbol{\mathcal{A}}$ is a set of vectors. $\|\mathbf{a}\|_p$ is the p-norm of $\mathbf{a}$. $|\mathbf{A}|$ is the determinant of $\mathbf{A}$, whereas $\mathbf{A}^T$, $\mathbf{A}^{-1}$ are its transpose and inverse. $\mathbf{I}$ is the identity matrix. $\mathcal{N}(\mathbf{m}, \mathbf{R})$ is a complex Gaussian random vector with mean $\mathbf{m}$ and covariance $\mathbf{R}$.

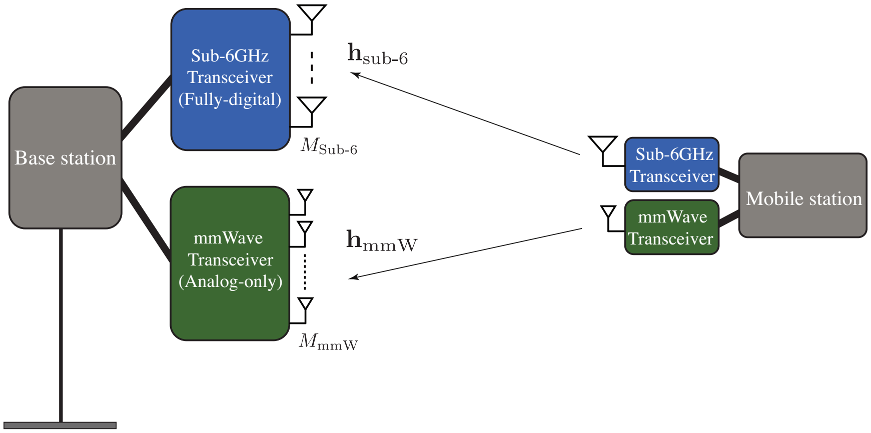

Base station
Sub-6GHz Transceiver (Fully-digital)
mmWave Transceiver (Analog-only)
$M_{\text{Sub-6}}$
M_{mmW}
$\mathbf{h}_{\text{sub-6}}$
$\mathbf{h}_{\text{mmW}}$
Sub-6GHz Transceiver
mmWave Transceiver
Mobile station

Figure 3.1: The Adopted System Model Where a Base Station and a Mobile User Communicate over Both Sub-6GHz and mmWave Bands. The Basestation and Mobile User Are Assumed to Employ Co-located Sub-6GHz and mmWave Arrays.

3.3 System and Channel Models

Consider the system model in Fig. 3.1 where a base station (BS) is communicating with one mobile user. The BS is assumed to employ two transceivers; one is working at sub-6GHz and employs $M_{\text{sub-6}}$ antennas, and the other is operating at a mmWave frequency band and adopts an M_{mmW}-element antenna array. For simplicity, we assume that the two antenna arrays belonging to the mmWave and sub-6GHz transceivers are co-located. As will be discussed in Section 3.5, however, the proposed concepts in this chapter can be extended to other setups with separated and distributed arrays. The mobile user is assumed to employ a single antenna at both mmWave and sub-6GHz bands. Note that the assumption of employing both sub-6GHz and mmWave transceivers at the base station and mobile user is based on the features of future wireless networks that will simultansouly operate at both sub-6GHz and mmWave frequency bands [85][86]. Further, it is important to highlight here that while we focus on point-to-point channels in this initial work, extending the proposed approaches to multi-user settings is a very interesting direction for future research. Next, we summarize the system operation and the adopted channel model.

System Operation: In this chapter, we consider a system operation where the uplink signaling happens at the sub-6GHz band while the downlink data transmission occurs at the mmWave band. If $\mathbf{h}_{\text{sub-6}}[k] \in \mathbb{C}^{M_{\text{sub-6}} \times 1}$ denotes the uplink channel vector from the mobile user to the sub-6GHz BS array at the kth subcarrier, $k = 1, ..., K$, then the uplink received signal at the BS sub-6GHz array can be written as

$$\mathbf{y}_{\text{sub-6}}[k] = \mathbf{h}_{\text{sub-6}}[k] s_{\text{p}}[k] + \mathbf{n}_{\text{sub-6}}[k], \tag{3.1}$$

where $s_{\text{p}}[k]$ represents the uplink pilot signal that satisfies $\mathbb{E}\left|s_{\text{p}}[k]\right|^2 = \frac{P_{\text{sub-6}}}{K}$, with $P_{\text{sub-6}}$ denoting the uplink transmit power from the mobile user. The vector $\mathbf{n}_{\text{sub-6}}[k] \sim \mathcal{N}_{\mathbb{C}}\left(\mathbf{0}, \sigma^2 \boldsymbol{I}\right)$ is the receive noise at the BS sub-6GHz array. The sub-6GHz transceiver is assumed to employ a fully-digital architecture, which allows for the channel estimation process to be done in the baseband.

For the downlink transmission, the BS employs the mmWave transceiver. Due to the large number of antennas and the high cost and power consumption of the RF chains at the mmWave frequency bands, the mmWave transceivers normally employ analog-only or hybrid analog digital architectures [40, 96]. Following that, the mmWave transceiver is assumed to adopt an analog-only architecture with one RF chain and M_{mmW} phase shifters. Extending the proposed solutions to more advanced architectures, such as hybrid analog/digital architectures, is both important and interesting for future research. This could potentially leverage the recent neural network architecture developed specifically for hybrid analog/digital architectures [96]. If $\mathbf{f} \in \mathbb{C}^{M_{\text{mmW}} \times 1}$ denotes the downlink beamforming vector, then the received signal at the mobile user can then be expressed as

$$y_{\text{mmW}}[\bar{k}] = \mathbf{h}_{\text{mmW}}^T[\bar{k}] \mathbf{f} s_{\text{d}} + n_{\text{mmW}}[\bar{k}], \tag{3.2}$$

where $\mathbf{h}_{\text{mmW}}[\bar{k}] \in \mathbb{C}^{M_{\text{mmW}} \times 1}$ represents the uplink channel from the mobile user to the BS mmWave array at the $\bar{k}$th subcarrier, $\bar{k} = 1, 2, ..., \bar{K}$. Due to the hardware constraints on the mmWave analog beamforming vectors, these vectors are normally selected from quantized codebooks. Therefore, we assume that the beamforming vector $\mathbf{f}$ can take one of candidate values collected in the codebook $\mathcal{F}$, i.e., $\mathbf{f} \in \mathcal{F}$, with cardinality $|\mathcal{F}| = N_{\mathrm{CB}}$.

Channel Model: This chapter adopts a geometric (physical) channel model for the sub-6GHz and mmWave channels [40]. With this model, the mmWave channel (and similarly the sub-6GHz channel) can be written as

$$\mathbf{h}_{\mathrm{mmW}}[k] = \sum_{d=0}^{D_{\mathrm{c}}-1} \sum_{\ell=1}^{L} \alpha_\ell e^{-j\frac{2\pi k}{K}d} p\left(dT_{\mathrm{S}} - \tau_\ell\right) \mathbf{a}\left(\theta_\ell, \phi_\ell\right), \tag{3.3}$$

where L is number of channel paths, $\alpha_\ell, \tau_\ell, \theta_\ell, \phi_\ell$ are the path gains (including the path-loss), the delay, the azimuth angle of arrival (AoA), and elevation AoA, respectively, of the ℓth channel path. T_{S} represents the sampling time while D_{c} denotes the cyclic prefix length (assuming that the maximum delay is less than $D_{\mathrm{c}}T_{\mathrm{S}}$). Note that the advantage of the physical channel model is its ability to capture the physical characteristics of the signal propagation including the dependence on the environment geometry, materials, frequency band, etc., which is crucial for our machine learning based beam and blockage prediction approaches. The parameters of the geometric channel models, such as the angles of arrival and path gains, will be obtained using accurate 3D ray-tracing simulations, as will be discussed in detail in Section 3.8.

3.4 Problem Definition

Adopting the dual-band system model described in Section 3.3, the objective of this chapter is to leverage the uplink channel knowledge at sub-6GHz band to enhance the achievable rate and reliability of the downlink mmWave link. More specifically, we focus on two important problems: (i) how can the uplink sub-6GHz channel be exploited to find the optimal downlink mmWave beamforming vector that maximizes the achievable rate and (ii) how can the knowledge of the uplink sub-6GHz channel be used to infer whether or not the line-of-sight link to the mobile user is blocked. Next, we formulate these two problems.

Problem 1: Beam Prediction Consider the system and channel models in Section 3.3, the downlink achievable rate for a mmWave channel $\mathbf{h}_{\text{mmW}}$ and a beamforming vector $\mathbf{f}$ is written as

$$R\left(\left\{\mathbf{h}_{\text{mmW}}[\bar{k}]\right\}, \mathbf{f}\right) = \sum_{\bar{k}=1}^{\bar{K}} \log_2\left(1 + \mathsf{SNR}\left|\mathbf{h}_{\text{mmW}}[\bar{k}]^T\mathbf{f}\right|^2\right), \tag{3.4}$$

with the per-subcarrier SNR defined as $\mathsf{SNR} = \frac{P_{\text{mmW}}}{K\sigma^2_{\text{mmW}}}$. The optimal beamforming vector $\mathbf{f}^\star$ that maximizes $R\left(\left\{\mathbf{h}_{\text{mmW}}[\bar{k}]\right\}, \mathbf{f}\right)$ is given by the exhaustive search

$$\mathbf{f}^\star = \underset{\mathbf{f}\in\mathcal{F}}{\text{argmax}} \quad R\left(\left\{\mathbf{h}_{\text{mmW}}[\bar{k}]\right\}, \mathbf{f}\right), \tag{3.5}$$

yielding the optimal rate $R^\star\left(\left\{\mathbf{h}_{\text{mmW}}[\bar{k}]\right\}\right)$. For ease of exposition, we drop the sub-carrier indices in the rest of the chapter; i.e., we will use $\mathbf{h}_{\text{mmW}}$ and $\mathbf{h}_{\text{sub-6}}$ to mean $\{\mathbf{h}_{\text{mmW}}[\bar{k}]\}$ and $\{\mathbf{h}_{\text{sub-6}}[k]\}$. It is important to note here that the beamforming vector $\mathbf{f}$ is assumed to be implemented in the analog/RF domain as discussed in Section 3.3. Therefore, the same beamforming vector is applied to all the subcarriers. Further, this beamforming vector can only be selected from the codebook $\mathcal{F}$. These constraints on the beamforming vector $\mathbf{f}$ renders the optimization problem in (3.5) non-convex, with the optimal solution only found via exhaustive search. Performing this search, however, requires either estimating the mmWave channel $\mathbf{h}_{\text{mmW}}$ or an online exhaustive beam training, both of which require large training overhead. To reduce (or eliminate) this training overhead, the objective of this work is to exploit the sub-6GHz channels $\mathbf{h}_{\text{sub-6}}$ to decide on the optimal beamforming vector. If $\hat{\mathbf{f}} \in \mathcal{F}$ denotes the predicted beamforming vector based on the knowledge of $\mathbf{h}_{\text{sub-6}}$, then the first objective of this work is to maximize the success probability in predicting optimal beamforming vector $\mathbf{f}^\star$, defined as

$$\kappa_1 = \mathbb{P}\left(\hat{\mathbf{f}} = \mathbf{f}^\star \left|\{\mathbf{h}_{\text{sub-6}}\}\right.\right). \tag{3.6}$$

Problem 2: Blockage Prediction The sensitivity of mmWave signals to blockages can critically impact the reliability of the high frequency systems. If the status of the link in terms of line-of-sight (LOS) (unblocked) or non-LOS (blocked) can be predicted, this can enhance the system reliability via, for example, proactively handing over the user to another base station/access point [45]. In this work, we explore the possibility of using sub-6GHz channels to predict whether the link connecting the base station and the user is blocked (NLOS) or unblocked (LOS). let $s \in \mathcal{B}$ denote the correct (ground-truth) blocked/unblocked status of the communication link between the base station and the user, with $s = 1$ indicating a blocked link and $s = 0$ indicating an unblocked link. If $\hat{s}$ is the predicted link status using the sub-6GHz channel knowledge, then the objective of the second problem in this work is to maximize the success probability of predicting the correct blockage status defined as

$$\kappa_2 = \mathbb{P}\left(\hat{s} = s \,|\{\mathbf{h}_{\text{sub-6}}\}\right). \tag{3.7}$$

In the next two sections, we present our proposed solutions that leverage machine learning tools to address the formulated mmWave beam and blockage prediction problems.

3.5 Predicting mmWave Beams Using Sub-6GHz Channels

Enabling the high data rate gains at mmWave communication systems requires the deployment of large antenna arrays at the transmitters and/or the receivers. Finding the best beamforming vectors $\mathbf{f}^\star$ for these arrays is normally done through an exhaustive search over a large codebook of candidate beams, which is associated with large training overhead [97][98]. In this section, we investigate the feasibility of exploiting sub-6GHz channels to predict/infer mmWave beams. If this is possible, we can expect dramatic savings in the mmWave beam training overhead as sub-6GHz channels can be easily estimated with a few pilots; ideally one pilot is required to estimate the uplink sub-6GHz channels. Leveraging sub-6GHz channels to predict mmWave beams is also motivated by the fact that future wireless networks, such as 5G, will likely be dual-band—operating at both sub-6GHz and mmWave bands. Next, we will first reveal in Section 3.5.1 that for any given environment, there exist a deterministic mapping from sub-6GHz channels to the optimal mmWave beams under certain conditions. Then, we will show in Section 3.5.2 how deep learning models can be exploited to predict the optimal mmWave beams using sub-6GHz channels with a probability of error that can be made arbitrarily small.

3.5.1 Mapping Sub-6GHz Channels to mmWave Beams

This section establishes the theoretical foundation for our proposed solution that predicts mmWave beams using sub-6GHz channels. More specifically, we will prove that, under certain condition, there exists a deterministic mapping from sub-6GHz channels to mmWave channels and beams. This proof extends the channel mapping concept that we proposed in Chapter 2. First, consider the dual-band system and channel models described in Section 3.3. Let $\mathcal{X} = \{\mathbf{x}_u\}_{u=1}^{U}$ represent the set of candidate user positions, with $\mathbf{x}_u \in \mathbb{R}^3$ denoting the position of user u and U is the total number of users. Further, let $\mathbf{h}_{\text{sub-6}}^{u} \in \mathbb{C}^{M_{\text{sub-6}} \times 1}, \mathbf{h}_{\text{mmW}}^{u} \in \mathbb{C}^{M_{\text{mmW}} \times 1}$ denote the channels from user u to the sub-6GHz and mmWave antenna arrays, respectively, and $\mathcal{S}_{\text{sub-6}} = \{\mathbf{h}_{\text{sub-6}}^{u}\}_{u=1}^{U}$, $\mathcal{S}_{\text{mmW}} = \{\mathbf{h}_{\text{mmW}}^{u}\}_{u=1}^{U}$ are the sets of all candidate user sub-6GHz and mmWave channels. Now, we can define the following mapping functions, $\boldsymbol{\psi}_{\text{sub-6}}, \boldsymbol{\psi}_{\text{mmW}}$, from the set of candidate positions $\mathcal{X}$ to the corresponding $S_{\text{sub-6}}$ and S_{mmW}:

$$\boldsymbol{\psi}_{\text{sub-6}} : \mathbf{x} \in \mathcal{X} \rightarrow \mathbf{h}_{\text{sub-6}} \in \mathcal{S}_{\text{sub-6}}, \tag{3.8}$$

$$\boldsymbol{\psi}_{\text{mmW}} : \mathbf{x} \in \mathcal{X} \rightarrow \mathbf{h}_{\text{mmW}} \in \mathcal{S}_{\text{mmW}}. \tag{3.9}$$

These two functions represent the same wireless environment, and hence, they encode the geometry of that environment and its propagation characteristics.

Further, for any given mmWave channel $\mathbf{h}_{\text{mmW}}^{u}$ and beamforming vector $\mathbf{f}_n \in \mathcal{F}$, the achievable rate $R\left(\mathbf{h}_{\text{mmW}}^{u}, \mathbf{f}_{n}\right) \in \mathbb{R}^{+}$ is calculated using (3.4). Based on that, we define the position to achievable rate mapping functions $\mathbf{g}^{n}(.), n=1,2,...,|\mathcal{F}|$ as

$$\mathbf{g}^{n}: \mathbf{x}_{u} \in \mathcal{X} \rightarrow R\left(\mathbf{h}_{\text{mmW}}^{u}, \mathbf{f}_{n}\right), \quad n=1,2,...,|\mathcal{F}| . \tag{3.10}$$

Note that the existence of these mapping functions $\mathbf{g}^{n}, \forall n$ follows directly from the existence of the position to mmWave channel mapping, $\boldsymbol{\psi}_{\text{mmW}}$, and the deterministic achievable rate function in (3.4) that relates the mmWave channels and the achievable rates with the $|\mathcal{F}|$ beamforming vectors. Next, we Chapter 2 and adopt the following bijectiveness assumption of the mapping function $\boldsymbol{\psi}_{\text{sub-6}}$ that maps the positions to sub-6GHz channels.

Assumption 2 *The position to sub-6GHz channel mapping function, $\boldsymbol{\psi}_{\text{sub-6}}$, is bijective*[1].

Assumption 2 means that any two user positions in $\mathcal{X}$ have different sub-6GHz channel vectors, i.e., two positions can not result in the same sub-6GHz channels. This bijectiveness assumption depends on the number of antennas, the array geometry, the number of paths, and the surrounding environment among other factors. It is possible, however, to show that a few antennas could be sufficient to make this bijectiveness assumption satisfied with very high probability in practical scenarios [74]. In addition, our work in [17] has shown that even with aggressive quantization schemes, the assumption could hold up given that large number of antennas is used. A deeper study on the practical conditions under which this bijectiveness assumption could be violated is particularly needed. The importance of this bijectiveness assumption (in Assumption 2) is that it guarantees the existence of the inverse mapping $\boldsymbol{\psi}_{\textit{sub-6}}^{-1}(.)$ that maps the sub-6GHz channels in $\mathcal{S}_{\text{sub-6}}$ to the corresponding positions in $\mathcal{X}$. Next, we present the main proposition on the existence of the mapping from the sub-6GHz channels to the optimal mmWave beams.

Proposition 2 *For any given communication environment, and under Assumption 2, there*

[1] A function is said to be bijective when it is both surjective (onto) and injective (one-to-one)

exists a set of sub-6GHz to achievable rate mapping functions $\mathcal{G} = \{\mathbf{\Phi}^n_{\text{sub-6}}\}_{n=1}^{|\mathcal{F}|}$ *that equal*

$$\mathbf{\Phi}^n_{\text{sub-6}} : \mathbf{h}_{\text{sub-6}} \in \mathcal{S}_{\textit{sub-6}} \rightarrow R\left(\mathbf{h}_{\text{mmW}}, \mathbf{f}_n\right), \;\; n = 1, 2, ..., |\mathcal{F}|, \tag{3.11}$$

and the optimal mmWave beamforming vector $\mathbf{f}^\star$ *for user* u *is obtained using*

$$n^\star = \underset{n \in \{1,2,...,|\mathcal{F}|\}}{\text{argmax}} \quad \mathbf{\Phi}^n_{\text{sub-6}}\left(\mathbf{h}^u_{\text{sub-6}}\right). \tag{3.12}$$

such that $\mathbf{f}^\star = \mathbf{f}_{n^\star}$.

Proof: The proof follows from the existence of the sub-6GHz channel to position mapping function $\boldsymbol{\psi}^{-1}_{\text{sub-6}}(.)$ and the existence of the position to mmWave achievable rate mapping functions $\text{g}^n(.)$. This leads to the existence of the composite mapping functions $\mathbf{\Phi}^n_{\text{sub-6}}$ since the co-domain of $\boldsymbol{\psi}^{-1}_{\text{sub-6}}(.)$ is the same as the domain of $\text{g}^n(.)$, and both equal to $\mathcal{X}$. Finally, since the mapping functions $\mathbf{\Phi}^n_{\text{sub-6}}(.)$ result in the achievable rates with the candidate beams, the optimal beamforming vector $\mathbf{f}^\star$ is found via the exhaustive search in (3.5). □

Proposition 2 shows that, under certain conditions, there exist mapping functions $\mathbf{\Phi}^n_{\text{sub-6}}, \forall n$, that can be leveraged to predict the optimal mmWave beam using sub-6GHz channels. Despite the existence of this mapping, though, it is very hard to identify it using classical (non machine learning) solutions as this mapping functions are normally very hard to be characterized analytically. This motivates utilizing deep learning to learn these non-trivial mapping functions.

3.5.2 Deep Learning Based Beam Prediction

Deep learning models have the interesting capability of learning and approximating non-trivial functions. Leveraging these models can effectively enable the prediction of the optimal mmWave beams directly from the knowledge of the sub-6GHz channels with an arbitrarily small error. Next, we use the universal approximation theory [78], to prove that.

Proposition 3 *Let $\mathbf{\Pi}_N^n(.)$ represent the output of a dense neural network that consists of a single hidden layer of N nodes. Then, for any $\epsilon_n > 0$, and a continuous achievable rate mapping function $\mathbf{\Phi}_{\text{sub-6}}^n(.)$, there exists a positive constant N such as*

$$\sup_{\mathbf{h} \in \{\mathbf{h}_{\text{sub-6}^u}\}} |\mathbf{\Pi}_N(\mathbf{h}, \Omega) - \mathbf{\Phi}_{\text{sub-6}}^n(\mathbf{h})| < \epsilon_n, \tag{3.13}$$

where Ω denotes the set of neural network parameters.

Proof: Proposition 3 follows directly from the universal approximation theorem [78, Theorem 2.2] by noticing that the set of sub-6GHz channels $\mathcal{S}_{\text{sub-6}}$ is a compact set since it is closed and bounded. □

Since the function $\mathbf{\Phi}_{\text{sub-6}}^n(\mathbf{h})$ maps the sub-6GHz channels to the mmWave achievable rate using the beamforming vector $\mathbf{f}_n \in \mathcal{F}$, Proposition 3 simply means that using a large enough neural network, we can predict the mmWave achievable rate $\hat{R}(\mathbf{h}_{\text{mmW}}, \mathbf{f}_n)$ associated with every beam $\mathbf{f}_n \in \mathcal{F}$ with arbitrarily small error. Next, we make an assumption on the codebook $\mathcal{F}$ before presenting the main result in Corollary 4.

Assumption 3 *The mmWave beamforming codebook $\mathcal{F}$ satisfies the following condition*

$$R(\mathbf{h}_{\text{mmW}}, \mathbf{f}^\star) - R(\mathbf{h}_{\text{mmW}}, \mathbf{f}_n) > 0, \ \forall \mathbf{h}_{\text{mmW}} \in \{\mathbf{h}_{\text{mmW}}^u\}, \tag{3.14}$$

where $\mathbf{f}_n, \mathbf{f}^\star \in \mathcal{F}$ and $\mathbf{f}_n \neq \mathbf{f}^\star$.

Assumption 2 simply means that there is only one optimal beamforming codeword for any channel $\mathbf{h}_{\text{mmW}} \in \mathcal{S}_{\text{mmW}}$. It is important to note here that while we need this assumption to prove the result in the following Corollary, violating this condition on the codebook $\mathcal{F}$ leads to the trivial case where two beamforming vectors can achieve exactly the optimal rate. Next, we present Corollary 4 that establishes the feasibility of predicting the optimal mmWave beams using sub-6GHz channels via deep neural networks.

Proposition 4 *Let $\mathbf{\Pi}_N^n(.), n = 1, 2, ..., |\mathcal{F}|$ represent the output of a dense neural network that consists of a single hidden layer of N nodes. Further, define the predicted beamforming vector $\hat{\mathbf{f}} = \mathbf{f}_{\hat{n}} \in \mathcal{F}$ with $\hat{n} = \underset{n=1,2,...,|\mathcal{F}|}{\text{argmax}} \quad \mathbf{\Pi}_N^n(.)$. Then, for any $\epsilon > 0$, and continuous achievable rate mapping functions $\mathbf{\Phi}_{\text{sub-6}}^n(.)\,, n = 1, 2, ..., |\mathcal{F}|$, there exists s positive constant N large enough such as*

$$\kappa_1 = \mathbb{P}\left(\hat{\mathbf{f}} = \mathbf{f}^\star \left|\mathbf{h}_{\text{sub-6}}\right.\right) > 1 - \epsilon.$$

Proof: The success probability in predicting the optimal mmWave beam $\mathbf{f}^\star$ using the sub-6GHz channels can be written as

$$\kappa_1 = \mathbb{P}\left(\hat{\mathbf{f}} = \mathbf{f}^\star \left|\mathbf{h}_{\text{sub-6}}\right.\right) \tag{3.15}$$

$$= 1 - \mathbb{P}\left(\hat{\mathbf{f}} \neq \mathbf{f}^\star \left|\mathbf{h}_{\text{sub-6}}\right.\right). \tag{3.16}$$

Since the predicted beam $\hat{\mathbf{f}}$ is obtained from the outputs of the $|\mathcal{F}|$ neural networks by applying $\hat{n} = \underset{n=1,2,...,|\mathcal{F}|}{\text{argmax}} \mathbf{\Pi}_N^n(.)$ and setting $\hat{\mathbf{f}}$ as the $\hat{n}$th beam in the codebook $\mathcal{F}$, then κ_1 can be expressed in terms of $\mathbf{\Pi}_N^n(.)$ as

$$\kappa_1 = 1 - \mathbb{P}\left(\mathbf{\Pi}_N^{\hat{n}}(\mathbf{h}_{\text{sub-6}}) > \mathbf{\Pi}_N^{n^\star}(\mathbf{h}_{\text{sub-6}})\right) \tag{3.17}$$

Now, given Proposition 3, we reach

$$\kappa_1 \geq 1 - \mathbb{P}\left(\mathbf{\Phi}^{\hat{n}} + \epsilon_{\hat{n}} > \mathbf{\Phi}^{n^\star} - \epsilon_{n^\star}\right) \tag{3.18}$$

$$= 1 - \mathbb{P}\left(\mathbf{\Phi}^{n^\star} - \mathbf{\Phi}^{\hat{n}} < \epsilon_{n^\star} + \epsilon_{\hat{n}}\right) \tag{3.19}$$

$$\overset{(a)}{\geq} 1 - \mathbb{P}\left(\mathbf{\Phi}^{n^\star} - \mathbf{\Phi}^{\hat{n}} < 2\bar{\epsilon}\right) \tag{3.20}$$

where (a) follows by defining $\bar{\epsilon} = \max_{n=1,2,...,|\mathcal{F}|} \epsilon_n$. Now, given Proposition 3 and Assumption 3, for any $\epsilon > 0$, there exists $\bar{\epsilon}$, such that $\mathbb{P}\left(\mathbf{\Phi}^{n^\star} - \mathbf{\Phi}^{\hat{n}} < 2\bar{\epsilon}_{n^\star}\right) < \epsilon$, which concludes the proof.

□

Corollary 4 is very interesting as it proves that it is possible to use neural networks to predict the *optimal* mmWave beamforming vector directly from the knowledge of the sub-6GHz channels once the achievable rate mapping functions, $\mathbf{\Phi}^{n}_{\text{sub-6}}\left(.\right), n = 1, 2, ..., |\mathcal{F}|$, exist. Further, we know from Proposition 2 that the existence of these mapping functions for any given environment requires only that the mapping from the candidate set of positions to the sub-6GHz channels is bijective – a condition that is achievable with high probability as we discussed earlier in Section 3.5.1.

Proposed Deep Learning Based System Operation: Consider the dual-band system model in Section 3.3. The proposed system operation that exploits deep learning to predict the optimal mmWave beam directly from the sub-6GHz channels operates in the following two phases:

- **Deep Learning Training Phase:** In this phase, the dual-band communication system operates as if there is no machine learning: For every coherence time, the uplink sub-6GHz channel is estimated requiring only one uplink pilot, and a search over the beams of the codebook $\mathcal{F}$ is done for the mmWave downlink to identify the best beamforming vector—the reason for this will be explained in Section 3.7. Let $n^{\star}_{u}$ denote the index of the best beamforming vector $\mathbf{f}_{n^{\star}} \in \mathcal{F}$ for user u. Then, at every coherence time, one new data point $(\mathbf{h}^{u}_{\text{sub-6}}, n^{\star}_{u})$ is added to the deep learning dataset. After collecting large number of data points, we use this dataset to train the deep learning model, which will be described in detail in Section 3.7.

- **Deep Learning Deployment Phase:** Once the deep learning model is trained, the base station uses it to directly predict the optimal mmWave beam using the sub-6GHz channels. More specifically, this phase requires the user to send only one uplink pilot to estimate the sub-6GHz channels and this channel is passed to the deep learning model which predicts which mmWave beam should be used for the downlink mmWave data transmission. This saves all the training overhead associated with the mmWave exhaustive beam training process.

It is important to note here that **the proposed deep learning based system operation has almost no learning overhead in terms of the system time-frequency resources**. That is because the mmWave beam training will typically be performed anyway in the classical system operation (that do not use machine learning) to figure out the best beamforming direction. This means that the dataset collection process and the deep learning training are done without affecting the classical mmWave system operation. Hence, even if a large dataset needs to be collected to capture the dynamics in the environment, that is feasible because it does not interfere with the classical system operation. It should be noted here, however, that for applications where the collection of large datasets is not feasible, synthetic (simulated) data could be used to pre-train the model; and a small sample of real data–collected in the fashion discussed above–could be, then, used to fine-tune the pre-trained model.

Practical Challenges: As shown in this section, for any given static environment, once the mapping from the candidate positions to the sub-6GHz channels is bijective (one-to-one), the sub-6GHz channels can be exploited to directly predict the optimal mmWave beams with a very high success probability. In practice, however, there are a few factors that can add some probabilistic error to this beam prediction such as the measurement noise, the phase noise, and the dynamic scatterers in the environment. These factors can make the position-to-channel mapping not perfectly bijective or create sub-6GHz channels that are different than those experienced before by the neural networks. In Section 3.8, we will evaluate the impact of some of these practical considerations on the beam prediction performance.

3.6 Predicting mmWave Blockages Using Sub-6GHz Channels

The reliability of the communication links is one of the main challenges for mmWave systems. This is mainly because of the sensitivity of mmWave signals to blockages, which can result in a sudden drop in the SNR if the line-of-sight (LOS) path is obstructed. With this motivation, [45] proposed to leverage machine learning to learn the mobility patterns of the transmitters, receiver, or scatterers, and hence predict blockages before they actually block the LOS path. This can enable the network to act proactively, for example by handing over the communication session to another base station, before the session is disconnected. In this chapter, we focus on a different but equally important problem which is the ability of the dual-band base stations to use the sub-6GHz channels to decide whether or not the mmWave LOS link is blocked. This knowledge can potentially help the BS in adapting its transmission strategy accordingly by, for example, changing the transmit power and modulation/coding scheme or handing off the communication session to the sub-6GHz band. In Section 3.6.1, we will investigate the conditions under which the sub-6GHz can indicate the LOS blockage/no-blockage status. Then, we show in Section 3.6.2 that this capability can be implemented using deep neural networks.

3.6.1 Mapping Sub-6GHz Channels to Link Blockages

Consider the system model in Section 3.3 with co-located sub-6GHz and mmWave arrays at the base station. Let $\mathcal{X} = \{\mathbf{x}_u\}$ represent the set of candidate user locations. To simplify the analysis in this section, we make the following assumption

Assumption 4 *For all the users in $\mathcal{X}$, if a blockage obstructs the LOS path to the mmWave array, it also obstructs the LOS path to the sub-6GHz array.*

Note that this assumption is typically satisfied in practice since the sub-6GHz and mmWave arrays are co-located. It is also worth mentioning here that while obstructing the mmWave

LOS link may completely block the link (due to the high penetration loss at mmWave), the obstruction of the sub-6GHz LOS ray will likely only reduce its power without a complete blockage. Our analysis, however, is general and independent of whether the LOS rays are completely of partially blocked. Now, define $s_u \in \mathcal{S} = \{0, 1\}$ as the blockage status of user u, with $s_u = 0$ and $s_u = 1$ indicating that the LOS path between user u and the BS is, respectively, unblocked or blocked. For a given environment, let $\mathbf{h}^u_{\text{sub-6, B}}$ denote the sub-6GHz channel of user u when the LOS path is obstructed/blocked and $\mathbf{h}^u_{\text{sub-6, UB}}$ denote the channel when the LOS path is not blocked. Further, let $\mathcal{H}_{\text{B}} = \{\mathbf{h}^u_{\text{sub-6, B}}\}$ and $\mathcal{H}_{\text{UB}} = \{\mathbf{h}^u_{\text{sub-6, UB}}\}$ represent the blocked and unblocked channel sets. Next, we define the mapping function Ψ that maps the user position and blockage status to a sub-6GHz channel.

$$\mathbf{\Psi} : \mathcal{X} \times \mathcal{S} \to \mathcal{H}, \tag{3.21}$$

where $\mathcal{X} \times \mathcal{S}$ is the Cartesian product of the user position and blockage status sets, and $\mathcal{H}$ represent the set of all blocked and unblocked channels, i.e., $\mathcal{H} = \mathcal{H}_{\text{B}} \cup \mathcal{H}_{\text{UB}}$. In the following proposition, we state the condition under which the LOS blockage can be identified using the sub-6GHz channels.

Proposition 5 *For any given environment, if the mapping function* $\mathbf{\Psi}$ *is bijective, then there exists a continuous discriminant function* $f : \mathcal{H} \to 0, 1$ *such that*

$$\forall \mathbf{h} \in \mathcal{H}_B, f(\mathbf{h}) = 1, \quad \forall \mathbf{h} \in \mathcal{H}_{UB}, f(\mathbf{h}) = 0. \tag{3.22}$$

Proof: When the mapping $\mathbf{\Psi}$ is bijective, each $(\mathbf{x}_u, s_u)$ tuple has a unique channel, which yields disjoint blocked and unblocked channel sets, i.e., $\mathcal{H}_{\text{B}} \cap \mathcal{H}_{\text{UB}} = \phi$. This leads to the existence of the continuous discriminant function $f(.)$ using the Urysohn Lemma [99]. □

Note that the bijectiveness condition of the mapping function $\mathbf{\Psi}$ means that (i) every user position in $\mathcal{X}$ will yield two different channels for the LOS-obstructed or unobstructed cases and that (ii) these LOS-obstructed/unobstructed channels are different for all the users in $\mathcal{X}$. Similar to Assumption 2, the bijectiveness condition of $\boldsymbol{\Psi}$ is expected to be satisfied with high probability in multi-antenna systems, as will be shown in Section 3.8.6.

3.6.2 Deep Learning Based Blockage Prediction

Using sub-6GHz channel knowledge, Proposition 5 proves that it is possible to decide whether the LOS link between the base station and user is blocked or not under some conditions. The discriminating function that does this, however, is hard to be characterized analytically and may be highly non-linear given the nature of the complex channel vectors. Intuitively, deciding whether the LOS link is blocked or not requires some spatial and power analysis of the rays that construct the channels which is a non-trivial task. Motivated by these challenges, we propose to leverage the powerful learning capabilities of deep neural networks to learn this LOS blockage discriminating function. This is addressed by the following proposition.

Proposition 6 *Let $\boldsymbol{\Pi}_N^n(.), n = 1, 2$ represent the output of a dense neural network that consists of a single hidden layer of N nodes. Further, define the predicted blockage status of using this network as $\hat{s}_N$. If the conditions in Proposition 5 are satisfied, then for any $\epsilon > 0$, there exists s positive constant N large enough such as*

$$\kappa_2 = \mathbb{P}\left(\hat{s}_N = s \left|\mathbf{h}_{\text{sub-6}}\right.\right) > 1 - \epsilon. \tag{3.23}$$

Proof: The proof is similar to that in Corollary 4 and is omitted due to space limitation.
□

Practical Challenges: Proposition 6 highlights the interesting ability of neural networks in classifying the sub-6GHz channel to LOS blocked or unblocked classes. One important challenge in this application, however, is obtaining the ground-truth blocked/unblocked labels. Therefore, it is important to develop practical labeling techniques that construct the required labels for training the neural networks. In Section 3.8.6, we propose a labeling strategy based on analyzing the mmWave beam training results and evaluate its performance compared to the case when the ground-truth labels are available.

3.7 Deep Learning Model

In Sections 3.5.2 and 3.6.2, we proved theoretically how neural networks can enable the prediction of the mmWave beams and blockages using sub-6GHz channels. In this section, we will describe our specific design of the neural network architecture and the adopted learning model. Before we delve into the description of the proposed model, it is important to note that the two tasks we consider in this chapter, namely predicting the optimal mmWave beam and predicting the link blockage status, involve a selection from a pre-defined set of options–a beam codebook or a binary set of blocked/unblocked status. These problems have then a striking similarity with the well-known classification problem in machine learning [31]. Specifically, in the beam prediction problem, each sub-6GHz channel is mapped to one of $D = |\mathcal{F}|$ indices, where D is the size of discrete set of options. This could be viewed as a classification problem where each beam index represents a class, and the job of the learning model is to learn how to *classify* channels into beam indices. In the recent years, deep-architecture neural networks have performed exceedingly well in handling classification problems [5][6], among other things. Motivated by these results and by the conclusions of Sections 3.5.2 and 3.6.2, we design a deep neural network model to address the mmWave beam and blockage prediction problems.

3.7.1 Deep Neural Network Design

The first step in designing a neural network is the choice of the network type, which should be based on the nature of the problem and the desired role of the model. For our beam/blockage prediction problems, the objective is to learn how to map the sub-6GHz channel vectors to a real-valued D-dimensional vector $\mathbf{p}$, where D is either the codebook size, $|\mathcal{F}|$, or 2 for the blockage status. For this objective, and motivated by the universal approximation results in Sections 3.5.2 and 3.6.2, we adopt a Multi-Layer Perceptron (MLP) network, which comprises a sequence of non-linear vector transformations [100]. The proposed network architecture has two main sections, namely the base network and the task-specific layer.

Base Network: The beam and blockage predictions are both posed as classification problems and both share the same input data (sub-6GHz channels). Therefore, to reduce the computational burden of the training process, we propose to have a common neural network architecture for the two problems, which, as will be shown shortly, enables leveraging *transfer learning* to reduce the training overhead. Based on that, a single *base* deep neural network is designed for the two prediction problems. This network comprises L_{NN} stacks of layers, each of which has a sequence of fully-connected with ReLU non-linearity and dropout layers, as illustrated in Figure 3.2. All fully-connected layers have the same breadth, M_{NN} neurons per layer.

Task-Specific Output Layer: The number of outputs in each prediction task (beam or blockage) differs as the number of classes changes; predicting a beam index means that there are $D = |\mathcal{F}|$ beam choices, while predicting blockage is a binary problem with $D = 2$ choices, *blocked* or *unblocked*. Hence, the base network is customized with an additional stack of layers that depends on the target task. For beam prediction, the final layer is designed to have a fully-connected layer with $D = |\mathcal{F}|$ neurons. It acts as a linear classifier that projects its M_{NN}-dimensional input feature vector onto a D-dimensional classification space. The projection is fed to a *Softmax* layer, which induces a probability distribution over all the available classes. Formally, it does so by computing the following formula for every element d in its input vector:

$$p_d = \frac{e^{z_d}}{\sum_{i=1}^{D} e^{z_i}}, \tag{3.24}$$

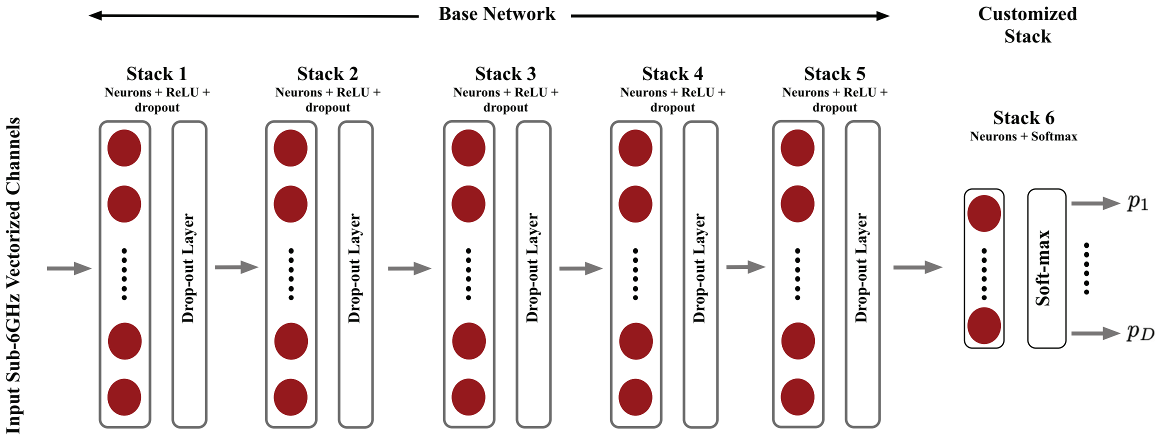


Figure 3.2: The Overall DNN Architecture. The First L_{NN} Stacks Comprising Multiple Fully-Connected, ReLU, and Dropout Layers form The Base Network. The Final Stack Represents The Customizable Output Layers. For Both Problems, It Comprises a Fully-Connected Layer Followed by a Soft-max. Their Size Depends of The Number of Classes in Each Task.

where $z_i, i = 1, ..., D$ is the ith element of the D-dimensional projection vector (input to the softmax), and p_d is the probability that the dth beamforming vector is the correct prediction–more on Softmax could be found in [73]. Finally, the index of the element with the highest probability is the index of the predicted beam-forming vector. For the blockage prediction task, a similar last stack is designed but with different dimensions. The classifier has $D = 2$ neurons, and the Softmax here produces two probabilities, namely blocked (p_1) and unblocked (p_2).

Transfer Learning: An interesting and advantageous characteristic of deep neural networks is their ability to exploit a learned function on a certain input data to perform another function on the same input data, which is referred to as *transfer learning*. In [101], it has been empirically shown that layers closer to the input learn generic features, i.e., those layers tend to learn the same mapping regardless of the task and final outputs of the neural network. However, as layers get farther away from the input and deeper into the network, features become more specific, i.e., they are more groomed to the task in question. Such empirical evidence suggests that reusing a trained network for a different task could provide an interesting boost in the network performance and help reduce the computational complexity associated with its training [101].

Now, given that both beam/blockage prediction problems could be faced by the same mmWave system, a resourceful way for good prediction performance in both cases is to apply transfer learning. As it will be discussed in Section 3.8, beam prediction is a more challenging problem than blockage prediction. This is mainly, but not exclusively, due to the large number of classes beam prediction has. Hence, the proposed training strategy in this work focuses on first training and testing the deep neural network for beam prediction. Once that cycle is done, the last stack of the trained network is replaced with that suitable to blockage prediction. Then, it undergoes another training and testing cycle (called fine-tuning) for the blockage prediction task. This offers faster convergence and improved performance compared to training from scratch for the blockage prediction task.

3.7.2 Learning Model

Our objective is to leverage the neural network architecture described in Section 3.7.1 to learn how to predict mmWave beams and blockages directly from the sub-6GHz channels. To achieve that, we adopt a supervised learning model that operates in two modes, a background training mode and a deployment mode. Next, we explain the two modes.

1. Background Training Mode: As described earlier in Section 3.5.2, the dual-band system operates as if there is no deep learning. It collects data points for the beam prediction dataset, $(\mathbf{h}^u_{\text{sub-6}}, n^\star_u)$, and, if the blockage status knowledge is available, it collects data points for the blockage prediction dataset, $(\mathbf{h}^u_{\text{sub-6}}, s_u)$. We will discuss how to obtain the blockage labels shortly. Both datasets needs to undergo pre-processing before being used for model training:

- **input normalization:** The sub-6GHz channels, which are the inputs to the neural network, are normalized using a global normalization factor. Let

$$\Delta = \sqrt{\frac{1}{N_{\text{train}}} \sum_{\forall u, \forall i, \forall k} \left| \left[\mathbf{h}^{u}_{\text{sub-6,k}} \right]_i \right|^2}, \tag{3.25}$$

  denote the global normalization factor where N_{train} is the total training samples, and $\left[\mathbf{h}^{u}_{\text{sub-6,k}} \right]_i$ is the ith element in the sub-6GHz channel vector of the kth subcarrier of user u. Then the sub-6GHz channels in the dataset $\mathcal{S}_{\text{sub-6}}$ are all normalized by Δ to have an average power of 1. Every normalized channel is decomposed into real and imaginary vectors that are stacked together to form a real-valued vector. Finally, all real-valued vectors of the K sub-6GHz subcarriers of a user u are stacked together to form a $(2 \times K \times M_{\text{sub-6}})$-dimensional vector, the input to the neural network. Writing the complex channel vector as a real-valued vector of the stacked real, imaginary, and subcarriers is to enable the implementation of real-valued computations of neural networks.

- **Labels construction:** The labels are modeled as D-dimensional one-hot vectors[2] indicating the class labels. For the beam prediction dataset, the one-hot vector for every sub-6GHz channel has 1 at the element that corresponds to the index of the optimal beamforming vector (which is calculated from (3.5)). For the blockage prediction task, the one-hot vectors are 2-dimensional with $[1, 0]$ for blocked and $[0, 1]$ for unblocked. In Section 3.8.6, we study the learning performance in two situations: (i) when the ground-truth blockage status is available and (ii) when the blockage status is estimated based on the angular distribution of receive power.

After preparing the dataset, the neural network model is trained to minimize the cross-

[2] One-hot vector refers to a binary vector where all elements are zero except for a single element with the value of one.

entropy loss function, L_{cross} defined as

$$L_{\text{cross}} = -\sum_{d=1}^{D} t_d \log_2(p_d), \tag{3.26}$$

where $\mathbf{t} = [t_1, ..., t_D]$ is the target one-hot vector and $\mathbf{p} = [p1, ..., p_D]$ is the network prediction. It is important to mention here that p_d represents the neural network predicted probability that the input sub-6GHz channel belongs to the dth class.

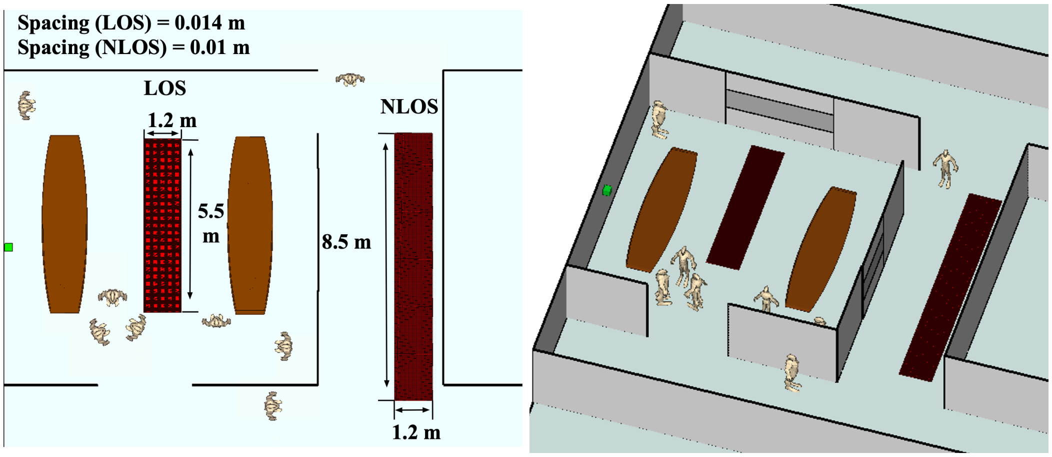
2. Deployment Mode: Once the neural network model is trained, it is then used to predict the mmWave beams and blockage status directly from the knowledge of the sub-6GHz channels. Please refer to Section 3.5.2 for more details.

3.8 Experimental Results

In this section, we evaluate the performance of the proposed mmWave beam and blockage prediction solutions using numerical simulations. First, we describe the adopted evaluation scenarios in Section 3.8.1. Then, we explain the construction of the deep learning dataset and neural network training process in Sections 3.8.2 and 3.8.4. Finally, we show and discuss the performance results of the sub-6GHz based mmWave beam and blockage prediction solutions in Sections 3.8.5 and 3.8.6.

3.8.1 Evaluation Scenarios

Two publicly available evaluation scenarios from the DeepMIMO dataset [68] are considered in the simulations. These scenarios are constructed using the 3D ray-tracing software Wireless InSite [87], which captures the channel dependence on the frequency. The first scenario is the LOS scenario ’O1’ that is available at two frequencies: ‘O1_28’ at 28GHz and ‘O1_3p5’ at 3.5GHz. It has a city street with multiple base station positioned at the sidewalks and users scattered along the street itself. The second scenario is the indoor mixed-user scenario ‘I2’ that is available also at two frequencies: ‘I2_2p4’ at 2.4 GHz and



(a) Top-view

(b) Perspective-view

Figure 3.3: Top and Perspective Views of The Second Scenario, I2_2p4 and I2_60. Both Show The Basestation Location Inside a Conference Room, Depicted As a Green Box. They Also Show The LOS and NLOS User Grids, and The Possible Scatterers and Blockages.

'I2_60' at 60 GHz. It has two base stations and two user grids, see Fig.3.3. For the LOS scenario, we adopt a single base station (BS 3) and equip it with two co-located uniform linear arrays (ULAs) at 28GHz and 3.5GHz. Similarly for the mixed-user scenario, we adopt a single base station (BS 2), and we equip with two co-located ULAs operating at 2.4 GHz and 60 GHz.

3.8.2 Dataset Generation

Given the two ray-tracing scenarios described in Section 3.8.1, we construct the following two datasets for the beam and blockage prediction problems.

- **Beam prediction datasets:** Here, we generate two datasets. The first adopts the LOS scenario ('O1_28' and 'O1_3p5') and use the DeepMIMO generator script [68] with the parameters described in Table 3.1. This DeepMIMO script generates the sub-

6GHz and mmWave channel sets $\mathcal{S}_{\text{sub-6}}, \mathcal{S}_{\text{mmW}}$, between the base station and every user u in the scenario. Given these channels we construct the beam prediction dataset explained in Section 3.7.2. Essentially, every data point in this dataset has the sub-6GHz channel and the corresponding one-hot vector that indicates the index of the optimal mmWave beam in the codebook $\mathcal{F}$. It is important to mention here that we adopt a simple quantized beam steering codebook. The cardinality of this codebook is set to be equal to the number of mmWave array elements ($|\mathcal{F}| = M_{\text{mmW}}$) where the nth beam, $n = 1, 2, ..., |\mathcal{F}|$ is defined as $\mathbf{f}_n = \mathbf{a}(\frac{2\pi n}{|\mathcal{F}|})$, with $\mathbf{a}(.)$ representing the mmWave array response vector. Adopting the same settings of the first dataset, a second dataset is generated using the mixed-user scenarios (I2_2p4 and I2_60). It is similar to that generated using the LOS scenario above with the difference that all of its users are NLOS (see Table 3.1).

- **Blockage prediction dataset:** This dataset considers the mixed-user scenario ('I2_2p4' and 'I2_60') and use the DeepMIMO generator script with the parameters in Table 3.2. The DeepMIMO script generates the blocked and unblocked sub-6GHz and mmWave channel sets with which the blockage dataset is constructed as described in Section 3.7.2. Each data point in that dataset consists of the sub-6GHz channel and the corresponding one-hot vector that indicates whether the LOS ray is obstructed (blocked) or not.

3.8.3 Performance Evaluation Metrics

Given that the addressed beam/blockage prediction problems in this chapter are formulated as classification problems, we adopt the *Top-1* and *Top-n* classification accuracies as the main performance metrics. The Top-1 accuracy, denoted $A_{\text{Top-1}}$, is defined as the frequency at which the deep neural network correctly predicts the class of the input. Formally,

Table 3.1: DeepMIMO Dataset Parameters for The First Dataset

Parameters	**LOS** (mmWave)	**LOS** (sub-6 GHz)
Scenario name	O1_28	O1_3p5
Active BS	3	3
Active users	700-1300	700-1300
Number of BS Antennas	64	4
Antenna spacing (wave-length)	0.5	0.5
Bandwidth (GHz)	0.5	0.02
Number of OFDM subcarriers	512	32
OFDM sampling factor	1	1
OFDM limit	32	32
Number of paths	5	15

it is written as

$$A_{\text{Top-1}} = \frac{1}{N_{\text{test}}} \sum_{n=1}^{N_{\text{test}}} \mathbb{1}_{\hat{d_n} = d_n^\star}, \tag{3.27}$$

where $\mathbb{1}_{(.)}$ is the indicator function, and $\hat{d_n}, d_n^\star$ are the predicted and target classes of the nth test point. Further, owing to the fact that a classifying deep neural network produces a probability distribution over all possible classes, it is interesting to study whether one of the top *n* predictions is the correct class instead of only focusing on the Top-1 prediction. This is customarily quantified using the Top-n accuracy. It is defined as the frequency at which the neural network correctly predicts the class of the input within its top-n predictions. In terms of beam prediction, it means that we test whether the optimal mmWave beam is within the best n predicted beam using the sub-6GHz channel. In addition to the Top-1 and Top-n accuracies, we also evaluate the performance of the proposed deep learning based model in terms of the achievable rates using the predicted mmWave beams.

Table 3.2: DeepMIMO Dataset Parameters for The Second Dataset

Parameters	**Mixed** (mmWave)	**Mixed** (sub-6 GHz)
Scenario name	I2_60	I2_2p4
Active BS	2	2
Active users	552-1159 (NLOS) 1-551 (LOS)	552-1159 (NLOS) 1-551 (LOS)
Number of BS Antennas	64	4
Antenna spacing (wave-length)	0.5	0.5
Bandwidth (GHz)	0.5	0.02
Number of OFDM subcarriers	512	32
OFDM sampling factor	1	1
OFDM limit	32	32
Number of paths	5	15

3.8.4 Neural Network Architecture and Training

In order to determine the number of stacks and the size of each one, we follow an empirical approach; we conduct a few experiments in which we vary the depths and breadths until we find the best performing network. Since beam prediction is expected to pose more challenge than blockage prediction does, we use the beam prediction dataset to identify the optimal[3] network architecture. Table 3.3 shows the results of those experiments in terms of Top-1 accuracy. It is clear that depth is of significant importance for this problem, which is in line with many recent findings in deep learning [5][6][102]. However, breadth has different impact depending on the depth; for the 6-stack network, it improves the performance while it has the opposite effect on the 2-stack network. As such, we choose to set $L_{\text{NN}} = 5$

[3] optimal for the task and dataset in hand.

stacks each of which has $M_{\text{NN}} = 2048$ neurons.

Table 3.3: Network Architecture Experiments

	512 neurons	1024 neurons	2048 neurons
2 Stacks	64.80%	64.03%	63.99%
4 Stacks	68.93%	69.51%	69.81%
6 Stacks	67.93%	69.87%	**70.57**$\%$

In all of our evaluation experiments, the neural network is trained using the datasets, explained in Section 3.8.2, for the beam and blockage prediction tasks. The training, as well as testing, samples are first contaminated with noise depending on the target SNR. Then, the network is trained in one of two ways, from scratch or transfer learning. The training approach is different in the beam and blockage predictions tasks: (i) For the beam prediction problem, the neural network training follows the training from scratch approach, where the weights are initialized randomly. The hyper-parameters are summarized in Table.3.4. (ii) For the blockage prediction problem, the neural network is trained with transfer learning. The weights of the best-performing network trained for beam prediction are used to initialize those of the base model used for blockage prediction. The only part that is trained from scratch is the end-stack. All experiments are done in MATLAB using its Deep Learning toolbox running on a machine with an RTX 2080 Ti GPU. Code files are available online at [103].

3.8.5 Beam Prediction Performance

In this subsection, we investigate the performance of the proposed sub-6GHz based mmWave beam prediction approach using the two beam-prediction datasets introduced in Section 3.8.2. First, we will start by a discussion that motivates the need for neural

Table 3.4: DNN Training Hyper-parameters

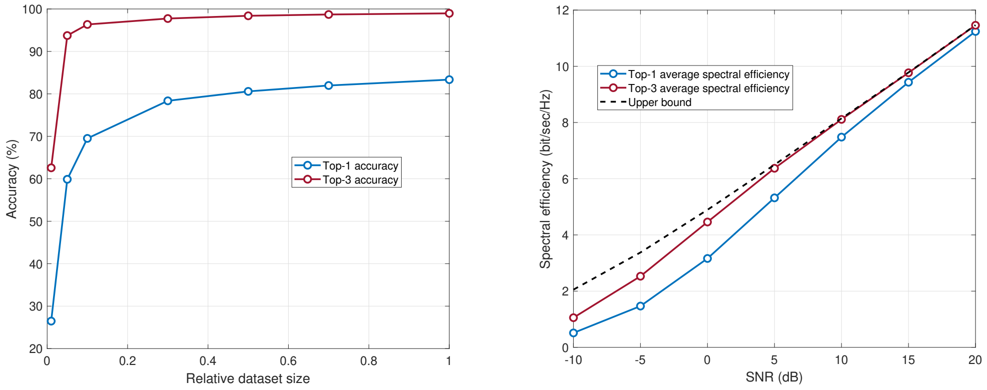
Parameter	Beam Prediction	Blockage Prediction
Solver	SGDM	SGDM
Learning rate	1×10^{-1}	1×10^{-1}
Learning rate schedule	0.1 @ $90th$ epoch	0.1 @ $40th$ epoch
Momentum	0.9	0.9
Dropout percentage	40%	40%
l_2 Regularization	1×10^{-4}	1×10^{-3}
Max. number of epochs	100	50
Dataset size $\lvert S_{band} \rvert$ (LOS)	$\approx 108 \times 10^3$	$\approx 66 \times 10^3$
Dataset size $\lvert S_{band} \rvert$ (NLOS)	None	$\approx 53 \times 10^3$
Dataset split	70%-30%	70%-30%

networks. Then, we will verify the basic claim that sub-6GHz channel can be directly used to predict the optimal mmWave beams using deep neural networks. We will also evaluate how this prediction performance is affected by noisy sub-6GHz channel measurements and the mmWave array size. Finally, we will conclude the discussion by benchmarking our proposed solution to another classical well-performing approach and showing the benefits of using deep learning.

Do we really need neural networks? This is a fundamental and important question to ask; neural networks are known for their relatively high computational requirements, and, thus, we start by exploring whether linear classifiers like Support Vector Machine (SVM) and Multinomial Logistic Regression can efficiently learn the beam prediction task in LOS settings, which is expected to be easier than mixed-user settings. We train both classifiers on the training set of the beam prediction dataset, and they both perform very poorly on the validation set. The support vector machine and logistic regression classifiers score, respectively, 3.8% and 2.9% Top-1 accuracies. These results indicate that the data samples are not linearly separable in their original space. This is usually combated using a form of feature extraction (or learning), where the input samples are transformed to another space. The goal is to obtain linear separation in the new space. Recent advances in machine learning suggest that ANN excel in such tasks [30][5][6], and, therefore, we have chosen them to tackle both prediction problems.

Neural networks learn how to predict mmWave beams from sub-6GHz channels: To validate Corollary 4 and the capability of deep neural networks in predicting the optimal mmWave beams directly from sub-6GHz channels, we plot the top-1 and top-3 beam prediction accuracies in Fig. 3.4a versus the training set size. In this figure, we adopt the LOS scenario and dataset, described in Sections 3.8.1 and 3.8.2 where the noisy channels measured at a 3.5 GHz 4-element ULA is used to predict the optimal beam for a 28 GHz 64-element array. The system operates under a high SNR regime of 20 dB. The x-axis values indicate the ratio of the training dataset samples that are actually used in training to the total number of training samples. First, Fig. 3.4a confirms the ability of neural networks in predicting the optimal mmWave beams directly from the sub-6GHz channels with high success probability that, for the adopted setup, approaches 85% and 99% for top-1 and top-3 accuracies, respectively. Further, the figure shows that 40% of the total training subset size is enough to get a beam prediction success probability κ_1 that is approximately 20% off of the upper bound. These results validate the capability of deep neural networks in effectively predicting the mmWave beams using sub-6GHz channels.

Impact of noisy channel measurements at sub-6GHz: In Fig. 3.4a, we considered a high SNR regime. Now, we want to evaluate the degradation in the mmWave beam prediction performance for different SNR regimes. Note that this SNR refers to the sub-6GHz and mmWave receive SNR, i.e., how noisy the sub-6GHz and mmWave channel



(a)

(b)

Figure 3.4: The Effect of Increasing The Training Set Size on The Beam-prediction Performance Is Shown in (a), Quantified by The Top-1 and Top-3 Prediction Accuracies. The Values on The x-axis Are Relative to The Total Training Set Size, $\approx 76,000$ Data Pairs. In (b), The Performance of The Deep Learning Solution Is Plotted When The Sub-6GHz Channels Are Contaminated With Noise. The SNR Represents The Sub-6GHz Receiver SNR.

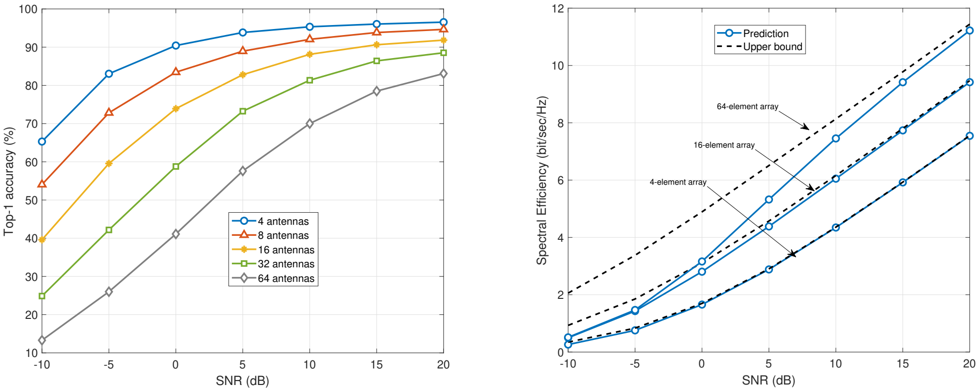
Table 3.5: Top-1 and 3 Accuracies for Sub-6GHz Based mmWave Beam Prediction

SNR (dB)	-10	-5	0	5	10	15	20
Top-1	13.3%	26%	41.1%	57.6%	70%	78.5%	83.1%
Top-3	35.9%	60%	80.4%	92.8%	96.8%	98.3%	98.8%

measurements are. While practically the SNR range for sub-6 GHz systems is higher than that of a mmWave system, we assume that the two have the same range, for simplicity. To do that, we considered the same setup of Fig. 3.4a while adding noise with different noise power values to the sub-6GHz and mmWave channels. Essentially, we study the beam prediction performance for the range of -10dB to 20dB sub-6GHz and mmWave SNR. For each SNR, the network is trained with the noisy subset of samples, and the Top-1 and Top-3 accuracies are measured on the noisy test subset. The prediction performance at this range is summarized in Table 3.5. As shown in this Table, the proposed deep learning model can clearly combat harsh noise situations. For example, the model can predict the optimal beam within its top-3 predictions with an accuracy close to 80% at 0 dB SNR, and it is the top-1 prediction around 40% of the time at the same SNR. This indicates that even in harsh conditions like that, **using the top 3 predicted beams, a very little mmWave beam training could refine the network output and improve the performance**, i.e., instead of sweeping across the whole codebook (64 beams in this case), the top-3 predictions are 80% likely to have the best one among them.

To translate this into wireless communication terms, Fig. 3.4b plots the mmWave achievable rates using the predicted beams for different values of SNR. At 0 dB SNR, the top-3 achievable rate is about 10% shy of the upper bound, which is only achieved with full knowledge of the mmWave channels. The top-1 rate, on the other hand, is not quite as close as the top-3 to the upper bound. It is about 40% off of that bound, yet it is acceptable considering the low SNR. Around 15 dB is where that gap starts closing up, dropping a little less than 5% for top-1. An important observation needs to be highlighted here. With the Top-1 accuracy at 0 dB a little above 41% in Table 3.5, it may seem a bit surprising that the rate only drops 40% from the upper bound. This implies that even when the DNN mis-classifies, it seems to select a beam that is not very far away from the correct one. Such claim is corroborated with the relatively high Top-3 accuracy. This could also be observed at 5 and 10 dB SNRs.

Performance with different mmWave array sizes: With that interesting performance above, one question could come to mind: Is such performance attainable with any number of mmWave antennas? A smaller number of mmWave antennas means there are less classes to learn. On the surface, this looks like an easier prediction task for the deep neural network, which is true. Fig. 3.5a shows the top-1 performance of the neural network with different



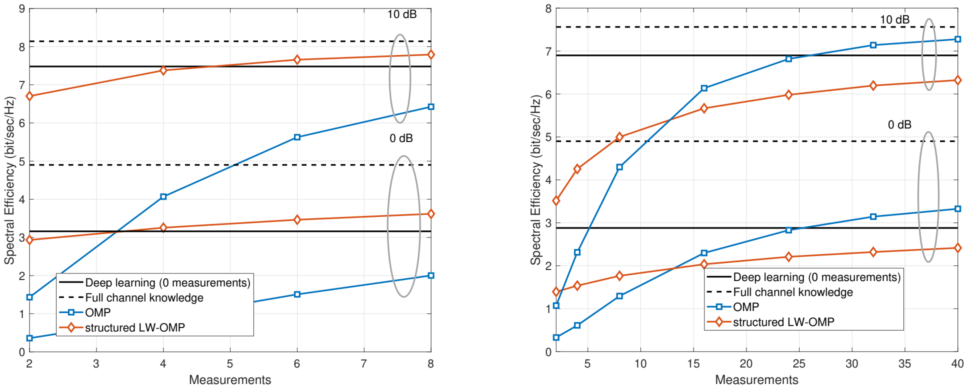
(a)

(b)

Figure 3.5: (a) Shows The Prediction Accuracy of The DNN As The Number of mmWave Antenna Elements Increases. Larger mmWave Antenna Amounts to Larger Beam-forming Codebook and, Therefore, Larger Number of Classes. For Some Choices of mmWave Antennas, The DNN Model Top-1 Achievable Rate Is Plotted with Its Upper Bound in (b). All Curves Are Obtained with AWGN Only.

numbers of mmWave antennas. It is very clear that the proposed deep learning model has better classification performance with a small number of antennas, no matter what the SNR level is. This trend translates to the top-1 achievable rate performance. Fig. 3.5b shows the average achievable rate against SNR for three different mmWave antenna arrays. Although the antenna gain is low with small number of mmWave elements, the deep neural network achieves a much smaller gap with the upper bound for small number of elements compared to that achieved for a large number of elements. This is an immediate reflection of the complexity of the classification task.

Performance compared to prior work: In order to further highlight the importance and novelty of our proposed solution, we perform a comparative study with some popular classical mmWave beam selection approaches, namely orthogonal matching pursuit



(a)

(b)

Figure 3.6: Two Comparative Performance Figures. In (a), The Performance of The Neural Network Is Benchmarked to That of Orthogonal Matching Pursuit and Logit-weighted Orthogonal Matching Pursuit Under LOS Setting and Two Different SNR Levels. Figure (b) Depicts The Same Performance Benchmarking But Under NLOS Setting.

(does not use sub-6GHz channels) and logit-weighted orthogonal matching pursuit (uses sub-6GHz channels) in [71]. We use the first and second beam-prediction datasets introduced in Section 3.8.2. Fig. 3.6a depicts the spectral efficiency versus number of mmWave meansurements in LOS setting and at two different SNR levels while Fig. 3.6b depicts the same thing but in NLOS setting. From the two figures, we can see that in both settings and at any SNR level, the proposed deep learning model exhibit clear advantage over the two classical approaches. For instance, in Fig. 3.6a and at 0 dB SNR, the logit-weighted orthogonal matching pursuit needs about 4 measurements to match the performance of our model, which requires 0 measurements. The value of our approach becomes even clearer when we turn to NLOS setting, as in Fig. 3.6b; both classical approaches require more that 15 measurements to get close to the spectral efficiency of the neural network. This gain is expected to further increase when the mobile user also employs an antenna array.

3.8.6 Blockage Prediction

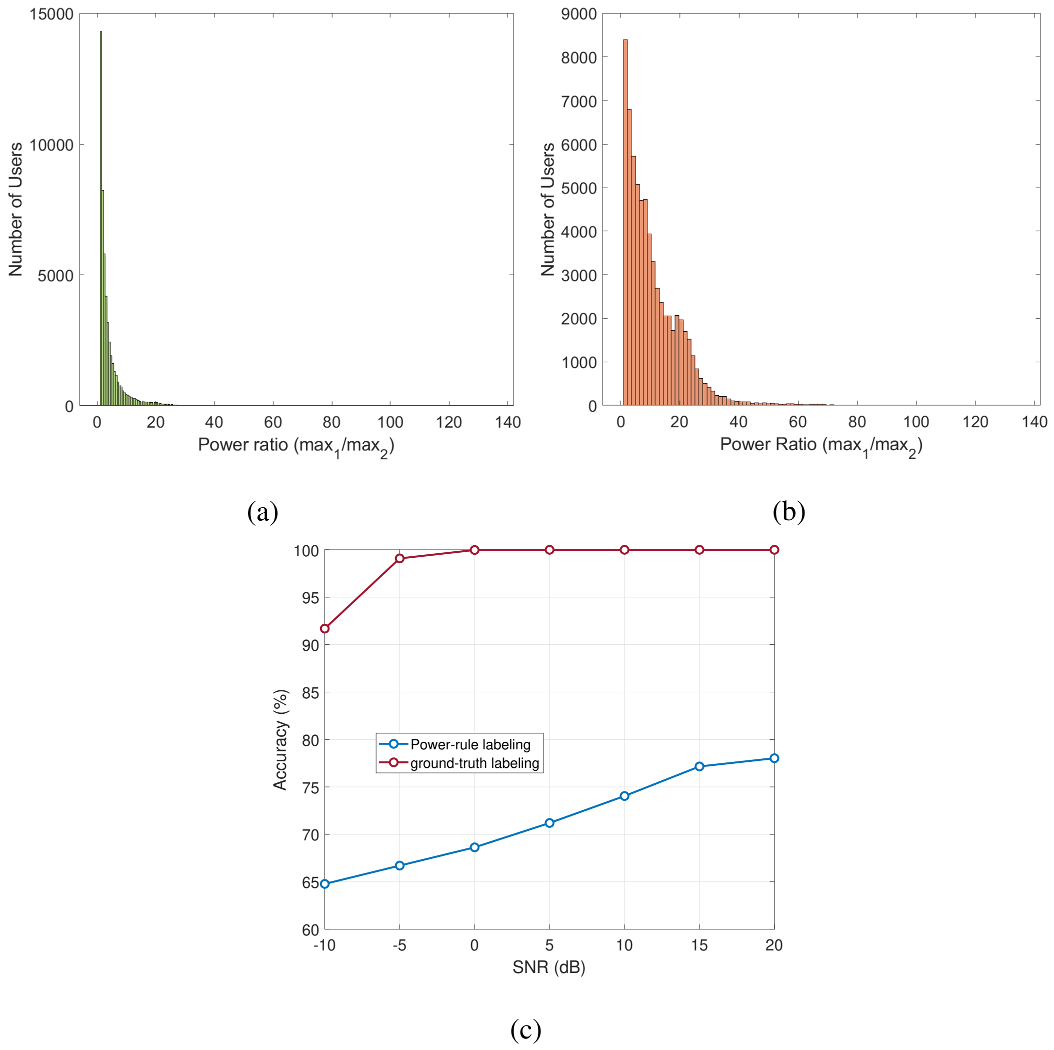
The second set of experiments aims at evaluating the ability of the deep neural networks to *differentiate* blocked and LOS users from the same spatial region. For this end, we adopt the mixed-user dataset described in Section 3.8.2, that mixes the LOS and blocked users. Given this dataset, we investigate the blockage prediction performance for the following two labeling approaches

- **Ground-truth labeling:** This approach assumes the availability of accurate user labels by some means such as, for example, simultaneous localization and mapping techniques. While this may not be a very practical approach, it provides an upper bound for the performance of the other labeling techniques.

- **Power-based labeling:** The LOS paths are normally much stronger (have higher power) compared to the NLOS ones. Therefore, one possible way to differentiate the blocked and unblocked users is by computing the ratio between the power of the strongest beam in the codebook to that of the second strongest beam for each user, referred to as the *power-rule* labelling. This ratio is expected to be large for unblocked users and small (close to one) for blocked users. Fig. 3.7 and Fig. 3.7b corroborate such intuition; they show two power-ratio histograms, one for the blocked users and the other for LOS users. It is clear that majority of blocked users have power-ratios close to one. With that, a threshold for labeling could be set and used to create the labels during the background training.

For the mixed-user setup, and as discussed earlier in Section 3.8.4, transfer learning is used to train the deep neural network. In Fig. 5.5.2, we plot the success probability (accuracy percentage) of blockage prediction. First, Fig. 5.5.2 illustrates that the deep learning model has excellent classification ability for the ground-truth labeling approach under a wide range of SNRs. This performance is then compared to the case when the power-rule labeling technique is used. Despite the label contamination, i.e., some miss-labeled users are present during training, the DNN model still performs relatively well; its accuracy exceeds 70% at high SNRs. This highlights the potential of using sub-6GHz channels to effectively predict mmWave link blockages.

These results are not surprising considering what have been reported in the literature, such as that in [62] and [94] for example. They propose different approaches, one relies on recurrent neural networks while the other resort to classical machine and statistical leaning. Our experiments, in comparison to those in the literature, confirm the ability of neural networks to perform exceedingly well in blockage prediction. They also highlight two interesting points. The first is that multi-layer perceptron networks are effective enough for current blockage prediction given a properly labeled dataset. This is important as those networks do not require sequences of channel observations as recurrent networks do. In addition, multi-layer perceptron networks have the interesting flexibility to adapt to changes in the environment, which is the second point. Our transfer learning experiments shows that a network trained on a different task can easily be fine-tuned for another task and environment. This is a great advantage for neural networks in general compared to classical machine and statistical learning approaches, in which the model needs to be re-engineered to adapte to new environments.



(a)

(b)

(c)

Figure 3.7: Power-ratio Histograms and The Prediction Accuracy. (a) and (b) Show The Histograms of Blocked and Unblocked Users, Respectively. (c), on The Other Hand, Depicts The Prediction Accuracy of Both Labeling Techniques.

Chapter 4

ENABLING DYNAMIC MASSIVE MIMO WITH DEEP LEARNING: FROM DETERMINISTIC TO STATISTICAL CHANNEL PREDICTION

4.1 Scope and Contributions

Scope

In the previous two chapters, namely Chapters 2 and 3, the channel/beam-training overhead in large-scale MIMO is tackled with the novel deterministic channel-prediction framework. The theoretical grounds of the framework and the role of ML have been established. They both have been empirically verified over four case studies spanning various large-scale MIMO implementations and using two different deep learning approaches. Like any engineering frameworks, however, deterministic channel prediction has its own shortcomings to which the first word in the name "deterministic" hints. The framework struggles with the dynamic and random nature of the wireless environment, as suggested by the results in Section 2.8. To further delve into the details of its shortcomings and address them, this paper presents a new framework that could be seen as a natural evolution of the deterministic channel prediction framework, namely the *statistical channel-prediction framework*.

The new framework represents an evolution from the perspectives of both large-scale MIMO and deep learning. It re-envisions the role of deep learning, and ML in general, in addressing the channel/beam-training overhead. The new framework does not target developing deep learning algorithms (e.g., DNNs) that completely eliminate the need for channel/beam training. It, instead, attempts to develop deep learning algorithms that significantly reduce the training overhead. This is done by learning a prediction function for some user-specific summary statistics in the form of a conditional channel covariance. This covariance helps the MIMO system perform good-quality yet light-weight channel training to estimate the target channels. From the ML perspective, on the other hand, this task of learning user-specific statistics brings about some interesting advantages: (i) it alleviates the impact of prediction error on the system performance; and (ii) it brings to light the value of designing robust deep learning algorithms. It does so by placing emphasis on unsupervised learning.

Contributions

Tapping into the recent advances in deep learning and specifically DNNs [24, 30, 73], this paper addresses the challenges of channel training and feedback in massive MIMO by proposing the novel *statistical channel-prediction framework*. It is based on the idea of learning the prediction of a conditional channel covariance that is conditioned on some observed channels, e.g., predicting a conditional downlink covariance given the uplink channels. In particular, the main contributions envelop a framework with two machine learning approaches, two proposed solutions, and a set of evaluation experiments. The following details those contributions:

- The statistical channel prediction framework is formally defined with an emphasis on the challenges it addresses. Two approaches are proposed to perform the statistical prediction task. Although addressing the same problem, the two approaches are fundamentally different from a machine learning perspective. The first relies on regression, which is a supervised learning approach, while the other relies on clustering, which is an unsupervised learning approach.
- Two DNN architectures are designed as possible solutions to the covariance prediction problem. Each one is an implementation of a proposed approach. They both take advantage of some of the most recent advances in the field of deep learning,

specifically residual learning [6] and deep denoising autoencoders [104].

- The ability of each architecture to reduce the channel/beam-training overhead is evaluated and studied using two dataset of uplink-downlink channel pairs, one per basestation. The datasets are generated from a dynamic communication scenario with two basestations, provided by the publicly available DeepMIMO dataset [68]. Both solutions report interesting performances, whether under single or multi-user settings.

4.2 Related Work

Overcoming the channel-training and channel feedback challenges in massive MIMO is a popular and active research direction [17, 20, 56, 71, 72, 80, 105, 106]. The work on those challenges could be loosely divided into two categories, signal-processing-based and machine-leaning-based. The former could be thought of as the classical view on how to handle the channel-related problems in communications. For instance, [105] uses signal processing to estimate the parameters of the uplink channels and use them to construct the downlink channel at another frequency. The work in [56] extends that in [105] and derive some lower bounds on the channel reconstruction (or extrapolation as the authors refer to it). On the same direction, [71, 72] attempts to utilize spatial channel correlation between sub-6 GHz uplink channels and mmWave downlink channel to reduce the design of beamforming vectors. All the previous work overlaps in the need to estimate some hand-picked latent parameters. This makes it subject to the limitations of the wireless systems in resolving those parameters.

More adaptive and data-driven approaches have surfaced recently as a way to address those limitations. They make up the second category relying on machine learning. [20], for instance, develops a channel compression scheme based on deep learning. It attempts to reduce the feedback overhead, but it does not address the training overhead. A more generic approach to dealing with both channel training and feedback has been proposed in [106], in which a deep learning framework is proposed to learn channel mapping across space and frequency. Such framework has given raise to spin-offs like [17, 80]. [80] utilizes the mapping concept to predict mmWave beamforming vectors given sub-6 GHz uplink channels. It shows clear improvement over classical approaches tackling the same problem. [17], on the other hand, extends the mapping concept to channel-estimation in one-bit ADC settings. It results in some interesting findings such as more antennas yield better performance. Aside from all the promising results, those machine learning approach are still in their infant years. They are lacking in terms of evaluation and optimality. Questions related to their suitability to real-world wireless environments and their ability to generalize to different settings remain open for further studying.

4.3 System and Channel Models

The system and channel models used throughout this paper are presented in the following two sections.

4.3.1 System Model

The system model in this work assumes two massive MIMO basestations deployed in a dynamic wireless environment, each of which has a set of antennas $\mathcal{M}_m$ where $m \in \{1, 2\}$ is a basestation index. Fig. 4.1 depicts an illustration of the system. Each uth user at location $\mathbf{x}_u \in \mathbb{R}^3$ is equipped with a single-element antenna and is able to communicate with both basestations. The system adopts K-subcarrier Orthogonal Frequency-Division Multiplexing (OFDM). The received uplink signal of the uth user at the mth basestation could be expressed as

$$\mathbf{y}_{k,u,m}^{\text{UL}} = \mathbf{h}_{k,u,m}^{\text{UL}} s_u + \mathbf{n}_{k,u}, \tag{4.1}$$

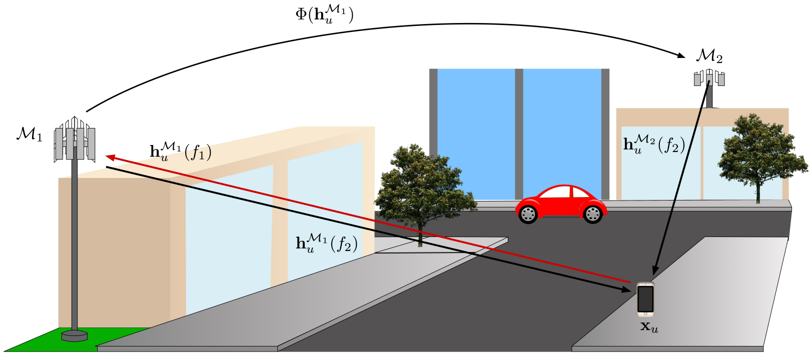

$\Phi(\mathbf{h}_u^{\mathcal{M}_1})$
$\mathcal{M}_2$
$\mathcal{M}_1$
$\mathbf{h}_u^{\mathcal{M}_1}(f_1)$
$\mathbf{h}_u^{\mathcal{M}_2}(f_2)$
$\mathbf{h}_u^{\mathcal{M}_1}(f_2)$
$\mathbf{x}_u$

Figure 4.1: An Illustration of a User Equipment Connected to Two Different Massive MIMO Basestations, $\mathcal{M}_1$ and $\mathcal{M}_2$. The User x_u Experiences Different Uplink and Downlink Channels with Both Basestations.

where $s_u \in \mathbb{C}$ is a complex symbol transmitted by the uth user; $\mathbf{h}_{k,u,m}^{\text{UL}} \in \mathbb{C}^{|\mathcal{M}_m|}$ is the uplink channels between the uth user and the mth basestation at the kth subcarrier; $|\mathcal{M}_m|$ is the number of antenna elements deployed at the mth basestation; and $\mathbf{n}_{k,u} \in \mathbb{C}^{|\mathcal{M}_m|}$ is a complex Gaussian noise vector sampled from $\mathcal{CN}(0, \sigma^2\mathbf{I})$. The downlink signal transmitted by the mth basestation and received by the uth user in the environment could be expressed as:

$$y_{k,u,m}^{\text{DL}} = (\mathbf{h}_{k,u,m}^{\text{DL}})^H \mathbf{f} s_m + n_{k,u}, \tag{4.2}$$

where $s_m \in \mathbb{C}$ is a complex symbol transmitted by the mth basestation; $\mathbf{f} \in \mathbb{C}^{|\mathcal{M}_m|}$ is the beamforming vector applied by the mth basestation; $\mathbf{h}_{k,u,m}^{\text{DL}} \in \mathbb{C}^{M_m}$ are the downlink channels between the mth basestation and the uth user at the kth subcarrier; and $n_{k,u} \in \mathbb{C}$ is a complex Gaussian noise sample drawn from $\mathcal{CN}(0, \sigma^2)$.

4.3.2 Channel Model

This paper adopts a geometric (physical) channel model [40]. With this model, the uplink and downlink channels can be written as

$$\mathbf{h}_{k,u,m}^{\text{band}} = \sum_{d=0}^{D_{\text{c}}-1} \sum_{\ell=1}^{L} \alpha_\ell e^{-j\frac{2\pi k}{K}d} p\left(dT_{\text{S}} - \tau_\ell\right) \mathbf{a}\left(\theta_\ell, \phi_\ell\right), \tag{4.3}$$

where "band" is either "UL" or "DL", L is number of channel paths, $\alpha_\ell, \tau_\ell, \theta_\ell, \phi_\ell$ are the path gains (including the path-loss), the delay, the azimuth angle of arrival (AoA), and elevation AoA, respectively, of the ℓth channel path. T_{S} represents the sampling time while D_{c} denotes the cyclic prefix length (assuming that the maximum delay is less than $D_{\text{c}}T_{\text{S}}$). The advantage of the physical channel model is its ability to capture the physical characteristics of the signal propagation including the dependence on the environment geometry, materials, frequency band, etc., which is crucial for the machine-learning-based framework proposed in this paper. Not that moving forward, the subcarrier subscript, i.e., $k \in \{1, \dots, K\}$, will be eliminated for simplicity.

4.4 Challenges to Deterministic Channel Prediction

The deterministic channel prediction framework is conceptualized as a solution to the channel-training problem in large-scale MIMO communications. However, like any framework, it has its shortcomings that motivates further development. The main issues that cause it to struggle in achieving its goal could be loosely grouped into three categories: (i) environmental challenges, (ii) hardware-based challenges, and (iii) machine learning challenges.

From a wireless communication perspective, the deterministic mapping framework does not capture the random nature of the wireless channel. That nature is shaped by the dynamics in the environment and the randomness caused by the communication hardware. Large- and small-scale fading both add up to a time-varying channel at the receiver

[107], which means that the channels-to-channels mapping function defined in [106] is time-varying. Such characteristic is not captured by the deterministic channel-prediction framework. In addition to the fading effect, hardware-based randomness such as noisy channel measurements and limited bandwidth ADCs contributes to the overall behavior of the wireless channel. These effects can be, to some extent, handled by the deterministic prediction framework, but the framework does not offer clear signs of robustness; the neural network could learn to combat those effects when they are exhibited in the training set [80, 106]. However, since the framework is targeting the channel-training challenge in large-scale MIMO, any small channel prediction error could have pervasive effects on the system performance. The prediction is performed for relatively high-dimensional vectors, and they are more prone to perturbation error than their low dimensional counterparts. This casts some doubts on how robust the framework is in practice.

The reliance of the deterministic channel=prediction framework on neural networks, despite being an asset to the framework, contributes to its challenges. Neural networks in general and DNNs in particular represent the state-of-the-art algorithms in machine learning [73], and as such, they are the driving power of the deterministic channel-prediction framework. They, however, have a relatively hefty training data requirement, especially when the network has a large number of parameters. Earlier work, like that in [80, 106], has shown that a number in the neighborhood of 15 thousand data samples is needed to learn the mapping function. That number of samples may not seem large at first, yet when the need to collect that amount of samples from a *stationary environment* is factored in, the challenge becomes clear; it should not be feasible to expect a communication system to collect thousands of data samples from a stationary environment. The collection process is more likely to happen over a relatively lengthy periods of time, within which the environment is not stationary at all. Such challenge is not clearly addressed in the framework and is expected to be a major aspect of any further development.

4.5 Statistical Channel Prediction: Problem Statement and Framework Outline

Aside from the elegance of the deterministic mapping framework, it is expected to fall short in practice due to the aforementioned challenges. Therefore, there is a need for a more practical framework that can jointly address those challenges and the question on mapping channels between different antenna sets and frequencies. In particular, the framework needs to account for the dynamics of the environment, show clear signs of robustness, and have practical machine learning requireements. One promising way to meet those requirements is the learning of a function predicting the *conditional channel covariance*. In an abstract sense, this means *given* the channels of a user observed at a set of antennas and a certain frequency (henceforth referred to as the observed channels), the function objective is to predict the conditional covariance of the user's channels at another set of antennas and/or another frequency (henceforth referred to as the target covariance and target channels, respectively). This predicted covariance is, then, used to estimate the channels using some minimal-overhead channel training.

The prediction of the target covariance is a claver approach to combat the challenges associated with deterministic channel prediction. The variability of the mapping function in [106] renders it too complex to be learned. A closer look at the cause of that complexity, as discussed in Section 4.4, reveals that it is the collective result of multiple sources of randomness, e.g., channel fading, noisy measurements, and phase noise among others. Such sources have a probabilistic nature and, hence, their effect on the observed and target channels can be characterized by some form of summary statistics like the conditional covariance; given some observed channels, the covariance of the target channels captures the underlaying vector subspace wherein the target channels vary. That is why instead of attempting to learn the mapping function itself, learning an alternative function predicting the conditional covariance is expected to be more robust and practical for real wireless

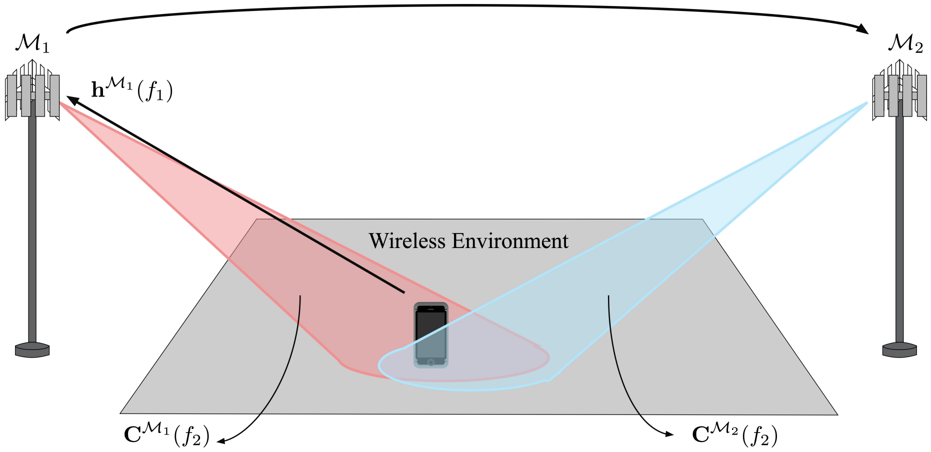

$\mathcal{M}_1$
$\mathcal{M}_2$
$\mathbf{h}^{\mathcal{M}_1}(f_1)$
Wireless Environment
$\mathbf{C}^{\mathcal{M}_1}(f_2)$
$\mathbf{C}^{\mathcal{M}_2}(f_2)$

Figure 4.2: An Illustration of Statistical Channel Prediction. It Shows a User Communicating with Two BSs. The Uplink Channels at One BS (Antenna set $\mathcal{M}_1$) Is Used to Predict The Conditional Covariance at The Same BS and Different Frequency $\mathbf{C}^{\mathcal{M}_1}(f_2)$ and/or The Conditional Covariance at Another BS and Frequency $\mathbf{C}^{\mathcal{M}_2}(f_2)$

communications.

4.5.1 Problem Definition

Consider the communication setup in Fig. 4.2, the uth user at location $\mathbf{x}_u$ is communicating with two basestations, $\mathcal{M}_1$ and $\mathcal{M}_2$, at the same uplink and downlink frequencies, respectively f_1 and f_2. The objective is to predict the covariance of the downlink channels at one basestation given the uplink channels at the same basestation or the other. Without loss of generality, say that the uplink channels between the uth user and the first basestation at the kth subcarrier are the observed channels and the conditional downlink-channel covariance at the second basestation is the target covariance. Such conditional covariance is formally defined as

$$\mathbf{C}_{u,k,2}^{\text{DL}} = \mathbb{E}\left[(\mathbf{h}_{u,k,2}^{\text{DL}} - \boldsymbol{\mu}_{\mathbf{h}_{\text{DL}}|\mathbf{h}_{\text{UL}}})(\mathbf{h}_{u,k,2}^{\text{DL}} - \boldsymbol{\mu}_{\mathbf{h}_{\text{DL}}|\mathbf{h}_{\text{UL}}})^H|\mathbf{h}_{u,k,1}^{\text{UL}}\right], \tag{4.4}$$

where $\boldsymbol{\mu}_{\mathbf{h}_{\mathrm{DL}}|\mathbf{h}_{\mathrm{UL}}} = \mathbb{E}\left[\mathbf{h}_{u,k,2}^{\mathrm{DL}}|\mathbf{h}_{u,k,1}^{\mathrm{UL}}\right]$ is the mean of the downlink channels given certain uplink channels. Note that: (i) the observed channels and target covariance both represent one user at a position $\mathbf{x}$ in the wireless environment; and (ii) they could be for a user communicating with one basestation or for a user communicating with two different basestations. Therefore, for simplicity of exposition, the user, subcarrier, and basestation indices will be dropped henceforth, i.e., $\mathbf{C}_{u,k,2}^{\mathrm{DL}}$, $\mathbf{h}_{u,k,1}^{\mathrm{UL}}$, and $\mathbf{h}_{u,k,2}^{\mathrm{DL}}$ will be denoted by $\mathbf{C}$, $\mathbf{h}_{\mathrm{UL}}$, and $\mathbf{h}_{\mathrm{DL}}$.

The task of predicting $\mathbf{C}$ could be abstractly viewed as a function-learning problem from a given dataset. Putting this in mathematical terms, let $\mathbb{P}(\mathbf{h}_{UL}, \mathbf{h}_{\mathrm{DL}})$ be a data-generating distribution that produces pairs of unplink and downlink channels, and let a prediction function $f_{\Theta}(\mathbf{h}_{\mathrm{UL}})$ be defined as $f : \Theta, \mathbf{h}_{\mathrm{UL}} \in \mathbb{C}^{W} \times \mathbb{C}^{M_1} \rightarrow \hat{\mathbf{C}} \in \mathbb{C}^{M_2 \times M_2}$ where Θ is a W-dimensional vector parameterizing that function. Given a dataset of pairs $\mathcal{S} = \{(\mathbf{h}_{\mathrm{UL}}, \mathbf{h}_{\mathrm{DL}})_u\}_{u=1}^{U}$ sampled from the distribution (henceforth referred to as the mother dataset), that function needs to be learned such that it maximizes the joint probability of correct prediction:

$$\max_{f_{\Theta}(\mathbf{h}_{\mathrm{UL}})} \mathbb{P}\left(\hat{\mathbf{C}}_1 = \mathbf{C}_1, \ldots, \hat{\mathbf{C}}_U = \mathbf{C}_U | \mathbf{h}_{UL_1}, \ldots, \mathbf{h}_{UL_U}\right) \tag{4.5}$$

where $\hat{\mathbf{C}}_u, \mathbf{C}_u, \ \forall u \in \{1, \ldots, U\}$ are respectively the downlink covariance predicted by $f_{\Theta}(\mathbf{h}_{\mathrm{UL}})$ using the uth uplink channel and the target covariance of the uth downlink channels in the dataset $\mathcal{S}$. A couple of important points about (4.5) need to be raised here:

- Owing to the fact that a dataset is collected from a massive MIMO wireless environment, estimating a conditional covariance could prove very challenging; it requires the collection of many downlink channels from a fixed user location over an extended period of time. The more acceptable form of data to be collected is pairs of uplink and downlink channels as in $\mathcal{S}$.

- The joint probability of correct prediction implies that the prediction function $f_{\Theta}(\mathbf{h}_{\mathrm{UL}})$ should ultimately exhibit the same prediction performance over all data samples in

$\mathcal{S}$.

The question now is how to learn the prediction function and satisfy (4.5). Two machine learning approaches are utilized in this work to address that question. The following subsection draws the general outline of the proposed solutions while the details are left to Sections 4.6 and 4.7.

4.5.2 Framework Outline

In an attempt to find the prediction function that satisfies (4.5), two approaches are developed. One relies on viewing the problem from a regression perspective while the other views it from a clustering perspective. The two views are very different. With regression, the relation between the target covariance and observed channels, which is probabilistic in nature, is assumed to be modeled by some function $f_\Theta(\mathbf{h}_{\text{UL}})$ corrupted by some noise source. This function is assumed to belong to some family of functions $\mathcal{H}$– examples are linear, polynomial, and exponential families [31, 73]. To learn the relation and, hence, the function $f_\Theta(\mathbf{h}_{\text{UL}})$, a training dataset $\mathcal{S}_{t_1}$ is generated using the samples in $\mathcal{S}$ such that for every $\mathbf{h}_{\text{UL}}$, there is a sample covariance $\mathbf{h}_{\text{DL}}\mathbf{h}_{\text{DL}}^H$. A neural network with a parameter set Θ is, then, fit to the data in $\mathcal{S}_{t_1}$ such that the predictions of $f_\Theta(\mathbf{h}_{\text{UL}})$ has minimum Mean Squared Error (MSE) with the target covariances in $\mathcal{S}_t$. Under certain conditions, as will be shown in Section 4.6, the result of such fitting process (training) is a prediction function that asymptotically produces the target covariances.

The other view on the problem follows a clustering approach. It relies on an interpretation of the geometric channel model in Section 4.3 that is inspired by the work in [47]. This interpretation sees the wireless environment partitioned into channel rings from the perspective of each BS, see Fig.4.3. The users within each ring share the same channel eigenspace (whether uplink or downlink), and, hence, their covariances given the uplink channels are approximately the same. The proposed approach utilizes the channel-ring no-

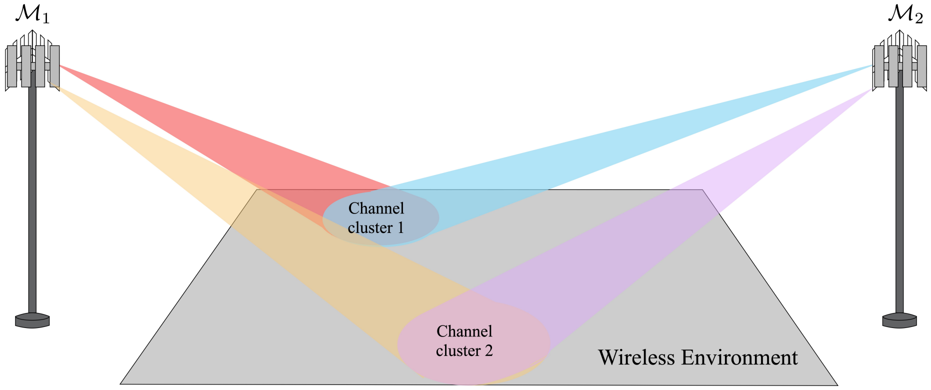


Figure 4.3: An Illustration of The Channel Cone Interpretation. The Wireless Environment Could Partitioned into Multiple Clusters of Major Reflectors. From The Perspective of Each BS, These Clusters Look Like Cones of Possible Channel Directions.

tion and translates the covariance prediction into a clustering problem. In this new light, given a training set $\mathcal{S}_{t_2}$, the function $f_\Theta(\mathbf{h}_{\text{UL}})$ is expected to produce a clustering of the uplink channels, and by knowing the clusters, the covariance of each clusters is estimated using the downlink channels of that cluster in $\mathcal{S}_t$.

4.6 Statistical Channel Prediction: A Regression Approach

With both the problem formulation (Section 4.5.1) and the general outline of the regression approach (Section 4.5.2) in mind, the following subsections give a formal treatment to how the conditional covariance is learned, a description of the proposed neural network architecture, and a discussion on the possible challenges to the regression approach.

4.6.1 Learning Target Covariance with Regression

It is common in supervised machine learning to look at the relation between the target and observed variables as probabilistic and governed by an unknown data-generating distribution [73]. In the realm of regression, this relation between the target variable and the observed variable is modeled by a parametrized function $f_\Theta(.)$ that belongs to some family of functions $\mathcal{H}$. The choice of the family is made based on some a priori domain knowledge. To capture the possible error in that choice and in the learning process, the relation between the target variable and the prediction $f_\Theta(.)$ makes is assumed to be corrupted by some noise source [31, 73]. The gaol of the machine learning algorithm, then, is to tune the parameters of the function until it finds the *best* representation that minimizes some error metric between the prediction and the target variable. This is done using a training set sampled from the data distribution.

To translate the above into a formal treatment for the covariance prediction problem defined in Section 4.5.1, the variables and prediction function need to be re-defined in the corresponding real-valued spaces. This choice is motivated by modern machine learning applications and frameworks, e.g., image classification [5–7], machine translation [1, 13] for applications and PyTorch [81] and TensorFlow [82] for frameworks. Let $\mathcal{S}_{t_1} = \{(\tilde{\mathbf{h}}_{\text{UL}}, \mathbf{C}_s)_u\}_{u=1}^{U}$ be a set of training samples where $\mathbf{C}_s = \tilde{\mathbf{h}}_{\text{DL}}\tilde{\mathbf{h}}_{\text{DL}}^T$ is referred to as the *sample covariance* and:

$$\tilde{\mathbf{h}}_{\text{UL}} = [h^r_{UL_1}, \dots, h^r_{UL_{M_1}}, h^{im}_{UL_1}, \dots, h^{im}_{UL_{M_1}}]^T \in \mathbb{R}^{2M_1}, \tag{4.6}$$

$$\tilde{\mathbf{h}}_{\text{DL}} = [h^r_{DL_1}, \dots, h^r_{DL_{M_2}}, h^{im}_{DL_1}, \dots, h^{im}_{DL_{M_2}}]^T \in \mathbb{R}^{2M_2}, \tag{4.7}$$

are the result of stacking the real and imaginary parts of $\mathbf{h}_{\text{UL}}$ and $\mathbf{h}_{\text{DL}}$. $\mathcal{S}_{t_1}$ is obtained from the mother dataset $\mathcal{S} = \{(\mathbf{h}_{\text{UL}}, \mathbf{h}_{\text{DL}})_u\}_{u=1}^{U}$. The prediction function is re-defined as:

$$f : \Theta, \tilde{\mathbf{h}}_{\text{UL}} \in \mathbb{R}^W \times \mathbb{R}^{2M_1} \rightarrow \mathbf{C}_s \in \mathbb{R}^{2M_2 \times 2M_2}, \tag{4.8}$$

where the parameter vector Θ is now restricted to the real-valued vector space $\mathbb{R}^W$. Using the above definitions, the relation between the observed channels and the sample covariances is modeled by the function $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ and a unimodal noise source[1] $\mathbf{N} \in \mathbb{R}^{2M_2 \times 2M_2}$ as

[1] Although a unimodal distribution might not seem like an appropriate assumption for some applications, it provides a fundamental building block to handle cases with multi-modal noise distributions, see [31].

follows:

$$\mathbf{C}_s = f_\Theta(\tilde{\mathbf{h}}_{\text{UL}}) + \mathbf{N}. \tag{4.9}$$

Under certain conditions, a machine learning algorithm learning the function $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ on $\mathcal{S}_{t_1}$ with an MSE training loss (metric) could asymptotically learn the sought-after conditional covariance and maximize the objective of (4.5). The following proposition presents the formal statement of the regression approach.

Proposition 7 *Under the following conditions:*

1. *The samples of $\mathcal{S}_{t_1}$ are independent and identically distributed (iid),*
2. *The noise term in* (4.9) *follows a matrix normal distribution $\mathcal{MN}(\mathbf{M}, \mathbf{I}, \mathbf{I})$ where $\mathbf{M}$ is an all zero $2M_2 \times 2M_2$ matrix and $\mathbf{I}$ is the $2M_2 \times 2M_2$ identity matrix [108],*
3. *The number of samples $U \to \infty$,*
4. *The downlink channels $\mathbf{h}_{DL}$ and their stacked version $\tilde{\mathbf{h}}_{DL}$ have zero conditional mean, i.e., $\boldsymbol{\mu}_{\mathbf{h}_{DL}|\mathbf{h}_{UL}} = \boldsymbol{\mu}_{\tilde{\mathbf{h}}_{DL}|\tilde{\mathbf{h}}_{UL}} = 0$,*

and using the regression model in (4.9), *the function $f_\Theta(\tilde{\mathbf{h}}_{UL})$ that is trained to minimize the MSE loss*

$$\mathcal{L} = \frac{1}{U}\sum_{u=1}^{U} ||\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})||_F^2,$$

over $\mathcal{S}_{t_1}$ is a maximizer to (4.5).

Proof: Given the regression model in (4.9), the function $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ that best models the relation between $\mathbf{C}_s$ and $\tilde{\mathbf{h}}_{\text{UL}}$ could be found by maximizing the following joint probability over $\mathcal{S}_{t_1}$ [31]

$$\max_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \mathbb{P}(\mathbf{C}_{s_1} = \hat{\mathbf{C}}_{s_1}, \dots, \mathbf{C}_{s_U} = \hat{\mathbf{C}}_{s_U} | \tilde{\mathbf{h}}_{UL_1}, \dots, \tilde{\mathbf{h}}_{UL_U}), \tag{4.10}$$

where $\hat{\mathbf{C}}_{s_u} = f_\Theta(\tilde{\mathbf{h}}_{UL_u})$ for $u \in \{1, \dots, U\}$. From condition 1, (4.10) can be expressed as

$$\max_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \prod_{u=1}^{U} \mathbb{P}\left(\hat{\mathbf{C}}_{s_u} = \mathbf{C}_{s_u} | \tilde{\mathbf{h}}_{UL_u}\right). \tag{4.11}$$

From (4.9) and condition 2, the objective of (4.11) follows a Gaussian distribution

$$\prod_{u=1}^{U} \mathbb{P}\left(\hat{\mathbf{C}}_{s_u} = \mathbf{C}_{s_u} | \tilde{\mathbf{h}}_{UL_u}\right) = \prod_{u=1}^{U} \frac{1}{\alpha} \exp\left[-\frac{1}{2} \operatorname{Tr}\left((\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u}))^T(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})\right)\right], \tag{4.12}$$

where $\alpha = \sqrt{(2\pi)^{2M_2}}$. As such, the optimization in (4.11) could be expressed as

$$\min_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \prod_{u=1}^{U} \frac{1}{\alpha} \exp\left[-\frac{1}{2} \operatorname{Tr}\left((\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u}))^T(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})\right)\right]. \tag{4.13}$$

Maximizing (4.13) is equivalent to minimizing the negative log-likelihood of the objective as follows

$$\min_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \sum_{u=1}^{U} \frac{1}{2} \operatorname{Tr}\left[(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u}))^T(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})\right]. \tag{4.14}$$

Noting that $(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u}))$ is a $2M_2 \times 2M_2$ symmetric matrix, the trace operator could be replaced with a Frobenius norm squared $|| \dots ||_F^2$ as follows

$$\operatorname{Tr}\left[(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u}))^T(\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})\right] = ||\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})||_F^2 \tag{4.15}$$

Substituting (4.15) into (4.14) yields

$$\min_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \sum_{u=1}^{U} \frac{1}{2} ||\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})||_F^2, \tag{4.16}$$

Noting that scaling the objective (4.16) by $1/U$ does not change the minimizer, the optimization in (4.16) becomes

$$\min_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \frac{1}{2U} \sum_{u=1}^{U} ||\mathbf{C}_{s_u} - f_\Theta(\tilde{\mathbf{h}}_{UL_u})||_F^2, \tag{4.17}$$

This yields the the MSE loss metric, which when minimized by $f_\Theta(\tilde{\mathbf{h}}_{\text{UL})}$ results in the sought-after prediction function minimizing (4.4). To show that, using condition 3 and the law of large numbers [109], the objective in (4.17) could be expressed as

$$\min_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \frac{1}{2}\left(\mathbb{E}\left[||\mathbf{C}_s - f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})||_F^2\right]\right) \tag{4.18}$$

Recall that $\tilde{\mathbf{h}}_{\text{UL}} \in \mathbb{R}^{2M_1}$, (4.6), and $\tilde{\mathbf{h}}_{\text{DL}} \in \mathbb{R}^{2M_2}$, (4.7), as well as the definition of expectation, the objective in (4.18) could be re-written as

$$\min_{f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \frac{1}{2} \int \int ||\mathbf{C}_s - f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})||_F^2 \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}, \tilde{\mathbf{h}}_{\text{DL}})\, d\tilde{\mathbf{h}}_{\text{DL}} d\tilde{\mathbf{h}}_{\text{UL}}. \tag{4.19}$$

To find the optimal prediction function, the objective in (4.19) is differentiated with respect to $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ and set to zero

$$\begin{aligned}
\frac{\delta \mathbb{E}}{\delta f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} &= \frac{1}{2} \int \frac{\delta}{\delta f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})} \left\{ ||\mathbf{C}_s - f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})||_F^2 \right\} \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}, \tilde{\mathbf{h}}_{\text{DL}})\, d\tilde{\mathbf{h}}_{\text{DL}} d\tilde{\mathbf{h}}_{\text{UL}} && (4.20) \\
&= \int \left[\mathbf{C}_s - f_\Theta(\tilde{\mathbf{h}}_{\text{UL}}) \right] \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}, \tilde{\mathbf{h}}_{\text{DL}})\, d\tilde{\mathbf{h}}_{\text{DL}} && (4.21) \\
&= \int \mathbf{C}_s \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}, \tilde{\mathbf{h}}_{\text{DL}})\, d\tilde{\mathbf{h}}_{\text{DL}} - \int f_\Theta(\tilde{\mathbf{h}}_{\text{UL}}) \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}, \tilde{\mathbf{h}}_{\text{DL}})\, d\tilde{\mathbf{h}}_{\text{DL}} && (4.22) \\
&= \int \mathbf{C}_s \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}, \tilde{\mathbf{h}}_{\text{DL}})\, d\tilde{\mathbf{h}}_{\text{DL}} - f_\Theta(\tilde{\mathbf{h}}_{\text{UL}}) \mathbb{P}(\tilde{\mathbf{h}}_{\text{UL}}) = 0. && (4.23)
\end{aligned}$$

Using the definition $\mathbf{C}_s = \tilde{\mathbf{h}}_{\text{DL}} \tilde{\mathbf{h}}_{\text{DL}}^T$ and solving for $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ yields

$$f_\Theta(\tilde{\mathbf{h}}_{\text{UL}}) = \int \mathbf{C}_s \mathbb{P}(\tilde{\mathbf{h}}_{\text{DL}} | \tilde{\mathbf{h}}_{\text{UL}}) d\tilde{\mathbf{h}}_{\text{DL}} = \mathbb{E}[\tilde{\mathbf{h}}_{\text{DL}} \tilde{\mathbf{h}}_{\text{DL}}^T | \tilde{\mathbf{h}}_{\text{UL}}]. \tag{4.24}$$

Given condition 4 and similar to the definition in (4.4), the expectation in (4.24) is the definition of the conditional covariance $\mathbb{E}[\mathbf{C}_s | \tilde{\mathbf{h}}_{\text{UL}}]$ of $\tilde{\mathbf{h}}_{\text{DL}} | \tilde{\mathbf{h}}_{\text{UL}}$, which will be referred to as the *conditional sample covariance* to differentiate it from $\mathbf{C}$.

The function $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ in (4.24) is also a maximizer for (4.5). This is a direct consequence of the relation between $\mathbf{C}$ and $\mathbb{E}[E[\mathbf{C}_s | \tilde{\mathbf{h}}_{\text{UL}}]] = \mathbb{E}[\tilde{\mathbf{h}}_{\text{DL}} \tilde{\mathbf{h}}_{\text{DL}}^T | \tilde{\mathbf{h}}_{\text{UL}}]$; every element in $\mathbf{C}$ could be found in the $2M \times 2M$ conditional sample covariance, see Appendix C, and, hence, if $f_\Theta(\tilde{\mathbf{h}}_{\text{UL}})$ predicts $\mathbb{E}[\mathbf{C}_s | \tilde{\mathbf{h}}_{\text{UL}}]$, then it also predicts $\mathbf{C}$. □

The statement of Proposition 7 provides an analytical view on how one might asymptotically learn the target covariance using a machine learning model in a regression setting. In reality, a machine learning model cannot fully capture the target covariance, yet it could produce a good approximation of that covariance given the right settings. Section 4.9 will empirically verify that statement.

4.6.2 Proposed Solution

The choice of the function family $\mathcal{H}$ is critical to any machine learning problem as it defines, in part, the representational capacity of the machine learning algorithm and model [73]. As the nature of the relation between the target and observation is not known a priori as well as their data-generating distribution, a good choice in this case is to consider a broad family of functions and utilize artificial neural networks as the algorithm of choice to traverse that family space. As discussed in [106], neural networks, whether shallow or deep, are considered universal approximators [78][34]. They are expected to capture the relation between the target covariance and the observed channels, and, hence, they are the core of the proposed regression solution.

Network Architecture

The network architecture adopted in this work is a standard fully-connected (dense) architecture, shown in Fig. 4.4. It consists of Z stacks of layers that represent the building blocks of the architecture. The first $Z-1$ stacks comprise a sequence of dense, ReLU, and dropout layers. These stacks differ in the number of neurons Q they implement. They all learn to perform non-linear transformations to their corresponding inputs, taking them from one vector space into another. The last stack of the architecture, the Zth stack, comprises only a single dense layer. It learns to perform a linear transformation that produces the prediction of the neural network. This layer is followed by neither a ReLU nor a dropout; this is necessitated by the nature of the prediction vector, a real-valued output vector.

Pre-processing and Loss Function

The proposed neural network is designed to take in a channel vector and spit out a target covariance. However, for the network to perform its task, predicting target covariances,

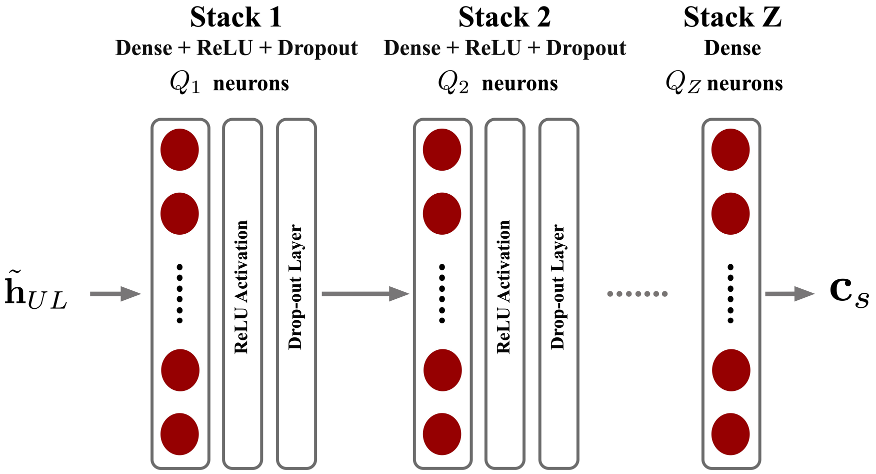


Figure 4.4: A depiction of the Z layer neural network used to learn conditional covariance prediction. The first $Z-1$ stacks comprise a sequence of dense, ReLU, and dropout layers with different number of neurons.

the channels and sample covariances need to undergo a pre-processing pipeline. The goals of this pipeline is two fold: (i) ensure that the input-output pairs are processed properly for efficient learning, and (ii) make sure those pairs are in suitable forms to be fed to the network. The following describes how the adopted pipeline prepares the final data samples:

- **Inputs:** Using the mother set $\mathcal{S}$, the first pre-processing component in the pipeline normalizes the uplink $\mathbf{h}_{\text{UL}}$ channels [79][80]. The channels are normalized to have a unity average element-wise power. In other words, the channel vectors are scaled by:

$$\Delta = \sqrt{\frac{1}{UKM_1}\sum_{u=1}^{U}\sum_{k=1}^{K}\sum_{m=1}^{M_1}|h_{m,k,u}|^2}, \tag{4.25}$$

  where $h_{m,k,u}$ is the mth element of the uth channel vector at the kth subcarrier. Since the dense network is the architecture of choice in this work, the inputs need to be in vector forms. Furthermore, as Proposition 7 shows, channels need to be in real-valued forms, which is interesting as modern day deep learning frameworks only support real-valued computations. The real and imaginary parts of the uplink channel vectors are stacked up to form $\tilde{\mathbf{h}}_{\text{UL}}$ as defined earlier in (4.6). As the considered system in Section 4.3 is OFDM, each user has multiple channel vectors across multiple subcarriers $\tilde{\mathbf{h}}_{\text{UL}}^{(k)}$, $\forall k \in \{1, \ldots, K\}$. These vectors are all concatenated in one high-dimensional vector $\tilde{\mathbf{h}}_{in} \in \mathbb{R}^{2M_1K}$, which is finally the input to the network.

- **Outputs:** Similar to the inputs, the pipeline prepares the sample covariances starting from the downlink channels $\mathbf{h}_{\text{DL}} \in \mathcal{S}$. The channels are normalized first, and, then, the real and imaginary parts are stacked as defined in (4.7). The sample covariances $\mathbf{C}_s$ are computed from those normalized and stacked channels $\tilde{\mathbf{h}}_{\text{DL}}$. In an OFDM system, each subcarrier produce a sample covariance $\mathbf{C}_s^{(k)}$ computed from $\tilde{\mathbf{h}}_{\text{DL}}^{(k)}$. The sample covariances of one user are averaged across subcarriers to produce $\mathbf{C}_s$. To attain stable and efficient training, the sample covariances are centralized by subtracting the element-wise average and, then, scaled by the element-wise standard deviation. Mathematically, this is expressed by:

$$\tilde{\mathbf{C}}_{s_{ab}} = \frac{\mathbf{C}_{s_{ab}} - \bar{\mu}}{\sigma_s}, \tag{4.26}$$

  where $\tilde{\mathbf{C}}_{s_{ab}}$ is the (a, b)th element of the standardized sample covariance matrix, $\bar{\mu}$ is the element-wise average of the target covariances $\bar{\mu} = \sum_{u=1}^{U} \sum_{a=1}^{2M_2} \sum_{b=1}^{2M_2} \mathbf{C}_{s_{ab}}$, and σ_s is the element-wise standard deviation $\sum_{u=1}^{U} \sum_{a=1}^{2M_2} \sum_{b=1}^{2M_2} (C_{s_{ab}} - \bar{\mu})^2$. The resulting standardized matrices are flattened into $4M_2^2$-dimensional vectors denoted $\tilde{\mathbf{c}}_s$. Noting that the target covariance $\mathbf{C}$ has half the number of entries of $\mathbf{C}_s$ (and $\tilde{\mathbf{C}}_s$), the final step combines those relevant entries from $\mathbf{C}_s$ to form an output vector $\mathbf{c}_s$ with a dimensionality equal to the number of entries in $\mathbf{C}$, i.e., some entries in $\tilde{\mathbf{c}}_s$ are combined such that the output vector has a dimensionality of $2M_2^2$.

The final training set obtained at the end of the pre-processing pipeline is $\mathcal{S}_{f_1} = \{(\tilde{\mathbf{h}}_{in}, \mathbf{c}_s)_u\}_{u=1}^{U}$.

Based on the statement of Proposition 7, the prediction function should be learned to minimize the MSE with the sample covariances. Hence, the training is carried with an average MSE loss:

$$\mathcal{L} = \frac{1}{U}\sum_{u=1}^{U} ||\mathbf{C}_s - f_\Theta(\mathbf{h})_{UL}||_F^2 \tag{4.27}$$

Section 4.8 will present more technical details on the training process.

4.7 Statistical Channel Prediction: A Clustering Approach

The second approach views the problem through a clustering lens. Such view is inspired by the work on joint spatial division and multiplexing in [47]. More specifically, the channel model in Section 4.3 is re-interpreted as a partitioning of the wireless environment into rings of scatterers. As shown in Fig. 4.3, each ring generates a *spatial channel cone* from the BS perspective, henceforth referred to as the channel cone. The key advantage of the channel-cone interpretation lies in the shared channel eigenspace across all users within a cone [47]– whether downlink or uplink channels. Mathematically, let the wireless environment be divided into a total of B channel cones, and let the channel covariance of the users within the bth channel cone be $\mathbf{C}_b$ where $b \in \{1, \dots, B\}$. The channels, whether downlink or uplink, of a user within the bth cone could be expressed using Karhunen-Loeve (KL) expansion [109]

$$\mathbf{h}_z^b = \mathbf{V}_z^b \left(\mathbf{\Lambda}_z^b\right)^{1/2} \mathbf{w}_z^b, \tag{4.28}$$

where $\mathbf{\Lambda}_z^b \in \mathbb{R}^{r_b \times r_b}$ is a diagonal matrix with the r_b positive eigenvalues of the channel covariance $\mathbf{C}_b$, $\mathbf{V}_{z_b} \in \mathbb{C}^{M_z \times r_b}$ is a matrix that has the eigenvectors associated with the eigenvalues in $\mathbf{\Lambda}_{z_b}$, $\mathbf{w}_{z_b} \in \mathbb{C}^{r_b \times 1}$ is a random combining vector following some distribution $p(\mathbf{w}_{z_b})$, and z is either DL for downlink or UL for uplink

Given the channel-cone interpretation, the task of predicting the conditional downlink covariance could be cast as a cluster-then-estimate problem; the prediction function attempts to learn the partitioning of the uplink channels into clusters representing each cone in the environment. Then, it learns an estimate to each conditional covariance. This approach is detailed in the following two sections.

4.7.1 Learning Target Covariance with Clustering

For the same problem defined in Section 4.5.1, the goal is to find a function $f_{\Theta}(\mathbf{h}_{\text{UL}})$ that maximizes the objective probability under the channel-cone model. This function should be able to discover the clustering of the uplink channels and predict the downlink covariance of each cluster. Such prediction process could be decomposed into two stages. The first one aims at identifying a *surface* that could separate the different clusters while the second must discover the downlink covariance of each cluster. This suggests that the function $f_{\Theta}(\mathbf{h}_{\text{UL}})$ is a composite of two functions, i.e., $f_{\Theta}(\mathbf{h}_{\text{UL}}) = (f^{(2)}_{\Theta_2} \circ f^{(1)}_{\Theta_1})(\mathbf{h}_{\text{UL}}) = f^{(2)}_{\Theta_2}(f^{(1)}_{\Theta_1}(\mathbf{h}_{\text{UL}}))$ where $\Theta = \{\Theta_1, \Theta_2\}$ is a set of two high-dimensional parameter vectors. $f^{(1)}_{\Theta_1}$ is a discriminant function separating the channel clusters, and $f^{(2)}_{\Theta_2}$ is an estimator function predicting the target covariance.

The bottleneck for the approach above is the clustering function $f^{(1)}_{\Theta_1}$. This is attributed to two main factors: (i) the users' membership (and hence the membership of the users' channels) to the channel cones composing the wireless environment is unknown; and (ii) the number of channel cones in a wireless environment is also unknown. Therefore, the main task of the function $f^{(1)}_{\Theta_1}$ (and the learning algorithm) is to uncover the cluster structure of the wireless channels, which could be seen as a latent variable. In other words, given only the mother dataset $\mathcal{S}$, the learning algorithm needs to discover the channel cones and the users' memberships to those cones. Such task lends itself to unsupervised (or self-supervised) learning by nature, which makes it challenging.

Once the clusters are uncovered and the users' channels are assigned to those cluster, the other task of learning the estimator function $f^{(2)}_{\Theta_2}$ could boil down to a simple per-cluster averaging of $\mathbf{h}^b_{\text{DL}} \left(\mathbf{h}^b_{\text{DL}}\right)^H$. More formally, let $\mathcal{S}_{t_2} = \{(\mathbf{h}_{\text{UL}}, \mathbf{h}_{\text{DL}}\mathbf{h}^H_{\text{DL}})_u\}_{u=1}^U$ be a dataset obtained from the mother set $\mathcal{S}$, and. let the mean of the downlink channels in the bth channel cone be zero, i.e., $\boldsymbol{\mu}_{\mathbf{h}_{\text{DL}}|b} = 0$. Using the learned $f^{(1)}_{\Theta_1}$, the set $\mathcal{S}_{t_2}$ could be broken down into

$$\mathcal{S}_{t_2} = \mathcal{S}^{(1)}_{t_2} \cup \mathcal{S}^{(2)}_{t_2} \cup \cdots \cup \mathcal{S}^{(B)}_{t_2} \tag{4.29}$$

where each subset $\mathcal{S}^{(b)}_{t_2}, b \in \{1, \ldots, B\}$ has the pairs $\{(\mathbf{h}_{\text{UL}}, \mathbf{h}_{\text{DL}}\mathbf{h}^H_{\text{DL}})_u\}^{U_b}_{u=1}$ that belong to the bth channel cone, and U_b is the total number of pairs in the bth channel cone. Then, the prediction function $f^{(2)}_{\Theta_2}$ can simply be given by

$$f^{(2)}(b) = \frac{1}{U_b} \sum_{u=1}^{U_b} \mathbf{h}_{\text{DL}_u} \mathbf{h}^H_{\text{DL}_u}. \tag{4.30}$$

The quality of the estimated covariance of (4.30) depends on the cardinality of $\mathcal{S}^{(b)}_{t_2}$, i.e., $|\mathcal{S}^{(b)}_{t_2}| = U_b$.

4.7.2 Proposed Solution

The proposed unsupervised solution rests on two components, an encoder built with a DNN and a k-means algorithm. In their original space, the observed channels are not expected to be linearly separable, and as such, identifying the clusters to which they belong becomes very difficult. The encoder is designed to learn a transformation into a high-dimensional space (embedding space) in which the channels could exhibit linear separability. The final clusters are, then, produced using a k-means algorithm applied to the transformed (or embedded) channels. These embedded channels are henceforth referred to as the features. The following three subsections will detail the components of the proposed solution and the pre-processing pipeline.

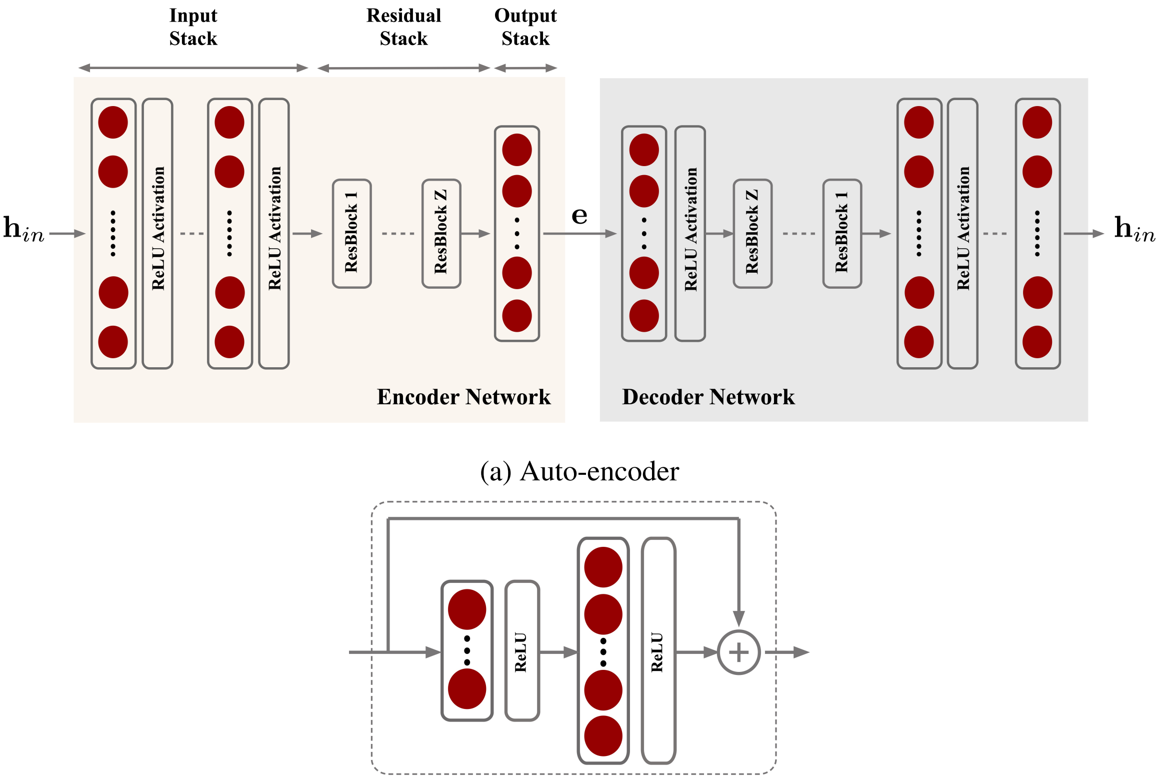
Encoder architecture

The encoder is designed to be one of two main networks of a stacked denoising autoencoder [73, 104], see Fig. 4.5a. This autoencoder has symmetric networks; the encoder and decoder have the same number of layers, and for each layer in the encoder, there is a layer in the decoder that inverts its operation. Each network is structured in three stacks. At the encoder side, there is an input stack build with Z_1-sequence of alternating dense and ReLU layers. A sequence of Z_2 residual blocks follows the input stack and constructs the residual stack. They feed into the output stack that consists of a single dense layer and outputs the feature vector. At the decoder side, the same three stacks are implemented but in reverse order, see Fig. 4.5a. The decoder differs from the encoder in the placement of some of its ReLU layers. In particular, the first layer of the decoder is followed with a ReLU while the last layer is not.

The middle stack (residual stack) is built from a two-layer residual module [6], see Fig. 4.5b. As depth is of importance to learning powerful representations [33][30], the residual blocks help increase the depth of the network without incurring training degradation [6]. A block has two dense layers each of which is proceeded with a ReLU, and the output of these layers is added to the input to produce the output of the block. In cases where the input and output are of different dimensionality, the path from the input to the sum operation (referred to as skip connection) implements a dense layer that learns to transform the input to the output space.

Producing clusters and covariances

The encoder in a trained autoencoder learns a non-linear transformation of the input channel $\mathbf{h}_{in}$ to some feature vector $\mathbf{e}$. In many many applications, the input vectors (channel in this paper) are not linearly separable in their original space. This makes the discovery of patterns (or clusters in this work) a difficult task. Hence, the ultimate goal of the encoder is to learn a transformation that result in linearly separable features, which could, subsequently, be separated rather easily. The algorithm of choice to learn those clusters in this work is k-means [31]. It is applied on top of linearly separable features.



(a) Auto-encoder

(b) Residual block

Figure 4.5: A Schematic Showing The Overall Autoencodeer Architecture in (a), and Describing The Inner Workings of The Residual Block in (b). The Encoder Network Is Finally Used to Extract a Feature Vector $\mathbf{e}$ for Every Input Channel $\mathbf{h}_{in}$.

The objective of this clustering approach is to predict a target downlink covariance given the observed uplink channels. Therefore, using the clusters uncovered by the encoder and k-means, a finite set of covariances is generated by averaging the sample covariances $\mathbf{h}_{\text{UL}}\mathbf{h}_{\text{UL}}^{H}$ of each cluster. These covariances are stored such that for any newly-observed channel vector, the learned encoder and k-means need only produce the cluster assignment of that vector.

Data Pre-processing

A similar pre-processing pipeline to that in Section 4.6.2 is adopted here. The details are described below:

- **Inputs:** The uplink channels from $\mathcal{S}$ are first normalized. Since the clusters are not commonly known, uplink channels alone may not be enough to learn the clusters. However, if every user position in the environment contributes P_{UL} uplink channels by sending P_{UL} pilots across K subcarriers, a rough estimate of the uplink covariance could be computed for each user. This estimate encodes some spatial information and, hence, is expected to help the unsupervised encoder and k-means discover the clusters. A *sample uplink covariance* is computed as follows:

$$\mathbf{H}_{u'} = \frac{1}{KP_{UL}} \sum_{u=(u'-1)P_{UL}+1}^{u'P_{UP}} \sum_{k=1}^{K} \mathbf{h}_{UL_u}^{(k)} (\mathbf{h}_{UL_u}^{(k)})^H, \tag{4.31}$$

  where $\mathbf{H}_{u'} \in \mathbb{C}^{M_1 \times M_1}$ is the u'th sample uplink covariance, $u' \in \{1, \dots, \tilde{U}\}$, and $\tilde{U} = U/P_{UL}$. Those sample covariances are complex-valued, so the next step in the pipeline is to convert them to real valued arrays, $\{\tilde{\mathbf{H}}_1, \dots, \tilde{\mathbf{H}}_{\tilde{U}}\}$ where $\tilde{\mathbf{H}}_{u'} \in \mathbb{R}^{M_1 \times M_1 \times 2}$. Since the encoder architecture is based on dense layers, the final step in the pipeline is to flatten those real valued arrays into single high-dimensional vectors. The resulting vector is $\mathbf{h}_{in} \in \mathbb{R}^{2M_1^2}$ and is used as the input to the encoder.

- **Outputs:** The downlink channels are also normalized first, but since the proposed solution is unsupervised, the downlink channels are not used to create target covariances. Instead, they are grouped into $\tilde{U}$ subsets such that each subset has the P_{UL} downlink channels corresponding to the P_{UL} uplink channels used to form $\tilde{\mathbf{H}}_{u'}$, $\forall u' \in \{1, \dots, \tilde{U}\}$.

The inputs and output are grouped into two sets $\mathcal{S}_{f_2} = \{\mathbf{h}_{in_{u'}}\}_{u'=1}^{\tilde{U}}$ and $\mathcal{S}_{f_3} = \{\mathcal{S}_{sub_{u'}}\}_{u'=1}^{\tilde{U}}$ where $\mathcal{S}_{sub_{u'}} = \{\mathbf{h}_{DL_u}\}_{u=(u'-1)P_{UL}+1}^{u'P_{UL}}$.

Encoder training

There is no specific training approach that guarantees producing linearly separable features; however, based on the work in [104], stacked denoising autoencoders are shown to produce *good* representations. As a result, this paper considers a denoising autoencoder followed by a k-means clustering algorithm to discover the channel clusters. The training is conducted using $\mathcal{S}_{f_2}$, and it goes through the following three stages:

- **Layer-wise greedy training:** The encoder has a symmetric architecture, and, hence, each matching pair of encoder-decoder layers are combined with a dropout layer in-between them to form a mini denoising autoencoder. Each mini autoencoder is then trained to reconstructs its input.

- **End-to-end autoencoder finetuning:** Using the trained mini autoencoders, the original autoencoder in Fig. 4.5a is re-assembled without the dropout layers. A second round of training is, then, conducted. It fine-tunes the parameters of the autoencoder.

- **K-means clustering:** When the end-to-end training is done, the encoder is used to extract feature vectors for all the data samples in $\mathcal{S}_{f_2}$. Those features are fed to the k-means algorithm along with an estimate of the number of clusters B. K-means is trained to cluster those samples and return B cluster centroids.

More details on the first two stages and their expected performance could be found in [73, 104]. Once the three stages are complete, the covariance of each clusters is calculated using the downlink channels in $\mathcal{S}_{f_3}$. The downlink channels in $\mathcal{S}_{f_3}$ are used to do that. The result is a finite set of covariances $\{\mathbf{C}_1, \dots, \mathbf{C}_B\}$.

4.8 Experimental Setup

Evaluating the proposed solutions requires datasets for training and validation. For that end, the first section describes the communication scenario considered for the evaluation experiments. The second section, then, introduces how the data is generated and how each solution is trained.

4.8.1 Communication Scenarios and Datasets

The communication scenario chosen in this paper is for a busy metropolitan street. It is a dynamic scenario provided by the DeepMIMO dataset [68], namely scenario "O1_dyn". The scenario is available at two sub-6 GHz frequencies, 3.4 GHz (O1_dyn_3p4) and 3.5 GHz (O1_dyn_3p5), and they are both used in this work. The scenario has a street populated with both stationary and dynamic objects. Fig. 4.6 shows a top-view of the street and its objects. The scenario has variety of buildings, high, medium, and low-rise, along both sides of the street and has variety of vehicles moving in both directions along the street at different speeds. The scenario has two massive MIMO basestations installed at opposite ends of the street. Each of the basestations implement a 64 ULA antenna, i.e., $M_1 = M_2 = 64$. They are serving a stationary user grid with 405 potential users. The users are spaced 1 meter away from each other, each one implements an omni-directional antenna.

Using scenarios "O1_dyn_3p4" and "O1_dyn_3p5" as well as the DeepMIMO generation script, the mother dataset $\mathcal{S}$ of uplink and downlink channels is generated, where channels at 3.4 GHz are considered uplink and 3.5 GHz are considered downlink. The generation hyper-parameters are listed in Table.4.1. Each user position in the scenario is considered a cluster. Therefore, the positions are also generated and used as "groundtruth" cluster indices. As described earlier in Section 4.6.2 and 4.7.2, the sets $\mathcal{S}_{f_1}$, $\mathcal{S}_{f_2}$, and $\mathcal{S}_{f_3}$ are generated from the mother set.

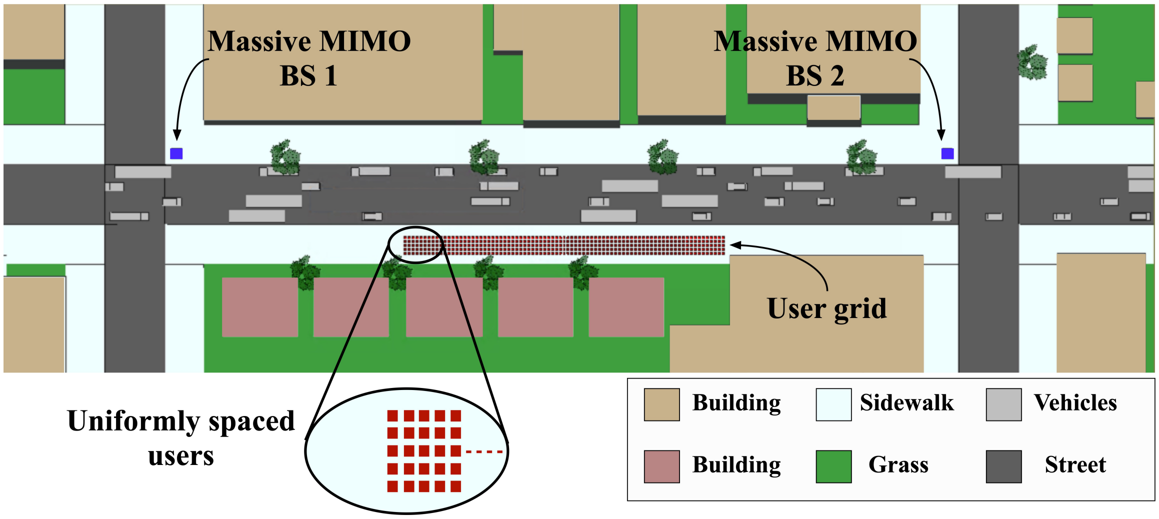


Figure 4.6: A Top-view of The Considered Scenario. It Shows a Snapshot of a Dynamic Environment with Moving Vehicles. It Shows The Positions of The Two Massive MIMO Basestations and The Uniformly-Spaced User Grid.

4.8.2 Architecture Details and Training

The two solutions rely on DNNs. Despite their matching objective, they have different architectures. The regression DNN has 5-stack architecture. The breadth of the dense layers are, respectively, 1024, 4096, 4096, 4096, and 8192. Three dropuot layers are inserted between the wide stacks, in particular between stacks: (i) 2 and 3, (ii) 3 and 4, and finally (iii) 4 and 5. On the other hand, the autoencoder used for the unsupervised solution has a deeper architecture than that of the regression network. Each of its networks has a total of 9 layers. At the encoder side, the input stack has 2 dense layers each followed by a ReLU non-linearity. The breadth of those layers are 4096 and 1024. The second stack (residual stack) is composed of 3 residual blocks, each of which has two dense layers (as in Fig. 4.5b) with the breadths of 512 and 1024 and implementing ReLU non-linearities. The final output stack has a single dense layer with 2048 neurons. The decoder network has almost the same layers but in reverse order, as explained in Section 4.7.2.

Table 4.1: DeepMIMO Hyper-parameters

Hyper-parameter	Value
Scenario name	“O1_dyn_3p4” and “O1_dyn_3p5”
Active BS	1 and 2
Active users	1 - 5
Number of BS antennas	(64,1,1)
Antenna spacing (wave-length)	0.5
Bandwidth (GHz)	0.02
Number of OFDM subcarriers	32
OFDM sampling factor	1
OFDM limit	16
Number of paths	15

Both solutions are trained on the generated datasets. All datasets are divided into training and validation sets with a split percentage of 70% to 30%. The regression network is trained to reduce the MSE loss on the training set and it is finally tested on the validation set. The training hyper-parameters are listed in Table.4.2 under the column “Regression.” The autoencoder undergoes the two-stage training strategy described in Section 4.7.2 on the 70% samples obtained from $\mathcal{S}_{f_2}$. Then, its output features are passed to the k-means algorithm to produce the clusters. Based on the clustering the covariances are estimated using $\mathcal{S}_{f_3}$. The second column of Table.4.2 lists the hyper-parameters for the autoencoder. Both DNNs are implemented using PyTorch [81] while k-means is implemented using Scikit-learn [110]. The training and testing for the two solutions took place on an RTX 2080 Ti running on a Linux system. Sample codes could be found at [111].

Table 4.2: Training Hyper-parameters

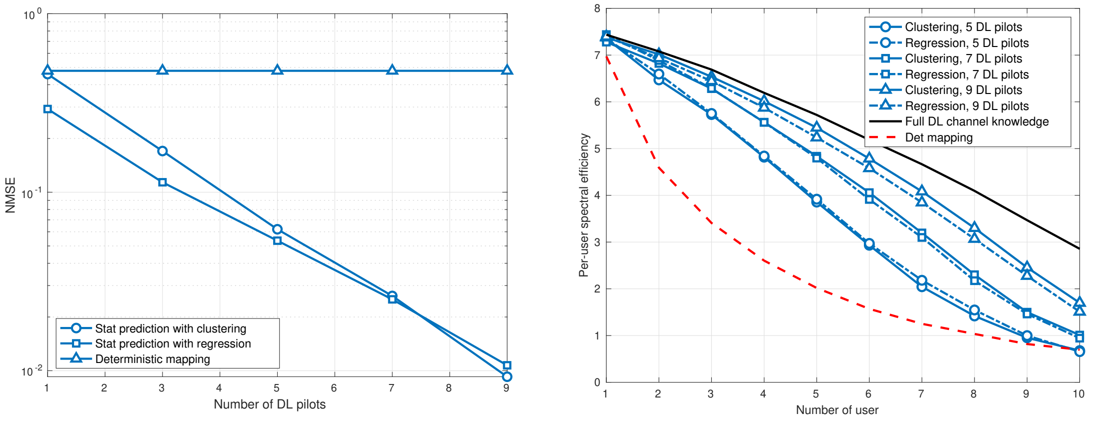
Training hyper-parameter	Regression	Clustering
Solver	Adam [84]	SGDM[2]
Learning rate	10^{-3}	10^{-1} (stage 1), 10^{-2} (stage 2)
Weight decay	0	0
Batch size	5000	5000
Number of Epochs	250	$120, 200$
Momentum	0	0.9
Learning rate factor	0.1	0.1
Learning rate schedule (@epoch)	20	100 (stage 1), 100 and 150 (stage 2)
Number of cluster	None	405

4.9 Experimental Results

The proposed statistical prediction framework is evaluated in this section. Both approaches, regression and clustering, are tested and compared to each other and to deterministic mapping. Using the generated dataset and the training settings in Section 4.8, a sequence of evaluation experiments are presented. They overall study the NMSE and beamforming gain performances of the proposed solutions in single user settings, and the per-user rate and sum rate performances in the cases of multi-users.

4.9.1 Moving to Statistical Prediction

The evaluation begins by empirically establishing the need for the statistical prediction framework. This is done by benchmarking both statistical solutions to a deterministic mapping solution—similar to that in [106]— in two communication settings, single and multi-user. The regression network proposed in Section 4.6.2 and detailed in Section 4.8.2



(a)

(b)

Figure 4.7: Predicting Downlink Channels form Uplink Ones at The Same Basestation. (a) Shows The NMSE Performance in Single-user Setting While (b) Shows The Per-user Spectral Efficiency with Different Number of Downlink (DL) Pilots

is modified to have 2048 neurons in the output stack. This modified network is, then, used as the deterministic mapping solution. The training hyper-parameters for that network are the same as those in Table.4.2. For the statistical approaches, a single UL pilot is assumed for the clustering solution, i.e., $P_{UL} = 1$. This is to have all three solutions on equal footing for comparison.

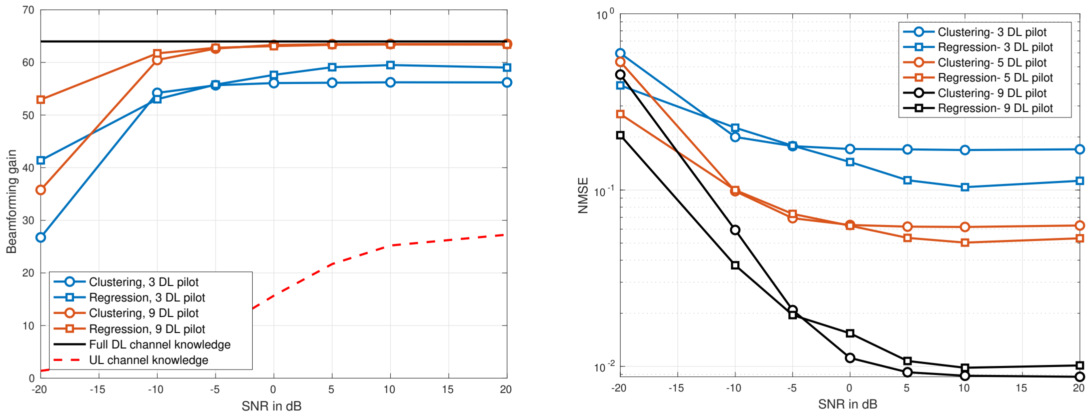
Using noiseless uplink channels at the first basestation, the three solutions are used to predict the downlink channels at the same basestation in a single-user setting, hereafter referred to as the in-place task. Fig. 4.7a shows the NMSE performance of all three versus the number of downlink pilots. Deterministic mapping requires zero pilots, and as such, it has a constant performance. Its NMSE does not match that of both statistical approaches, which is a clear drawback. The clustering and regression approaches show almost linear improvement as the number of pilots increases. With around 9 pilots, the NMSE of both statistical approaches drop down to the neighborhood of 10^{-2}. This is approximately 14% of the total number of pilots required by a classical channel-estimation approach given the number of antennas. As the number of pilots reduces, the performance of both approaches degrades, yet clustering takes a stronger hit than that taken by regression. A very likely reason for that is the learning approach each one follows. Regression is a supervised approach where the algorithm experiences desirable responses throughout training, which is not the case in unsupervised apprroaches.

The aforementioned discussion establishes the value of the statistical prediction framework; however to extend it further, the performance of all three solutions is studied in a multi-user setting with the in-place task. Such setting is expected to bring up the true color of statistical prediction, for channel estimation accuracy is critical in mitigating user interference. Fig. 4.7b plots the per-user spectral efficiency of each solution versus the number of users served simultaneously. Statistical solutions with different number of downlink pilots exhibit a quite interesting performance; with up to 5 users, statistical prediction achieves about 58% to 95% of the upper bound[3] using between 5 to 9 downlink pilots, respectively. In comparison, the deterministic solution performs very well with a single user yet degrades almost exponentially in the number of users. This is expected as the estimation quality does not have room for improvement in terms of number of pilots.

4.9.2 Effect of Downlink Pilots

With the need for the statistical prediction framework established, this section takes a deeper dive into the specifics of the framework. In particular, the effect of downlink pilots is studied in single-user setting and under noisy channel conditions. Similar to the previous section, a single uplink pilot is assumed for fairness of comparison, and the same in-place prediction task is considered. The only difference is having Additive White Gaussian Noise (AWGN) added to both uplink and downlink channels. Fig. 4.8a depicts the beamforming gain versus SNR for both solutions as well as the trivial choice of using uplink channels

[3] Achieved using full downlink channel knowledge and a zero-forcing precoder.



(a)

(b)

Figure 4.8: Performance of The Statistical Channel Prediction Solutions for an In-place Uplink-to-downlink Task and Under Noisy Conditions. (a) Depicts The Beamforming Gain Versus SNR While (b) Shows The NMSE Versus SNR. Both Are Plotted for Multiple Choices of Downlink (DL) Pilots.

as downlink approximates. It shows the robustness of both approaches in noisy settings. With 3 downlink pilots, both approaches set 2% shy of the upper bound at an operating SNR of -10 dB. This gap with the upper bound further shrinks at the same SNR when 9 pilots are used. When the SNR increases, both solutions gradually improve, closing the gap even further. Such behavior reflects very well on the spectral efficiency of both solution, especially under a low-SNR regime. A worthy note to raise here is the subtle improvement the regression solution exhibit over the clustering solution in a high SNR regime. This is a reflection of the supervision advantage the regression solution enjoys.

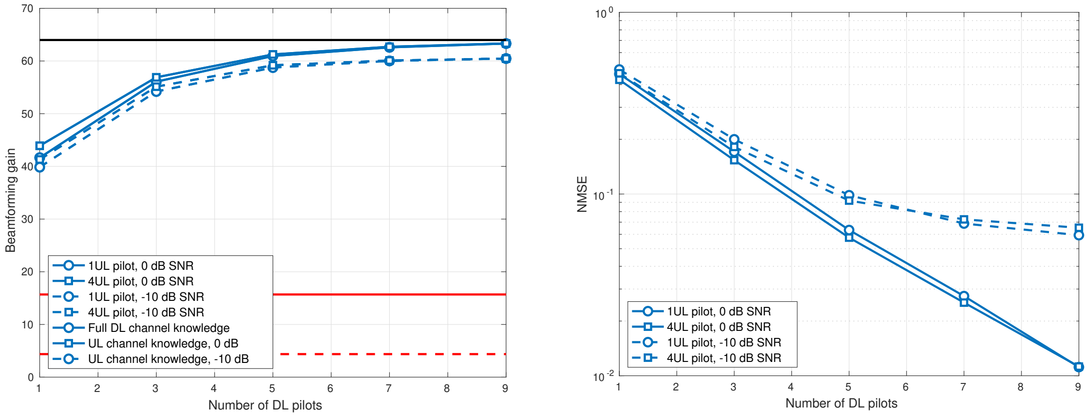
The performance of both solutions is next studied from the perspective of channel estimation quality. As beamforming gain is not a clear indicator of that quality, Fig. 4.8b plots the NMSE of both solutions versus SNR for different choices of downlink pilots. The figure reveals a new side of the statistical framework performance; the number of downlink pilots is very critical in combating harsh SNR conditions. With a very small number of pilots, like 3 pilots, the NMSE at -10 dB for both solutions is in the neighborhood of 0.2. This might not be a problem for a single user setting, as Fig. 4.8a has shown, but it is expected to reflect rather poorly on the performance in a multi-user setting. However, the problem could readily be remedied by a slight increase in the number of pilots. For instance, using 9 pilots at -10 dB drives down the NMSE to the vicinities of 0.06 and 0.04 for the clustering and regression solutions, respectively. This is almost a fall of 75% from the NMSE with 3 pilots. As the SNR level increases, the improvement gained from the slight increase of number of pilots becomes further clearer. For example, the NMSE with 9 pilots and at 0 dB drops around 90% from that with 3 pilots and at the same SNR.

4.9.3 Effect of Uplink Pilots

Since it follows an unsupervised learning approach, the clustering solution is equipped with an extra design parameter, which is the number of uplink pilots used to form its inputs. The effectiveness of this parameter is studied under the same experimental settings used in Section 4.9.2, i.e., an in-place prediction task with noisy channel conditions. Fig. 4.9a plots the beamforming gain of the clustering solution versus the number of downlink pilots for 2 choices of uplink pilots and 2 choices of SNR levels. The beamforming gain suggests that number of uplink pilots is of small importance; only when the number of downlink pilots is small does the increase of uplink pilots have some minor impact on the beamforming gain. For instance, at 0 dB SNR, requiring 4 uplink pilots results in a subtle improvement in the beamforming gain with one downlink pilot. The same could be observed at -10 dB SNR. This observation is backed up by the NMSE performance in Fig. 4.9b.

4.9.4 Channel Prediction Across Space and Frequency

Predicting downlink channels of one basestation using the uplink channels of another (or cross-space in-place) could be viewed as the most distinguishable and intriguing prop-

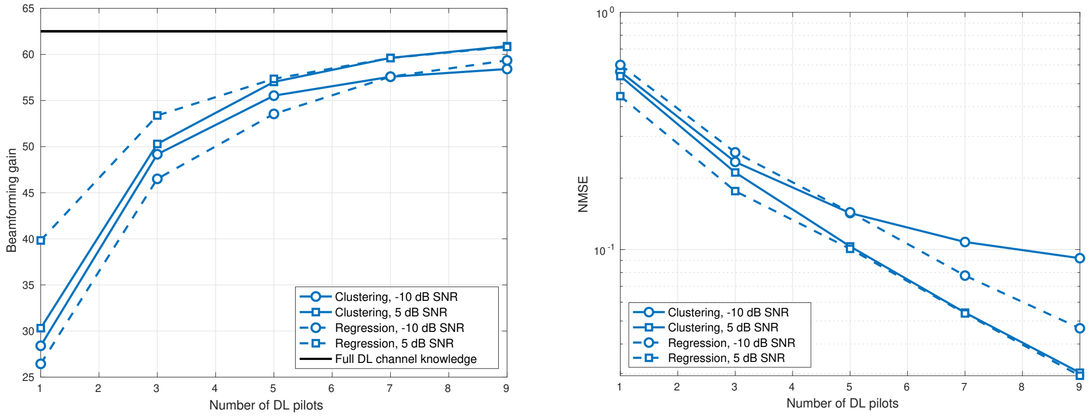


(a)

(b)

Figure 4.9: Performance of The Clustering Solution for an In-place Uplink-to-downlink Task and Under Noisy Conditions. (a) Depicts The Beamforming Gain Versus Downlink (DL) Pilots While (b) Shows The NMSE Versus Downlink Pilots. Both Are Plotted for 2 Choices of Uplink (UL) Pilots and SNRs.

erty of both deterministic and statistical frameworks. [106] has explored this property for the deterministic framework, and as such, the last set of experiments in this work is dedicated to studying it for the statistical framework. The beginning is with a single-user setting and noisy channel conditions. Fig. 4.10a shows the beamforming gain versus the number of downlink pilots for two different SNRs values—the clustering solution assumes a single uplink pilot. What immediately catches the eye in the figure is the good beamforming gain the two solutions achieve at two different SNR regimes with little training overhead; with only 5 pilots and an SNR level ranging between -10 and 5 dB, both solutions predicts **downlink covariances at another basestation** that achieve between 83% to 89% of the upper bound. This is quite interesting, for the fluctuation in the performance is very narrow (∼7%) for a 15 dB change in SNR. This reflects a good level of robustness for both solutions. The figure also confirms the advantage supervised learning have over unsupervised learning at high SNR. For instance, at 5 dB and with 3 downlink pilots, the regression



(a) (b)

Figure 4.10: Performance of The Statistical Channel Prediction Solutions for a Cross-space Uplink-to-downlink Prediction Task and Under Noisy Conditions. (a) Depicts Beamforming Gain Versus Downlink Pilots for 2 SNR Values, and (b) Shows NMSE Versus Number of Downlink Pilots for The Same 2 SNR Values As in (a).

solution slightly outperforms clustering by approximately 8%.

The good performance both solutions display extends to the channel-estimation quality. Fig. 4.10b shows the NMSE of the predicted channels versus the number of downlink pilots for the same two SNR values used in Fig. 4.10a. At high SNR, both solutions show consistent NMSE improvement as the number of pilots increases, and subtle lead regression has over clustering with small number of pilots persists. The consistent NMSE drop could also be observed at low SNR, but with a slightly worse overall performance compared to high SNR. The value of that NMSE performance is better reflected in a multi-user setting. To show that, Fig. 4.11 depicts the sum-rate spectral efficiency versus number of users for both solutions and under different number of downlink pilots and at 5 dB SNR. With small increments in the number of pilots, a massive MIMO system is capable of multiplexing more users at one basestation given only the uplink channels from another basestation. For instance, a central unit managing two massive MIMO basestations can multiplex 4, 5, and

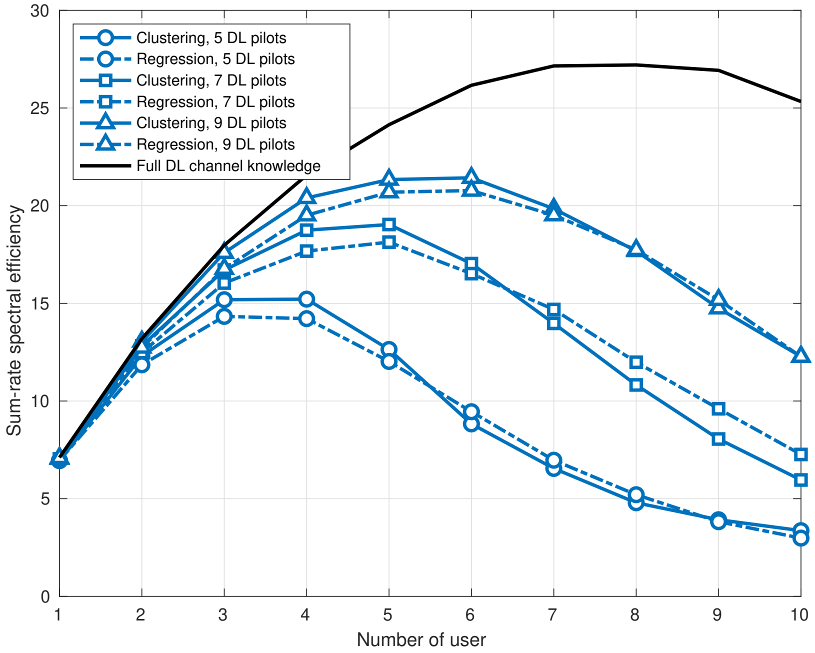


Figure 4.11: Sum-rate Performance for Cross-space Task Under Different Number of Downlink (DL) Pilots and at 5 dB SNR. It Extends The Advantage of Cross-space Prediction to The Multi-user Setting.

6 users at basestation 2 given their uplink channels observed at basestation 1 with 3, 7, and 9 pilots, respectively. Note that not only the number of multiplexed users increases with more downlink pilots but also their the sum-rate. With 3 pilots, 4 users are multiplexed with a sum-rate performance that is 32% away from the upper bound. However, 9 pilots allows more users and smaller sum-rate gap with the upper bound, approximately 16% away from that bound.

Part II

MULTIMODAL LEARNING

Part I of this dissertation has established the value of ML, and especially deep learning, to large-scale MIMO communications. Easy-to-acquire (or simply accessible) wireless data, like uplink channels, could be a great source of information about the wireless environment. Chapters 2, 3, and 4 have collectively demonstrated how deep learning could be used to extract and utilize such information to tackle challenges like downlink channel and beam training.

A common characteristic across all the work in those three chapters is the unimodality of its data, i.e., the developed algorithms learn from wireless data alone. This raises an interesting question on whether large-scale MIMO could benefit from other sources of information (multimodal data) or not. Such question could be seen as a natural extension to the work in Part I as it might hold some answers to many large-scale MIMO challenges; data sources like LiDAR sensors or RGB cameras provide contextual information about a wireless environment that, with the right ML algorithm, could help overcome some pervasive challenges. For instance, in high-frequency wireless networks, information about the motion of some objects in the environment can prove valuable to anticipate and prevent incoming LOS link blockage.

The second part of this dissertation recognizes the potential of multimodal data, and it attempts to investigate the role multimodal ML can play in large-scale MIMO. More specifically, it presents the Vision-Aided Wireless Communications (ViWiComm) framework which is a melting pot of deep learning, computer vision, and high-frequency communications.

Chapter 5

MILLIMETER WAVE BASESTATIONS WITH CAMERAS: VISION-AIDED BEAM AND BLOCKAGE PREDICTION

5.1 Scope and Contributions

Scope

An interesting and unconventional approach to handle challenges in high-frequency large-scale MIMO networks could be found in embracing a striking resemblance between high-frequency communication and computer vision systems, which is their reliance on LOS. High-frequency signals struggle in penetrating objects in the wireless environment and loose significant amount of power due to scattering [44]. Thus, there is a quite large SNR margin between LOS and NLOS communication links that skews in favor of LOS. This makes LOS a preferable setting in high-frequency communications, and it draws a connection with computer vision, which is inherently LOS. The data usually captured and analyzed in a computer vision system depict what is *visible* in the scene, starting with simple patterns (e.g., edges, colors,... etc) to abstract concepts (e.g., human, dog, tree,...etc). That information could be as valuable to a high-frequency system as it is to a computer vision system, begging the question:

> **Q.1:** *Could computer vision be used to mitigate some of the challenges in high-frequency large-scale MIMO?*

Contributions

The main objective of this chapter is to present the promise and potential ViWiComm has by addressing the beam and blockage prediction tasks using RGB, sub-6 GHz channels, and deep learning. When a pre-defined beam-forming codebook is available, learning beam prediction from images degenerates to an image classification task; depending on the user location in the scene, each image could be mapped to a class represented by a unique beam index from the codebook. On the other hand, detecting blockage in still images could be slightly trickier than beams as the instances of no user and blocked user are visually the same. Hence, images are paired with sub-6 GHz channels to identify blocked users. Each problem is studied in a single-user wireless communication setting.

5.2 Prior Work

The majority of the work adopting deep learning focuses on wireless sensory data to drive the learning and deployment of intelligent solutions, which begs the question of whether other forms of sensory data could be utilized to deal with the control overhead problem or not. Solutions like those in [64, 112–115] provide a *partially* positive answer to that question, where depth sensors are exploited to help wireless communication objectives. In this work, *Vision-Aided Wireless Communications* (ViWiComm) is presented as a new wholistic paradigm to tackle the overhead problem. It ultimately utilizes not only depth and wireless data, but also RGB images to enable mobility and reliability in mmWave wireless communications.

5.3 System and Channel Models

The following two subsections will present the system and channel models adopted throughout this chapter.

5.3.1 System model

Consider a system where a Base Station (BS), operating at both sub-6GHz and mmWave bands, is communicating with a single-antenna user, as depicted in Fig. 5.1. The BS

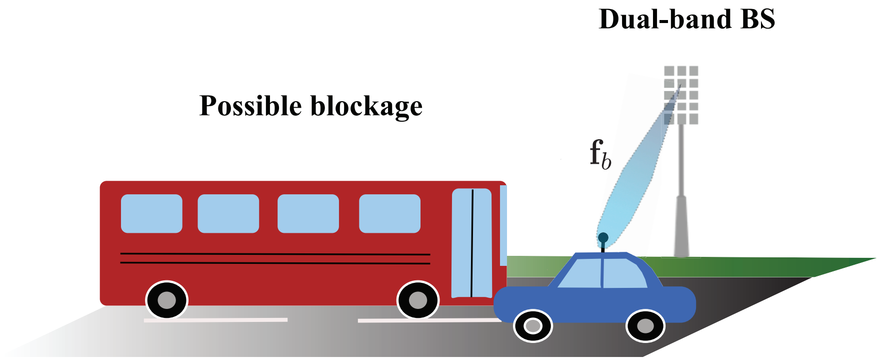


Figure 5.1: This Figure Shows a Downlink Communication Scenario Where The Base Station (BS) is Serving One User (The Car) Over The mmWave Band. The BS and User Are Equipped with Dual-band Sub-6GHz and mmWave Transceivers.

is assumed to be equipped with an M_{mmW}-element mmWave antenna array, an $M_{\text{sub-6}}$-element sub-6 GHz antenna array, and an RGB camera. The system adopts Orthogonal Frequency-Division Multiplexing (OFDM) with K_{mmW} subcarriers at the mmWave band and a $K_{\text{sub-6}}$ subcarriers at sub-6 GHz. Further, the mmWave BS systems is assumed to employ analog-only beamforming architecture while the sub-6 GHz transceiver is assumed to be fully-digital [40]. For mmWave beamforming, a beamforming vector is assumed to be selected from a pre-defined beam codebook $\mathcal{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_B\}$ where $\mathbf{f}_b \in \mathbb{C}^{M_{\text{mmW}} \times 1}$, $\forall b \in \{1, \dots, B\}$ and $B = |\mathcal{F}|$. To find the optimal beam, the user is assumed to send an uplink pilot that will be used to train the B beams and select the one that maximizes the user's average achievable rate, averaged across all subcarriers. This beam is then used for downlink data transmission. If beam $\mathbf{f}_b$ is used in the downlink to serve the uth user, then the received signal at the user's side can be expressed as

$$y_u^{\text{mmW}}[k] = \mathbf{h}_u^{\text{mmW}}[k]^T \mathbf{f}_b s_u^{\text{mmW}}[k] + n^{\text{mmW}}[k], \tag{5.1}$$

where $\mathbf{h}_u^{\text{mmW}}[k] \in \mathbb{C}^{M_{\text{mmW}} \times 1}$ is the mmWave channel of the uth user at the kth subcarrier, $\mathbf{f}_b$ is the bth beamforming vector in the codebook $\mathcal{F}$, $s_u^{\text{mmW}}[k]$ is the symbol transmitted on the kth mmWave subcarrier, and $n^{\text{mmW}}[k] \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2)$ is a complex Gaussian noise sample of the kth subcarrier frequency.

For blockage prediction, we assume that the BS will use the uplink signals on the sub-6 GHz band. If the mobile user sends an uplink pilot signal $s_u^{\text{sub-6}}[k] \in \mathbb{C}$ on the kth subcarrier, then the received signal at the BS can be written as

$$\mathbf{y}_u^{\text{sub-6}}[k] = \mathbf{h}_u^{\text{sub-6}}[k] s_u^{\text{sub-6}}[k] + \mathbf{n}^{\text{sub-6}}[k], \tag{5.2}$$

where $\mathbf{h}_u^{\text{sub-6}}[k] \in \mathbb{C}^{M_{\text{sub-6}} \times 1}$ is the sub-6 GHz channel of the uth user at the kth subcarrier, and $\mathbf{n}^{\text{sub-6}}[k] \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2_{\text{sub-6}}\mathbf{I})$ is the complex Gaussian noise vector of the kth subcarrier.

5.3.2 Channel model

This work adopts a geometric (physical) channel model for the sub-6 GHz and mmWave channels [40]. With this model, the mmWave channel (and similarly the sub-6 GHz channel) can be written as:

$$\mathbf{h}_u^{\text{mmW}}[k] = \sum_{d=0}^{D-1} \sum_{\ell=1}^{L} \alpha_\ell e^{-\jmath \frac{2\pi k}{K} d} p\left(dT_{\text{S}} - \tau_\ell\right) \mathbf{a}\left(\theta_\ell, \phi_\ell\right), \tag{5.3}$$

where L is number of channel paths, $\alpha_\ell, \tau_\ell, \theta_\ell, \phi_\ell$ are the path gains (including the path-loss), the delay, the azimuth angle of arrival, and elevation, respectively, of the ℓth channel path. T_{S} represents the sampling time while D denotes the cyclic prefix length (assuming that the maximum delay is less than DT_{S}). Note that the advantage of the physical channel model is its ability to capture the physical characteristics of the signal propagation including the dependence on the environment geometry, materials, frequency band, etc., which is crucial for considered beam and blockage prediction problems.

5.4 Problem Formulation

Beam and blockage predictions are interleaved problems for any mmWave system. However, for the purpose of highlighting the potential of ViWiComm, they will be formulated and addressed separately in this work.

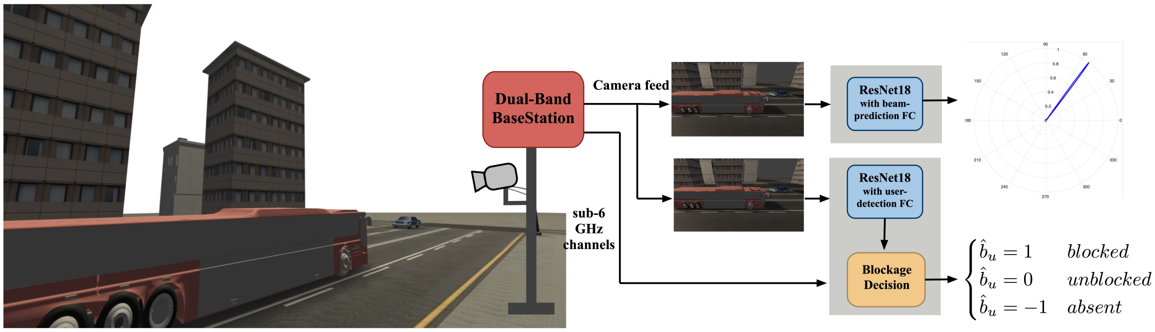


Figure 5.2: A Block Diagram of a Vision-Aided Dual-Band BS. Two ResNet18 Models Are Deployed to Learn Beam Prediction and User Detection, Respectively. Each Network Has a Customized Fully-Connected Layer That Suits The Task It Handles. A Network Is Trained to Directly Predict The Beam Index While The Other Predicts The User Existence (Detection) Which Is, Then, Converted to Blockage Prediction Using The Sub-6 GHz Channels.

5.4.1 Beam prediction

The main target of beam prediction is to determine the best beamforming vector $\mathbf{f}^\star$ in the codebook $\mathcal{F}$ to serve a user u. This is done such that the average achievable rate of that user, $R_u(\mathbf{f}^\star, \mathbf{h}_u^{\text{mmW}}[k]) \in \mathbb{R}^+$, is maximized. Formally, this beam is the solution of the following optimization problem:

$$\mathbf{f}^\star = \underset{\mathbf{f}\in\mathcal{F}}{\text{argmax}} \frac{1}{K_{\text{mmW}}} \sum_{k=1}^{K_{\text{mmW}}} \log_2 \left(1 + \rho \left|\mathbf{f}^T \mathbf{h}_u^{\text{mmW}}[k]\right|^2\right), \tag{5.4}$$

where ρ is the signal to noise ratio.

In this work, the problem is viewed from a different perspective than that in the literature; the selection process depends on the camera feed instead of the explicit channel knowledge (i.e., $\mathbf{h}^{\text{mmW}}[k]$) or beam training– both requiring large overhead. The optimal $\mathbf{f}^\star$, in this work, is found using an input image $X \in \mathbb{R}^{H\times W\times C}$, where H, W, and C are, respectively, the hight, width, and number of color channels of the image. This is done using a *prediction function* $f_{\Theta}(X)$ parameterized by a set of parameters Θ and outputs a probability distribution $\mathcal{P} = \{p_1, \dots, p_B\}$ over the vectors of $\mathcal{F}$. The index of the element with maximum probability in $\mathcal{P}$ determines the index of the predicted beam vector in $\mathcal{F}$. Formally, this expressed by:

$$n = \underset{n \in \{1,\dots,B\}}{\text{argmax}} \; \{p_1, \dots, p_n, \dots, p_B\}, \tag{5.5}$$

such that the predicted beam $\hat{\mathbf{f}} = \mathbf{f}_n \in \mathcal{F}$. The prediction function $f_{\Theta}(X)$ should be chosen to maximize the probability of correct prediction given an image X for any user in the communication environment. Formally, this is given by:

$$\max_{f_{\Theta}(X)} \prod_{u=1}^{U} \mathbb{P}_u \left(\hat{\mathbf{f}} = \mathbf{f}^{\star} | X \right), \tag{5.6}$$

where U is the total number of users in the environment. Note that the product in (5.6) is a result of a conditional independency assumption, i.e., the probability of correct beam prediction for the uth user is conditionally independent from other users' prediction probabilities given its image.

5.4.2 Blockage prediction

Determining whether a user's LOS link is blocked or not is a key task to boost reliability in mmWave systems. LOS status could be assessed based on some sensory data obtained from the communication environment. Examples of that are RGB images and sub-6 GHz channels, which are the sensory data of choice in this paper. Hence, let $(X, \mathbf{h}_u^{\text{sub-6}}[k])$ be the pair of an RGB image of the scene and the user's sub-6 GHz channels, and let $b_u \in \{-1, 0, 1\}$ be the actual LOS status, where 1, 0, and -1 refer to the statuses: blocked link, unblocked link, and absent user. In similar spirit to beam prediction, the target of the system is to predict with high probability the status of the user $\hat{b}_u$ given $(X, \mathbf{h}_u^{\text{sub-6}})$ using a prediction function $G_{\Theta}(X, \mathbf{h}^{\text{sub-6}})$, which can be expressed with the following optimization problem:

$$\max_{G_\Theta(X,\mathbf{h}^{\text{sub-6}})} \prod_{u=1}^{U} \mathbb{P}\left(\hat{b}_u = b_u | (X, \mathbf{h}_u^{\text{sub-6}})\right), \tag{5.7}$$

where U is the total number of user positions. Note here that the product of $\mathbb{P}\Big(\hat{b}_u = b_u | (X, \mathbf{h}_u^{\text{sub-6}})\Big)$ is a result of the assumption that the LOS status of a user position is conditionally independent from that of other positions. Despite that this assumption may not be accurate, it is a helpful simplification of the problem.

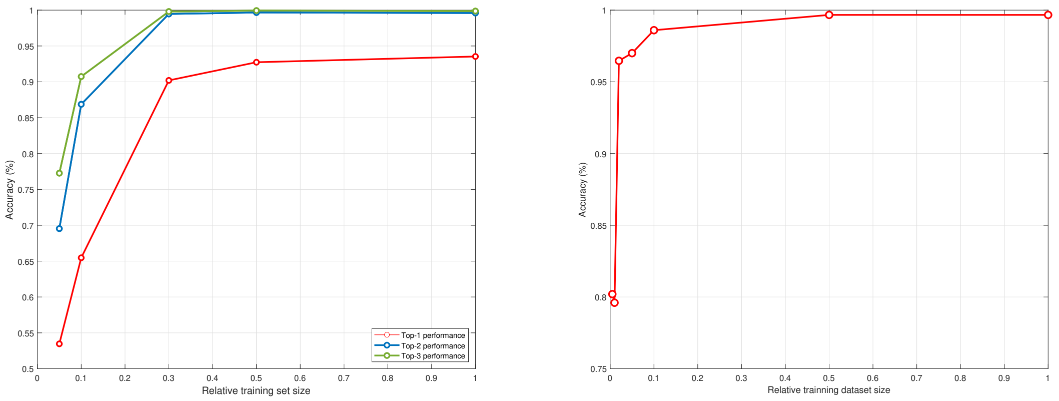
5.5 Proposed Camera-Based Solutions

Two deep learning based solutions are proposed for the two problems. They both rely on deep convolutional networks and the concept of transfer learning. The cornerstone in each is the 18-layer Residual Network (ResNet-18) [6] that is trained on the popular ImageNet2012 [29] and fine-tuned for the problem of interest. Figure 5.2 depicts a block diagram of the two solutions, and the following two subsections present their details.

5.5.1 mmWave beam prediction

The idea of predicting the best beamforming vector from a codebook using an image has a strong analogy with image classification; the beam vectors divide the scene (spatial dimensions) into multiple sectors, and the goal of the system is to identify to which sector a user belongs. Clearly, assigning images to classes labeled by beam indices is possible in LOS situations as it relies on the knowledge of the user's location in the scene. Hence, the objective is to learn the class-prediction function $f_\Theta(X)$, see Section 5.4.1, using images from the environment.

The proposed approach to learn the prediction function is based on deep convolutional neural networks and transfer learning. A pre-trained ResNet-18 model is adopted and customized to fit the beam prediction problem; its final fully-connected layer is removed and



(a) (b)

Figure 5.3: The Performances of The Proposed Solutions Are Shown in (a) and (b). The Former Shows The Results for Beam-prediction While The Latter Shows The Results for User Detection. Both Figures Present Their Respective Accuracies Versus Relative Training Set Size.

replaced with another fully-connected layer with a number of neurons equal to the codebook size, B neurons. This model is then fine-tuned, in a supervised fashion, using images from the environment that are labeled with their corresponding beam indices. It basically learns the new classification function (i.e., $f_{\Theta}(X)$), that maps an image to a beam index. The training is conducted with a cross-entropy loss given by:

$$l = \sum_{i=1}^{B} t_i \log p_i, \tag{5.8}$$

where t_i is 1 if i is the beam index and 0 otherwise. p_i is the probability distribution induced by the soft-max layer.

5.5.2 Link-blockage prediction

The blockage prediction problem is not very different from beam prediction in terms of the learning approach; it relies on detecting the user in the scene, and, thus, it could be viewed as a binary classification problem where a user is either detected or not. This, from a wireless communication perspective, is problematic as the absence of the user from the *visual* scene does not necessarily mean it is blocked; it could simply mean that it does not exist. As a result, this paper proposes integrating images with sub-6 GHz channels to distinguish between absent and blocked users.

A valid question might arise at this point: why would the system not predict the link status from sub-6 GHz channels directly? This is certainly an interesting question, and the work in [80] has shown that neural networks can effectively learn blockage prediction from sub-6 GHz channels. However, a major issue with that approach is its need for labeled channels; there is no clear signal processing method for labelling sub-6 channels as blocked or not, and, on the other hand, labelling images is relatively easier. Therefore, a network trained to detect users could help predict blockages from still images when it is combined with sub-6 GHz channels. This approach could be used to label sub-6 GHz channels and use them later for training model like those in [80].

Blockage prediction here is performed in two stages: i) user detection using deep neural network, and ii) link status assessment using sub-6 GHz channels and the user-detection result. The neural network of choice for this task is also a ResNet-18 but with a 2-neuron fully-connected layer. Similar to Section 5.5.1, it is pre-trained on ImageNet data and fine-tuned on some images from the environment. It is first used to predict whether a user exists in the scene or not. If a user is detected, the link status is directly declared as unblocked. On the other hand, when the user is not detected, sub-6 GHz channels come into play to identify whether this is because it is blocked or it does not exist. When those channels are not zero, this means a user exists in the scene and it is blocked. Otherwise, a user is declared absent.

Table 5.1: Hyper-parameters for Channel Generation

Parameter	Value	
Name of scenario	dist_cam	colo_cam_blk
Active BSs	3	1
Active users	1 to 5000	1 to 5000
Number of antennas (x, y, x)	(64,1,1)	(128,1,1)
System BW	0.5 GHz	0.5 GHz
Antenna spacing	0.5	0.5
Number of OFDM sub-carriers	512	512
OFDM sampling factor	1	1
OFDM limit	64	64
Number of paths	5	5

5.6 Simulation Results

For the sake of emphasizing their potential, the two solutions are independently tested. Two datasets of synthetic data samples are used in these tests as, currently, there is no publicly-available dataset that combines real-world images and wireless channels. The following few subsections discuss the datasets, training of the neural networks, and their performance evaluation.

5.6.1 Scenario and datasets

The publicly available ViWi framework [?] is used to generate the datasets for testing the beam and blockage prediction solutions. ViWi provides four single-user communication scenarios and a data generator script. Two of those four are chosen for evaluation, namely the direct distributed-camera and blockage co-located-camera scenarios.

Table 5.2: Hyper-parameters for Network Fine-tuning

Parameter	Value	
Batch size	150	150
Learning rate	1×10^{-4}	1×10^{-4}
Weight decay	1×10^{-3}	1×10^{-3}
Learning rate schedule	epochs 4 and 8	epochs 4 and 8
Learning-rate reduction factor	0.1	0.1
Data split (training-testing)	70%-30%	70%-30%

The direct distributed-camera scenario is used to generated data samples for the beam prediction experiments. The generated dataset has 5000 images and their corresponding mmWave channels; for each image depicting a user at some location, the corresponding mmWave channels of that user are generated using the generator package of ViWi. Table 5.1 gives a summery of the channels generation hyper-parameters. An important point needs to be mentioned here. When generating the image-beam dataset, every image is paired with a beam from the codebook of the *serving* BS, which is the one that *sees* the user.

For blockage prediction experiments, the blockage co-located-camera scenario is used. A dataset of 5000 images is generated but without any mmWave or sub-6 GHz channels. The reason behind that lies in the role the neural network is playing in the blockage prediction solution. Its main job is to learn to recognize the user's existence, which only requires training with the RGB images of the scenario.

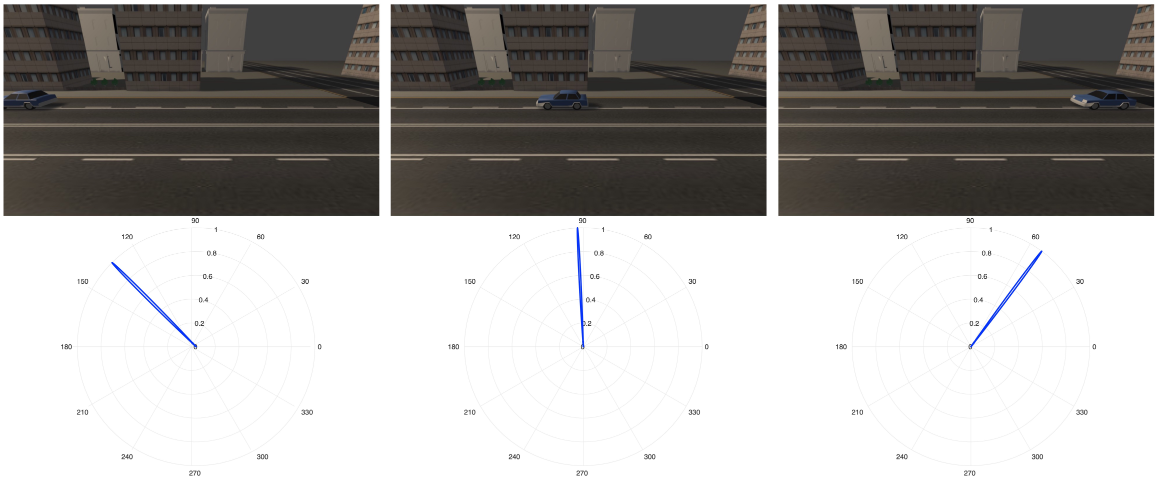


Figure 5.4: A Visualization of The Neural Network Inputs and Outputs When It Is Deployed for Beam Prediction. For Each Column, The RGB Image Showing The Location of The User (Car in The Image) Is Fed to The Trained Network, and The Result Is a Beam Index with The Pattern Shown Below The Image.

5.6.2 Network training

For both experiments, ResNet-18 is customized by removing the last fully-connected layer and replacing it with either a 64-neuron (for beam prediction) or 2-neuron (for user detection) fully-connected layers. Each of the two new layers is initialized from a normal distribution with zero-mean and unit variance. The network, then, is fine-tuned on the training subset of one of the two datasets describe above. The training hyper-parameters, including the dataset split, are listed in Table 5.2. Codes for the beam prediction experiment are made available at [83].

5.6.3 Prediction performance

The ability of the neural network to predict beams from images is examined by studying the top-1, 2, and 3 accuracies[1] versus the number of training samples. Figure 5.3-a shows the results of such test. The network shows good prediction performance with very little training samples, i.e., the accurate label is its first prediction around 90% of the time after training with only 0.3 samples of the total training set size (1500 out of 3500). This gets improved further when the top-2 and 3 best predictions are considered; the accuracy jumps to almost 100% with the same number of training samples. Top-1 accuracy continues to improve with more training data, and it hits 94% when the whole training set is used.

For blockage prediction, the critical point is identifying the user's existence in the scene. As such, Figure 5.3-b depicts the user detection accuracy of a fine-tuned ResNet-18 versus training dataset size. It is evident that the network is capable of learning such task very well with little training; it requires a little less than 0.05 of the training samples (175 samples out of 3500) to produce an accuracy of around 96%. Again, with more training samples, this accuracy approaches 100%, e.g., in Figure 5.3-b, accuracy is around 99% with half the training samples.

From a practical point of view, these numbers may not be very reflective if scenarios with dynamic environment are considered. However, they hint at the great boost a mmWave system could get in supporting mobility and maintaining reliability shall visual-perception be incorporated, which is the objective of this paper.

[1] They are the complements of top-1, 2, and 3 errors commonly used as metrics for quantifying classification accuracy. See [**?**] and [29]

Chapter 6

TRANSMITTER IDENTIFICATION VIA DEEP LEARNING: ENABLING MULTIUSER VISION-AIDED 6G COMMUNICATIONS

6.1 Scope and contribution

Scope

The framework of Vision-Aided Wireless Communications (ViWiComm) [116, 117] stands out among the proliferating machine learning approaches and frameworks for wireless communications; the research into machine learning for the wireless communications is dominated by unimodal learning that is tailored to utilize wireless data alone. As such, ViWiComm deviates from that by introducing the concept of multimodal learning to wireless communications. This is done by pairing visual and wireless data. More specifically, ViWiComm brings to the table a new range of capabilities that may not be available with wireless data alone. A good and very important example is proaction; a machine learning algorithm within the ViWiComm framework could anticipate adversarial events and take proactive mitigation measures. For instance, consider a typical mmWave communication system with a basestation and at least a single mobile device. The LOS link between that basestation and the device is of great importance to the quality of service, and, hence, any possible blockage of that link by any object could throw off the system performance. With ViWiComm, however, LOS blockages could be anticipated through a form of scene understanding and proactive mitigation measures like user hand off could be initiated [64, 116]. Aside from its originality and potential, the ViWiComm framework faces a critical challenge to its practicality in real wireless communication settings. The roots of that challenge are found in the ability of ViWiComm to handle situations with multiple candidate radio transmitters. To illustrate that, consider again the example of mmWave LOS blockage. In reality and from the basestation perspective, the surrounding could be full of objects that could constitute either possible mmWave transmitters or LOS blockages. Therefore, a machine learning algorithm needs to demonstrate a heightened level of understanding to the scene in order to be able to anticipate LOS blockages effectively. In particular, it needs to discern which object is the source of the signal and which one is the possible LOS blockage. This need gives rise to the following important question:

> **Q.1:** *How could a machine learning algorithm identify the transmitter responsible for the wireless signal in the visual data (images, video frames,...etc)?*

Answering such question is in the core of the this paper; it attempts to address that question by first defining the novel task of *transmitter identification* and, then, developing a ViWiComm solution for that task using Deep Neural Networks (DNNs)

Contribution:

The following points provide a rundown of the contributions:

- **A new ViWiComm wireless communication task:** we define the task of transmitter identification as a new fundamental task for ViWiComm. The task revolves around capitalizing on visual and wireless data to answer the following two questions: (i) Does a radio transmitter exist in the visual data? and (ii) if it does, which object is it?
- **A deep learning solution:** we propose a two-stage DNN architecture for the identification problem. The solution taps into the success of deep learning in computer vision and multimodal learning. The architecture is developed on a vision-wireless dataset collected from real wireless communication environments.
- **A vision-wireless development dataset:** Due to the novelty of the task and in recognition to the value of publicly available datasets, we construct a bimodal vision-

wireless dataset collected form real wireless communication environments and make it publicly available[1]. This is done by first building a complete vision-aided wireless mmWave communication testbed operating in the 60 GHz frequency band. We deploy that testbed in various locations with different types of candidate transmitters, and from each location, we collect tuples of RGB frames, mmWave beamforming vectors[2], and received power as data samples.

6.2 Literature Review

Recent years have seen an increasing interest in machine learning (or artificial intelligence) as a driving power for many future wireless communication technologies. This is evident in the multitude of wireless problems that are addressed using a form of machine learning, i.e., supervised, unsupervised, or reinforcement learning. This interest, in a broad sense, could be traced back to two key factors that machine learning enables, which are data-driven adaptability and multimodal learning. The work on machine learning for wireless communications could be divided into two categorizes based on the type and number of modalities of the learning data. The following two subsections provide a concise overview of that literature.

Unimodal learning from wireless data: Many wireless communication challenges addressed using machine learning utilize unimodal learning based solely on wireless data. For instance, [118] tackles the problem of channel-training overhead in mmWave MIMO communications. It trains a DNN to learn the best beamforming vector for downlink communication using observed uplink channels. That paper lays the groundwork for the holistic framework of channel mapping proposed in [106]. On the blockage prediction front,

[1] The publishing time hinges on the decision about this paper.

[2] A beamforming vector is a complex-valued vector representing the phases and amplitudes of the different elements of a mmWave antenna array, see [52] for more information.

the work in [80] proposes to utilize sub-6 GHz (low frequency) channels for identifying LOS and NLOS mmWave links. Similarly, [62] develops an LSTM-based architecture to perform the same task but for sub-6 GHz radio transmitters. [45] takes a more proactive approach to blockage prediction; it uses sequences of mmWave beamforming vectors to predict whether a moving transmitter is heading towards a stationary blockage or not. A major concern with this line of research is its inability to exploit rich sources of information, which majorly results in reactive decision making.

Bimodal learning from vision and wireless data: A more recent direction of research on machine learning for wireless communications is centered around the concept of bimodal learning, in particular, learning from vision and wireless data. It aims to enable proaction by adding a rich source of information such as visual data to the learning process. The work in [116, 117] first proposed the ViWiComm framework, in which RGB images and video frames are used to aid the communication system. For example, [116] utilizes RGB images and sub-6 GHz channels to address the problems of mmWave beam and blockage prediction. However, RGB frames are not the only rich source used in this direction of research; the work in [64] introduces link blockage prediction based on depth maps and received signal strength. All that work focuses on communication problems with single-candidate transmitter. This invokes a critical question on the performance of ViWiComm in real communication environments. Those environments usually have multiple candidate transmitters, and the ability to distinguish those transmitters is surely needed. The work in [119] takes the first step towards addressing that question; it presents an approach for identifying the transmitting radio equipment using vision and wireless data. Despite the novelty, the proposed approach is lacking in terms of practicality because it relies on visually detecting the transmitter equipment itself. This compromises its ability in real environments, for such equipment is usually invisible or hard to detect visually (in a person's hand or pocket or inside a vehicle).

A similar line of work to ViWiComm is the research direction on vision-wireless sensing. It attempts to utilize bimodal learning to address machine learning tasks in computer vision. For instance, human activity recognition (basic activities like sitting, walking, and standing) is addressed in [120] using RGB cameras and a commercial WiFi device. Vision and wireless data in [121, 122] have been shown to complement one another in learning interwind tasks such as through the wall pose estimation and action recognition. [121] utilizes RGB frames in a teacher architecture to train a wireless-based pose estimator to be a stand-alone estimator for invisible objects. [122], on the other hand, builds on top of that pose estimation to perform action recognition for invisible or partially occluded objects. Another interesting use of vision-wireless data could be seen in [123]. In that work, a single camera and multiple wireless receivers are utilized to collect bimodal data and do object localization. Despite the various tasks studied by vision-wireless sensing, its outcomes do not directly service the objectives of a wireless communication system. Hence, it could be considered an adjacent research direction to ViWiComm

6.3 Transmitter Identification

A machine learning algorithm in the ViWiComm framework is expected to learn from the observed visual and wireless data how to perform a wireless communication task. The performance of such algorithm is contingent on its ability to recognize objects transmitting a radio signals (henceforth referred to as *transmitters*) and those that do not transmit (henceforth referred to as *distractors*). This ability could be learned implicitly through the training on a wireless task or learned explicitly by posing it as a task in itself. The latter is the approach of choice in this work, and it is termed the *transmitter identification* task.

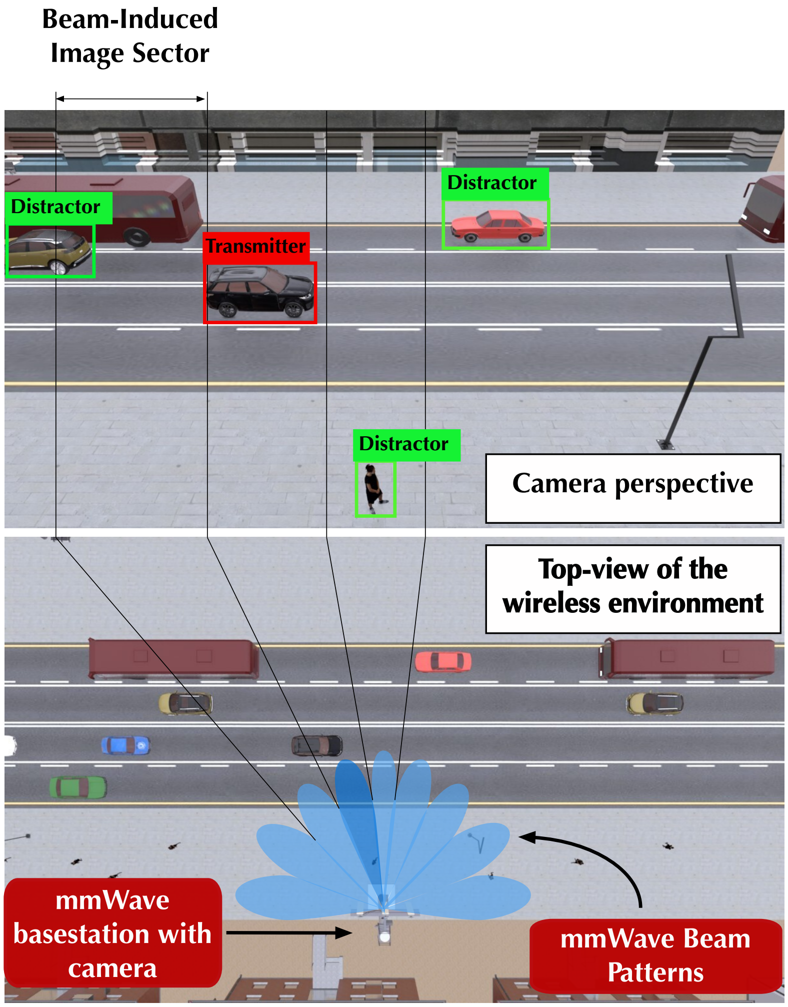

Beam-Induced Image Sector
Distractor
Transmitter
Distractor
Distractor
Camera perspective
Top-view of the wireless environment
mmWave basestation with camera
mmWave Beam Patterns

Figure 6.1: Two Views for The mmWave Wireless System. The Bottom Image Shows The Basestation and The Patterns of Its Codebook. The Top Image Shows The Environment from The Camera Perspective and Shows The Beam-induced Sectoring.

6.3.1 Communication System Model

The transmitter identification task in this paper is posed and studied in a mmWave communication system; this choice is majorly motivated by two facts. The first is the role mmWave plays in shaping the future of large-scale MIMO communications. It is considered a key component to modern (5G) and future wireless communication systems, see for instance [16, 22]. The second fact is the nice parallel mmWave communication has with vision systems. Due to its relatively high frequency (30-300 GHz), mmWave systems are heavily dependent on LOS, which an intrinsic property of any vision system.

The communications system model considered in this paper comprises: (i) A basestation equipped with an RGB camera and an M-element Uniform Linear Array (ULA) operating at a mmWave frequency band, and (ii) a mobile mmWave transmitter equipped with a single-element antenna. Fig.6.1 shows an illustration of this system. The basestation adopts a beam-steering codebook $\mathcal{F} = \{\mathbf{f}_q\}_{q=1}^{Q}$ where $\mathbf{f} \in \mathbb{C}^{M\times 1}$, see [124] for more information. Using this codebook, the signal radiated by the transmitter and received at the basestation could be expressed as

$$r = \mathbf{f}_{\text{opt}}^{H}\mathbf{h}s + \mathbf{f}_{\text{opt}}^{H}\mathbf{n} \tag{6.1}$$

where $s \in \mathbb{C}$ is the transmitted symbol satisfying $\mathbb{E}\left[|s|^2\right] \leq P$, $P \in \mathbb{R}$ is the power budget per symbol, $\mathbf{h} \in \mathbb{C}^{M\times 1}$ is the mmWave uplink channel, $\mathbf{f}_{\text{opt}}$ is the best beamforming vector in $\mathcal{F}$ maximizing

$$\mathbf{f}_{\text{opt}} = \underset{\mathbf{f}\in\mathcal{F}}{\text{argmax}}|\mathbf{f}_q^{H}\mathbf{h}|^2, \tag{6.2}$$

and $\mathbf{n} \in \mathbb{C}^{M\times 1}$ is an i.i.d. complex Gaussian noise vector with each element drawn from $\mathcal{N}_C(0, \sigma^2)$.

The mmWave channel, or $\mathbf{h}$, characterizes the propagation paths from a mobile transmitter to each ULA element of the basestation at any time instance t. Such channel could be described using a geometric model [52], which is expressed as

$$\mathbf{h} = \sum_{\ell=1}^{L} \alpha_\ell \mathbf{a}(\phi_\ell) \tag{6.3}$$

where L is the number of propagation paths, α_ℓ is the complex gain of the ℓth path, and $\mathbf{a}(\phi_\ell)$ is the array response vector, see [124].

6.3.2 Problem Definition

With the system model above in mind, the transmitter identification task is defined in this subsection. A general definition of the task is first presented. It provides a description of the premise of the task and its key components. Then, this definition is translated into formal terms for the specific communication model adopted in this work.

What is transmitter identification? The task is defined as follows:

> Transmitter identification is a bimodal machine learning task in which a learning algorithm is presented with visual and wireless data obtained from a wireless communication environment and is expected to identify the object in the image responsible for the wireless data. This identification includes: (i) determining if the object is present in the image or not, and (ii) if it is present, determining which one it is.

The above definition states that the task, in essence, is a visual detection task, in which the presence of the object of interest cannot be determined using visual data alone. It requires knowledge of a form of wireless data like wireless channel information, beamforming vectors, received power,...etc. This requirement specifies the first component of the task that is the bimodality of its data. The second component of the task is specified by the entity that utilizes that bimodal data, which is a machine learning algorithm. The algorithm should be able to learn from the data how to answer the following two questions: is the transmitter present in the visual data? If yes, which object is it? The answers of these two questions serve the wireless system adopting the ViWiComm framework, and this is the third component of the task. It is critical at the point to emphasize that, unlike the work in [119], **the identification task requires the detection of the object responsible for the radio signal and not the equipment transmitting that signal**. The main reason behind this requirement is the fact that transmitting equipment is usually invisible or hard to detect, yet their carriers are commonly visible objects. Examples could be seen in people holding their phones or placing them in their pockets during a call.

MmWave Transmitter identification: In this paper, the problem of transmitter identification is addressed in mmWave communication settings. The input bimodal data for this problem is composed of an RGB frame and a pair of best beamforming-vector index and the received power at the basestation. It is important to note here that a better choice for wireless data is the wireless channel vector (i.e., $\mathbf{h}$) as it encodes all the information about the propagation paths between the transmitter and receiver. Nevertheless, in mmWave communications, such information is rarely available; obtaining it commonly entails a process riddled with communication overhead.

The problem could be formally described as follows. Let $\mathbf{X} \in \mathbb{R}^{W \times H \times C}$ be an RGB image with width W, height H, and color channels C, and let $q^\star$ represents the index of $\mathbf{f}_{\text{opt}}$ in the codebook $\mathcal{F}$. Given $\mathbf{X}$, $q^\star$, and the received power $|r|^2$, the identification task boils down to detecting a bounding box vector $\mathbf{b}_{\text{Tx}} \in \mathbb{R}^4$ that marks the transmitter in the image $\mathbf{X}$. From a machine learning perspective, learning to predict that box could be posed as a function learning problem. More to the point, a function $f_\Theta(\mathbf{X}, q^\star, |r|^2)$ parameterized by a set of parameters Θ needs to be learned from a labeled dataset $\mathcal{D} = \{(\mathbf{X}, q^\star, |r|^2, \mathbf{b}_{\text{Tx}})_u\}_{u=1}^{U}$ such that it predicts bounding boxes $\{\hat{\mathbf{b}}_{\text{Tx},u}\}_{u=1}^{U}$ with high fidelity to the groundtruth bounding boxes in $\mathcal{D}$. Formally, assuming the samples in $\mathcal{D}$ are i.i.d., the objective of the learning process is expressed as

$$\max_{f_\Theta} \prod_{u=1}^{U} \mathbb{P}(g(\hat{\mathbf{b}}_{\text{Tx},u}, \mathbf{b}_{\text{Tx},u}) \geq \gamma | \mathbf{X}_u, q_u^\star, |r_u|^2), \tag{6.4}$$

where $g(\hat{\mathbf{b}}_{\text{Tx}}, \mathbf{b}_{\text{Tx}})$ is a function assessing the level of fidelity, and γ is a fidelity threshold. A popular choice for $g(\hat{\mathbf{b}}_{\text{Tx}}, \mathbf{b}_{\text{Tx}})$ is the Intersection over Union (IoU) measure [**?**].

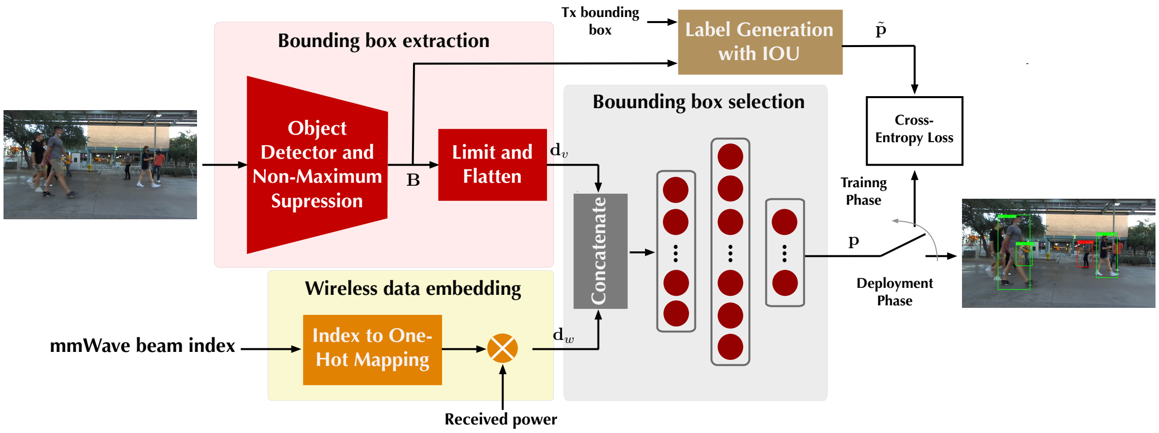


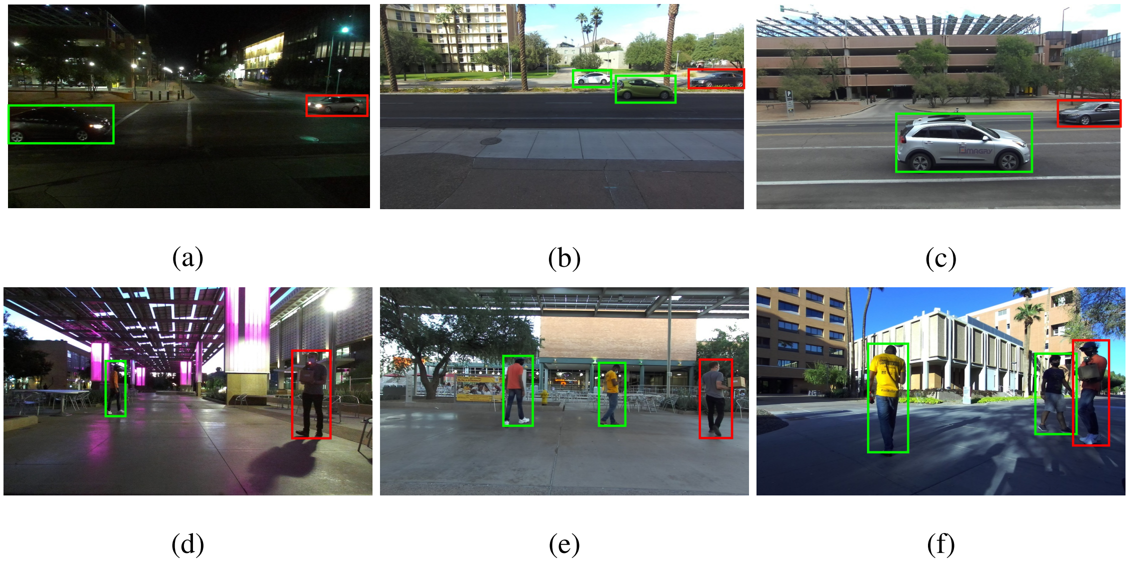
Figure 6.2: A Schematic Depicting The Proposed DNN Architecture. It Highlights The Different Components of The Architecture and Shows How It Is Implemented During Both Training and Deployment.

6.4 Proposed Solution

In designing a multimodal machine learning algorithm, a critical first step is the exploration of the observed modalities and their relation to each other. It is the knowledge of what each of them provides and lacks that could effectively guide the design process. For that end, the discussion on the proposed solution starts by exploring the relation between the input modalities and developing some intuition about the solution. Then, it proceeds to detail the proposed solution.

6.4.1 The Key Idea

Communications in the mmWave frequency range has two main characteristics: (i) the dependency on LOS links. This is due to the high signal penetration loss that makes it hard for mmWave signals to go through many materials. (ii) the use of large antenna arrays with directive radiation patterns to overcome the severe path-loss of the these high-frequency signals. Directivity in antenna arrays could be intuitively seen as a way to focus the atten-


(a) (b) (c)

(d) (e) (f)

Figure 6.3: The Six Wireless Environments Where The Vision-aided mmWave Testbed Was Deployed. (a), (b), and (c) Show Locations Where The Candidate Transmitters Were Vehicles While The Rest Show Locations Where The Candidates Were People.

tion of the array on a certain direction in space. For ULAs, directivity is achieved using the beamforming vectors in the codebook $\mathcal{F}$. In idea cases, the use of such codebook could results in a non-overlapping *sectoring* of the azimuth angle, in which every beamforming vector focuses the attention of the ULA on a unique direction in space. See the bottom image of Fig. 6.1 for an illustration.

The sectoring a beamforming codebook in mmWave communication induces could be translated into a visual effect. Recall that an RGB image is merely a projection of the 3D space onto the image 2D plan. Therefore, the sectoring defied by the beamforming vectors in $\mathcal{F}$ could also be projected onto the 2D plan of the image. The top image in Fig.6.1 illustrates that. Such effect means that, ideally, knowledge of the optimal beamforming vector $\mathbf{f}_{\text{opt}}$ could be interpreted as a form of attention in the image; it places emphasis on the direction in the image from which the current received signal arrived.

It is very important to point out here that such interpretation, i.e., the existence of non-overlapping sectors, is only valid under some ideal conditions. The first one is the need for a clear and dominate LOS connection between the transmitter and receiver. The dynamics and multi-path propagation in the mmWave environment make that condition hard to maintain [40, 42]. Another important condition is related to the beamforming codebook design. In reality, producing sharp and very directive patterns with no side-lobes, like those in Fig.6.1 is quite challenging considering the hardware limitations and impairments in the array architecture, see [125] for more information. Strict conditions such as those two could render ideal sectoring impossible; however, the beamforming vectors still induce a form of rough sectors, which could be learned by a machine learning algorithm.

The proposed solution in this paper attempts to capitalize on that notion of non-ideal sectoring and its visual effect. Given the recent advances in object detection [126, 127], objects that resemble candidate transmitters could be discovered in the image. Then, the direction information encoded into the beamforming vector could be used to identify the transmitter object from the distractors. The final result is expected to look like that in the top image of Fig.6.1.

6.4.2 Two-Stage Neural Network

Using the developed intuition in Section 6.4.1, a two-stage DNN architecture is proposed to learn the transmitter identification task. The architecture is composed of two sequential stages, see Fig.6.2, the details of which are given below.

Bounding box detection: the role of this stage is to identify the candidate objects, i.e., objects that could be transmitters. It does so by tapping into the success of Convolutional Neural Networks (CNNs) in performing object detection tasks [126, 127]. This stage adopt a pre-trained object detector, and adjusts and finetunes its classifier layer to fit the number of candidate transmitter classes in the dataset. Since object detectors commonly produce predictions with different confidence, the output of the detector is filtered using a Non-

Maximum Suppression (NMS) algorithm to keep high confidence bounding boxes. These boxes are organized in a matrix $\mathbf{B} \in \mathbb{R}^{N\times 4}$, where the number of rows N represents the maximum number of boxes an image is expected to have. For the cases where the total number of extracted boxes from an image is less that N, the matrix $\mathbf{B}$ is padded with zero vectors. Following NMS, $\mathbf{B}$ is flattened into a high dimensional vector $\mathbf{d}_v \in \mathbb{R}^{4N\times 1}$ representing the visual feature vector.

Bounding box selection: this stage is where both visual features and wireless features are merged and processed to extract the final transmitter bounding box. At the beginning of this stage, the mmWave beam index $q^\star$ is embedded into a one-hot vector that is scaled by the received power $|r|^2$. This produces the wireless feature vector $\mathbf{d}_w \in \mathbb{R}^{Q\times 1}$, which is stacked with the visual feature $\mathbf{d}_v$, as shown in Fig.6.2. The selection process in this stage is posed as a classification problem, in which the output produces a probability distribution $\mathbf{p} \in \mathbb{R}^{(N+1)\times 1}$ over $N+1$ classes, representing the extracted boxes and the case of no transmitter. The selection network in this stage is designed as a Multi-Layer Perceptron (MLP) network with four stacks. The first three are composed of sequences of fully-connected, batch normalization, and ReLU layers [73] while the last classifier stack is made of fully-connected and softmax layers. The breadths of these four stacks are, respectively, 256, 1024, 1024, and 11.

6.5 Experimental Setup

In order to develop or benchmark solutions to the transmitter identification task, there is a need for a bimodal dataset that have vision-wireless data collected from real wireless environments. Such dataset, to the best of our knowledge, is only available in the form of synthetic data-generation frameworks, e.g., ViWi [117]. Hence, this section presents the details of how a transmitter identification development dataset has been constructed from real wireless communication environments.

6.5.1 Testbed Description and Development Dataset

A ViWiComm system (or testbed) is built for the sake of data collection. The testbed comprises two stand-alone unites, a basestation and a mobile transmitter. Both unites operate in the 60 GHz frequency band. The basestation consists of two main terminals and a controller. The first terminal is a 16-element ($M = 16$) mmWave phased array adopting a beamforming codebook with 16 beams ($Q = 16$) while the second is an RGB camera. The two terminals are installed on top of one another, so their fields of view are aligned. They are also connected to a laptop to operate them and read out the collected data. On the other side, the mobile unit has a single-element antenna and is connected to its own local controller and power supply. The mobile unit is only initialized by the laptop of the basestation at the beginning of it operation, and throughout the data collection session it operates in a stand-alone fashion.

The development dataset of transmitter identification is constructed by deploying the testbed at 6 different locations, see Fig.6.3. These location depict various outdoor wireless environments. In three of these locations, Fig.6.3a, 6.3b, and 6.3c, the mobile unit is carried by a moving vehicle. These three locations broadly represent environments dominated by vehicle transmitters (transmitters inside a vehicles). The three locations are visited at different times of the day to obtain visual data with variable lighting conditions. In the other three locations, Fig.6.3d, 6.3e, and 6.3f, the mobile unit is carried by a walking person. These three locations broadly represent environments dominated with human transmitters. Similar to the first three locations, the testbed is also deploy at different times of the day to get diverse images.

Using the data collected from the six locations (henceforth referred to as the raw data), the development dataset for transmitter identification is constructed. It has a little over 3000 data samples which are divided into training and validation sets with a split of $70 - 30\%$.

Table 6.1: Object Instances

Class type	Object instances	Tx instances
Car	1840	1528
Person	5052	1359
Cyclist	17	0
No Objects	155	None

To get that dataset, the raw data undergoes a processing pipeline. The first step in the pipeline is to extract samples with meaningful information; data samples where the mobile unite is out of range with the basestation are filtered out since they neither have visual nor wireless information. The next step in the pipeline is annotation. All visual samples are manually annotated to have groundtruth bounding boxes for all instances of candidate objects and for the transmitter object. The list of classes used in the annotation process is: transmitter, vehicle, person, and cyclist. Table 6.1 lists the number of object-class and transmitter instances (Tx instances), i.e., how many times an object appeared in the dataset and in how many of those instances the object is the transmitter. It is important to note here that since this dataset is constructed from real measurements, there are data samples where candidate objects appear in the RGB image while the actual transmitter does not, see the last row of Table.6.1. This is due to the slightly wider field of view of the phased array.

6.5.2 Network Training

The proposed DNN is trained in two stages on a Linux system with an NVIDIA^{TM}Quadro RTX 6000 GPU. First, the object detector is finetuned on the training dataset to detect candidate transmitters. Then, the selection network is trained on the same dataset using the outputs of the detector and the wireless data.

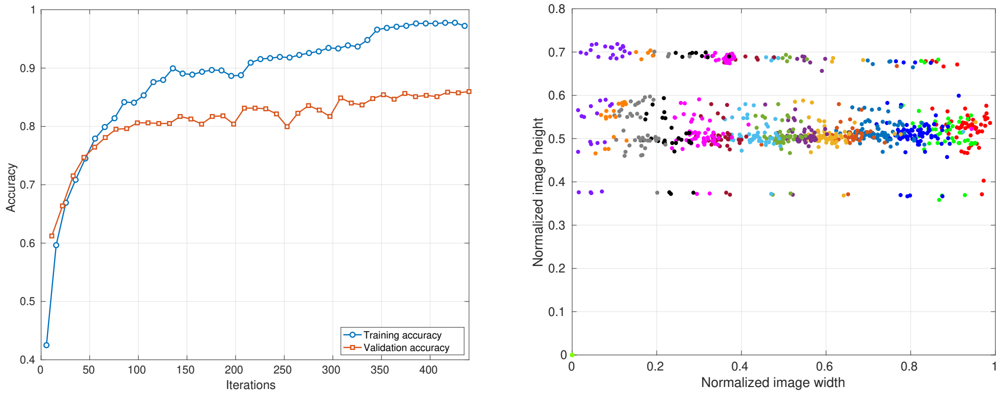
Table 6.2: Training Hyper-parameters

Solver	Adam [84]
Learning rate	1e-2
Learning rate schedule	0.1 @ epoch 30
Number of epochs	40
Dropout	50%
Batch size	200
IoU threshold (γ)	0.5
Maximum number of boxes (N)	10

The details of the training process are as follows. A Yolo object detector [126] trained on the COCO dataset [128] is adopted in the proposed architecture. The detector is fine-tuned, so its classifier layer detects the candidate objects: vehicle, person, and cyclist. It is trained with a stochastic gradient descent with momentum (SGDM) solver, a learning rate of 1×10^{-5}, a weight decay of 5×10^{-4}, a momentum of 0.9, and 50 training epochs. The selection network, on the other hand, is trained on the bounding boxes extracted by the trained detector and the wireless data in the training set. A cross entropy loss [73] is used in the training, and the training hyper-parameters are shown in Table.6.2. The training progress is illustrated in Fig. 6.4a More on the implementation could be found in [83].

6.6 Experimental Results

A sequence of experiments are conducted to study the performance of the proposed architecture and gain some insights into the task itself. The sequence starts with putting the notion of beam-induced sectoring to test, and form its findings, the rest of the experiments will explain the choices behind the proposed DNN, highlight its advantages, and discuss its



(a) The Training Progress of The Selection Network. It Is Shown in The Form of Prediction Accuracy, Training and Validation.

(b) The Distribution of Transmitter Bounding Boxes Across The 2D Image Plan, Using The Center of Each Box.

shortcomings.

6.6.1 Image Sectoring

The first experiment aims to study the sectoring effect induced by the mmWave beams. Fig.6.4b depicts a scatter plot for 1000 randomly picked transmitter bounding boxes from the training set plotted on the 2D plane of the image. These boxes are represented by the coordinates of their centers, and they are grouped according to the beamforming vector they are associated with (i.e., clustered according to their groundtruth beam indices). The immediate observation from the figure is that, on the image plane, the clusters do not define the ideal sectoring effect discussed in Section 6.4.1. They actually define overlapping sectors, which suggest that identifying transmitter boxes by finding their sectors may not be as simple as one might think.

Recognizing that bounding boxes are 4D vectors and what Fig.6.4b shows is merely the first two dimensions, the figure may not paint a complete picture on beam-induced sectoring. Hence, a more grounded approach to study how ideal the sectoring is could be to train a linear classifier to cluster the transmitter boxes based on their groundtruth beam

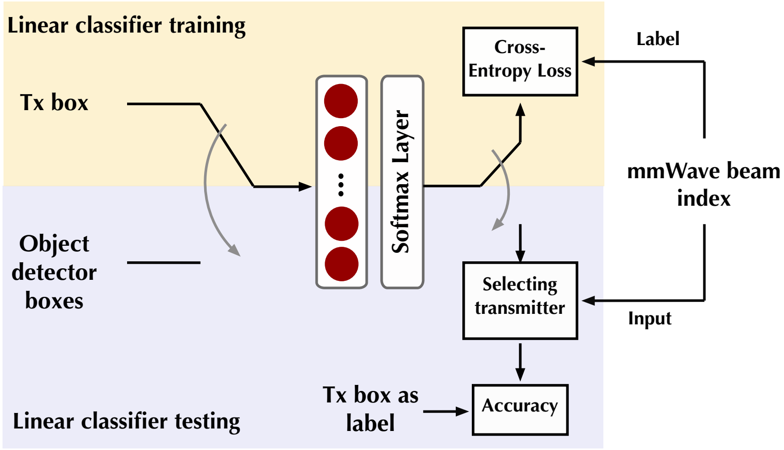
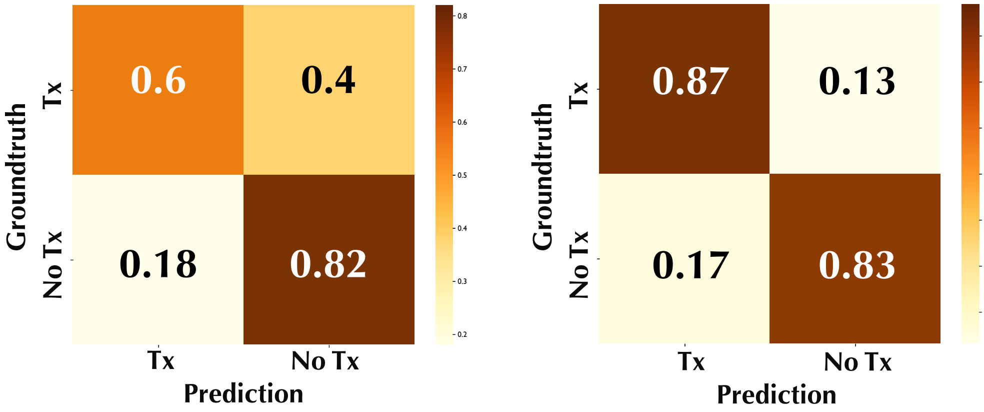


Figure 6.5: A Schematic for The Linear Classifier Test.

indices. To that end, a simple classifier is built using 16-neuron fully-connect and softmax layers. It is trained to see a transmitter bounding box from the training dataset and predict its beam index, as shown in Fig.6.5. Once it is trained, its ability to identify the transmitter bounding box is put to test. This is done as follows: (i) feed the detected bounding boxes by the object detector to the classifier to predict the beam index of each box (i.e., sectors), (ii) match the observed beam index to those indices predicted by the classifier to pick the transmitter bounding box, and, finally, (iii) measure IoU between the predicted bounding box and the target box to assess the prediction accuracy. Following this approach, we get an identification accuracy of $\sim 17\%$ on the validation set of transmitter identification. Such low accuracy empirically suggests that ideal sectoring does not hold in the dataset, ergo real wireless environments. This motivates the design of more sophisticated learning algorithms.

6.6.2 Proposed DNN Performance

A different approach to transmitter identification is to train a classifier to pick from the bounding boxes extracted by the object detector. This approach relies on the contextual



(a)

(b)

Figure 6.6: The Confusion Matrices of The Bounding Box Selection Network. (a) Linear Classifier, and (b) Multi-layer Classifier.

information the detected boxes provide during training, as opposed to only seeing the transmitter boxes. To that end, we train two bounding-box selection networks, a simple linear classifier, similar to that in the previous section, and a multi-layer classifier, like that proposed in Section 6.4.2. Both are fed the stacked visual and wireless feature vectors. Fig.6.6 depicts the confusion matrices of the two classifiers on the validation set. Given that the number of positive cases (transmitter exists) in the validation set is 769 while the number of negative cases is 144, the accuracies of both networks are 63% for the linear classifier and 86% for the multi-layer classifier. This shows a clear advantage to the multi-layer classifier, which is the reason behind choosing that architecture in the proposed solution.

By taking a closer look at Fig.6.6, one could see that the multi-layer classifier outperforms the linear one in detecting positive cases by a landslide. This is clear in the precision-recall performance; the linear classifier achieves $\sim 94\%$ precision at only 60% recall while for the other classifier, precision hits $\sim 96\%$ at much higher recall, 87%. These numbers re-affirm the notion that beam-induced sectors are not ideal and, hence, they are not easily discerned. Of course, the linear classifier performs better when it sees all the boxes along

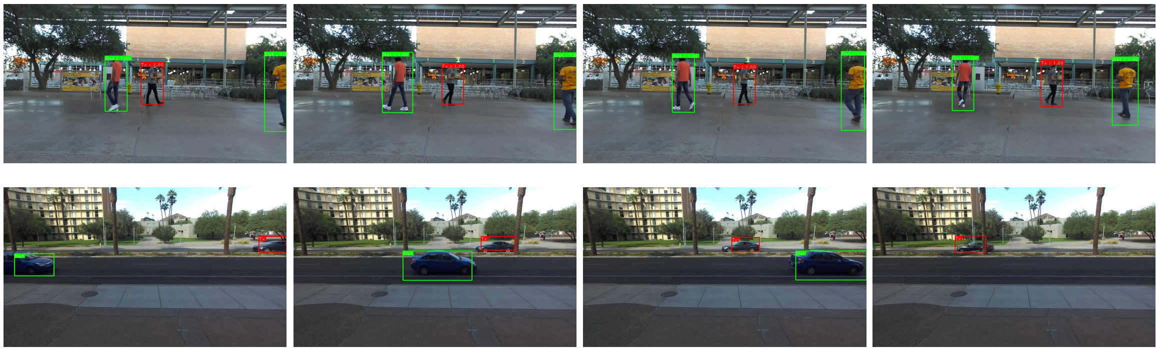


Figure 6.7: Two Example Sequences of Transmitter Identification in Action. Upper Row Shows The Detected Human Transmitter in an Environment with Multiple Human Candidates While The Bottom Row Presents Another Example for Transmitter Detection in an Environment with Multiple Vehicle Candidates.

with the wireless feature than in the case discussed in Section 6.6.1. However, it could be conjectured that the relation between the visual and wireless features is better captured by a multilayer classifier.

The last experiment in this section studies the role of received power information $|r|^2$. To do that, we train the proposed solution with the multi-layer classifier but without the received power, i.e., the beam embedding is not scale by the power. The result is a slight dip in the total accuracy of the DNN by around 1% compared to the accurcay with received power. Such result may, on the surface, indicate that received power is not of great importance to the architecture, which is not quite correct. Power, in general, reflects a sense of distance between the transmitter and receiver, but this sense is commonly characterized with a range of error. In other words, when two transmitters are in close proximity to one another, received power may not be a clear indicator of the distance between the receiver and each one of them. This is mainly due to the fading effect in wireless channels, see [107]. Before wrapping up this section, it would be interesting to put all the above analysis in some visually pleasing context, and this is what Fig. 6.7 simply does. It depicts two short frame sequences from two different locations with the bounding box predictions and their labels. In both sequences, the architecture is able to track the transmitter successfully in spite of the different candidate objects.

Chapter 7

CONCLUSION

Both parts of this dissertation have shown that ML (and more specifically deep learning) is instrumental to modern and future large-scale MIMO wireless networks. A simple reason behind that could be the fact that ML enables those networks to utilize their own experiences to improve their performance. Frameworks 1, 2, and 3 are good examples of that, for all utilize the experiences that come in the form of estimated channels of various users and visual data of the wireless environment to deal with some of the most pervasive challenges to large-scale MIMO communications. The following three subsections present summaries and concluding remarks for the topics covered in this dissertation.

7.1 Framework 1: Deterministic channel prediction

The deterministic channel-prediction framework provides a simple yet interesting argument for how full channel/beam-training might not be needed in large-scale MIMO networks. It argues that within the same wireless environment, a mapping function relating some wireless channels at some frequency band and other wireless channels at another band exists, and, therefore, it proposes to use ML and more specifically DNNs to learn that function and eliminate the need for expensive channel/beam-training. The framework is studied in two different large-scale MIMO settings. The results and findings of those studies are discussed in the following three subsections. The first two summarize the main experimental results and their direct implications on each setting while the third subsection presents the main takeaways pertaining to Framework 1.

TDD and FDD massive MIMO

Two major challenges in massive MIMO systems have been addressed with the deterministic channel-prediction framework. The fist is the challenge of eliminating the need for downlink-channel training in FDD massive MIMO. It is addressed by posing it as an uplink-to-downlink regression task, in which the uplink channels (which are easy to obtain) are the observed variables and the downlink channels are the targets. A DNN is designed to learn the relation between the observed and target variables and preform downlink channel prediction. The performance of the proposed DNN for that task is quite interesting; in a distributed massive MIMO deployment and at a high SNR regime ($\geq$ 5 dB), the network achieves ≈ 0.01 NMSE. Similar results are obtained in co-located massive MIMO deployment. The DNN is able to achieve ≈ 0.01 NMSE at a high SNR regime ($>$ 5dB). The other challenge considered is the elimination of full franthaul-channel feedback, which is important for TDD distributed massive MIMO. The developed DNN is able to reconstruct full channels using a small sample of them selected at random. More specifically, at a high SNR regime ($\geq$ 5 dB), the NMSE of reconstructed channels is ≈ 0.01. A final important takeaway from this task is that the bijectiveness assumption seems to be satisfied with relatively small number of antennas. The experiments show that approximately 16 channel samples are enough to learn the prediction function.

The experiments conducted in Chapter 2 do not only point to the potential of the framework, but also its shortcomings. Right off the bat, the experiments show that good performance is only achieved when the DNN is presented with large enough dataset. This could be seen as an obvious and expected observation, yet when combined by the numbers and the fact that the environments are all stationary, it points to a clear drawback. In particular, for both tasks mentioned about, a training dataset with more than $\approx 100 \times 10^3$ samples is needed to get a reasonable performance. The other shortcoming is rooted in the NMSE performance the DNN achieves. Again on both tasks, it is clear that in the best possible conditions, the NMSE rarely falls below 0.01. This might not be a problem when user interference is not factored in the performance evaluation. All experiments in the Chapter focuses on the performance in single-user settings, and the reported beamforming gains corroborate the conclusion that the DNN performs well in those settings. However, when multi-user settings are considered, the network performance is expected to drop significantly. This is demonstrated in the reported results in Chapter 4.

Dual-Band Sub-6 GHz and mmWave MIMO

Chapter 3 establishes the conditions under which the mapping functions from a sub-6GHz channel to the optimal mmWave beam and blockage status exist. Leveraging the universal approximation theory, a large enough neural network is proven to be able to learn those mapping functions such that the success probabilities of predicting the optimal mmWave beam and blockage status be arbitrarily close to one. Therefore, a neural network is designed such that it performs both prediction tasks using sub-6GHz channels. With the help of accurate 3D ray-tracing software, development datasets are construct to evaluate and test the designed network. The results show promising and impressive performance; the network, when trained with enough data, does both tasks with relatively high fidelity, even in the presence of noisy sub-6GHz channels. Beam-prediction experiments reveal an interesting tendency of the network to learn correct beam direction. Although it sometime mis-predicts the mmWave beam, it often selects a beam in the vicinity of the optimal one. This is attainable with small or large mmWave antenna arrays and at reasonable SNRs. Such performance extends to the blockage prediction task; the network, under high SNRs, is capable of predicting the LOS link status with more than 90% success probability. This could yield interesting gains for the reliability of mmWave systems. For future work, it would interesting to develop learning models that can handle the dynamics of the environment, investigate the practical conditions under which the bijectiveness conditions are violated, and design more efficient and practical labeling approaches to label blockage data.

Main Takeaways

What could be learned from the various discussions and experimental results in Chapters 2 and 3 is summarized in the following points:

1. Framework 1 emphasizes the value of ML, and especially deep learning, to large-scale MIMO systems; it presents a good example on how ML can help a large-scale MIMO system utilize its experiences, e.g., previously estimated uplink and downlink channels, and overcome its channel-related challenges (see Chapter 1.2.1).

2. Although the discussion on deterministic channel prediction in Chapters 2.4 considers a "static communication environment," the developed theoretical argument does not explicitly depend on that assumption; the main condition for a channel-to-channel prediction function to exist is the bijectiveness of one position-to-channel function (Assumptions 1 and 2). If such condition is maintained in dynamic communication environments (i.e., realistic wireless communication environments), the channel-to-channel prediction function is expected to exist, and, hence, the task of predicting channels from others degenerates to a problem of designing the right DNN to learn that prediction function.

3. The experimental results in Chapter 2.7 points to an interesting conclusion regarding the design of a DNN. One deep-enough fully-connected neural network architecture could be designed to learn the channel-to-channel function in different communication environments with different large-scale MIMO system deployments, e.g., a cell-free massive MIMO or co-located massive MIMO. The results suggest that the architecture needs to be trained on a dataset from each environment.

4. Transfer learning could be an effective approach to deal with the need to train one DNN architecture to perform well in different communication environments. Although it has not been investigated in Chapter 2.7, its benefits to adapting a DNN to two different tasks in a communication environment have been studied in Chapter 3.8.6. The results encourage further investigation of transfer learning across different environments.

5. Learning the channel-to-channel prediction function is very likely to be challenging in dynamic communication environments (realistic communication settings). This is suggested by the training results of Chapter 2.8; the proposed DNN can learn the prediction function but with low fidelity. In fact, the fidelity issue gets exacerbated with increased training, revealing an overfitting trend. This could be attributed to the complexity of the time-varying prediction function, for the results of the bijectiveness study in Appendix A indicate that for a dynamic communication environment similar to that used in Chapter 2.8, channels of different users are likely to be bijective.

7.2 Framework 2: Statistical channel prediction:

In an effort to overcome the challenges associated with Framework 1 and highlighted in Chapters 2.4.3 and 4.4, the statistical channel-prediction framework has been proposed in Chapter 4. It aims to predict s summary statistic about the target downlink channels using deep learning and the easy-to-acquire uplink channels. The summary statistic is quantifies in thee form of a conditional downlink-channel covariance, conditioned on some observed uplink channels. This covariance summarizes the large-scale fading behavior of the wireless channel, which makes it sufficient to reduce the downlink channel- and beam-training overheads. Such covariance represents an interesting and robust alternative to predicting the downlink channels directly.

The covariance prediction task is addressed from two different perspectives in Chapter 4, supervised and unsupervised. The following two subsections provide a more pointy discussion on the two.

Regression Approach

The supervised learning perspective presents a regression approach to tackle the covariance prediction task. A theoretical argument for this approach is presented to formally establish its premise. It states the conditions and the type of training loss under which regression can asymptotically produce the sought-after conditional covariance. A solution based on neural networks is, then, proposed to validate the argument empirically and study its conditions. Using a communication scenario from the publicly available DeepMIMO dataset, the regression solution is trained and tested under various settings. The experiments show promising results that agree with the theoretical argument and emphasize the value of the framework; for an uplink-to-downlink prediction task at the same basestation, the results show a reduction of more than 85% of the channel-training overhead for a basestation with 64-antenna ULA. This is achieved while attaining more than 78% of the optimal performance and with up to 4 multiplexed users. This good performance extends to noisy channel conditions, where 85% overhead reduction could maintain $\sim$90% of the perfect single-user beamforming gain at a -10 dB SNR.

Clustering approach

The unsupervised learning perspective takes a clustering approach to the covariance prediction task. The motivation behind it is rooted into the multi-ring channel model (referred to as multi-channel cone model), which sees the wireless environment broken down into multiple rings of scatters. Under that model, the clustering approach is formally shown to be capable of predicting downlink covariances given that an ideal clustering function could be learned. Owing to the fact that the channel cones and their number are difficult to characterize in a wireless environment, such function is difficult to learn in practice. An alternative is to learn a sub-optimal (surrogate) clustering function, which could be empirically shown to produce good performance. This is exactly what the experimental results reported. Using the same DeepMIMO scenario used for the regression solution, the clustering solution displays competitive performance. It achieves similar beamforming gain and channel-estimation quality to those achieved by the regression solution. For instance, it achieves more than 85% training-overhead reduction while attaining more than 78% of the optimal performance and with up to 4 user. Lack of supervision in this approach, however, reflects on the performance. This could be seen in the case where the system operates under a high-SNR regime; clustering is 8% behind the regression performance.

Overall, both approaches feature intriguing properties that could be the key to enabling massive MIMO in real wireless environments. Using machine learning to enhance, not replace, the classical system operation is an essential advantage to the statistical prediction framework. It brings the best of the two worlds, signal processing and machine learning. This leads to more robust operation and backward-compatible evolution. Another advantage arises with the framework is the extra degree of freedom that manifests in the role of downlink pilots. Varying the number of pilots provides some form of adaptability in the system, which is helpful when dealing with the dynamics of the wireless environment.

Main Takeaways

The main outcomes of Chapter 4, especially when contrasted to the results in Chapters 2 and 3, are as follows:

1. Statistical channel prediction is an effective approach to alleviate the learning challenge associated with the time-vary nature of the channel-to-channel prediction function introduced in Chapter 2 and emphasized in the “Main Takeaway” list of Chapter

7.1. It overcomes that challenge by learning to predict a covariance of the target channels conditioned on some observed channels. The advantage of such covariance is that it is, to a large extent, time-invariant, and learning its prediction function could be less challenging in practical communication environments.

2. Framework 2 trades off the learning complexity of a time-varying function for a relatively small increase in channel-training overhead. This trade-off is illustrated in the experimental results of Chapter 4.9; Framework 1 struggles to maintain satisfactory channel estimation NMSE whereas Framework 2 could achieve an order of magnitude smaller NMSE than that of Framework 1 using a fraction of the total downlink training pilots needed for per-antenna training. The edge of Framework 2 over Framework 1 is further extended when user multiplexing is considered. The average per-user spectral efficiency drops at somewhat exponential rate with respect to number of users for Framework 1, but it maintains a close to optimal performance with Framework 2 operating with light-weight downlink channel training.

3. From a ML perspective, learning the covariance prediction function with regression is expected to be more challenging compared to clustering. This is a consequence of two observations:

   (a) Learning to uncover the channel clusters (channel cones) requires only the uplink channels. Those channels are significantly easier to acquire compared to downlink channels, making the development dataset of clustering easier to construct as opposed to that of regression.

   (b) Clustering produces a downlink channel covariance per cluster not per user position. This is advantageous for multiple users falling within the same channel cone; they share the same scatterers, and, as such, their channel large-scale statistics is expected to be the same, i.e., they have the same covariance.

Note here that the clustering approach still requires the estimation of downlink channels to estimate the per-cluster covariances; *however, different to the regression approach, those downlink channels need not be estimated for every uplink channel*. The wireless system have some room to pick when the downlink channels could be estimated. For instance, a highly mobile user cannot contributed to a regression dataset as the overhead of its downlink channel is prohibitive, yet it could easily contributed to a clustering dataset.

4. Results in Chapter 4.9 may give the impression that regression is slightly better than clustering, which could be deceiving. The apparent subtle edge regression gains over clustering could be explained by the fact that regression follows a supervised learning paradigm, which is a well studied and understood paradigm in the ML community. On the other hand, clustering follows a self-supervised or unsupervised learning paradigms, which are far behind on the evolutionary scale of understanding in the ML community compared to supervised learning—they have been gain more traction recently, though.

5. Framework 2 is prone to *catastrophic channel-subspace expansion* which is not an issue with Framework 1. When the channel cone widens (i.e, scatterer are widely spread around a user), the downlink channel covariance is expected to have a high rank (i.e., indicating large angle spread), and this defines an event referred to in this dissertation as catastrophic channel-subspace expansion since the covariance is unable to reduce the channel-training overhead significantly. Statistical channel prediction aims to predict the downlink channel covariance, but it does not have a say in its rank. Thus, in such event, the prediction of the downlink channel covariance may not result in a meaningful reduction of the training overhead. This is not the case for Framework 1 as it learns to predict the wireless channels directly and with zero

training overhead.

7.3 Framework 3: Vision-aided wireless communications

ViWiComm presents a novel and unorthodox framework to deal with high-frequency large-scale MIMO challenges. It recognizes and utilizes an interesting parallel between computer vision and high-frequency communications, which is the reliance on LOS. This parallel allows ViWiComm to tap into advances in deep learning and computer vision to address challenges like beam prediction, beam tracking, and LOS-blockage prediction. The parallel introduces new ML tasks, as well, that are not commonly known in classical fields such as computer vision and NLP. The transmitter identification task in Chapter 6 is a good example of that. The following two subsections summarize the discussions and main takeaways of Chapters 5 and 6.

Vision-aided beam and blockage prediction

Using computer vision and deep learning to tackle beam and blockage prediction problems is one promising approach to realize the potential of mmWave systems. The proposed solutions has clearly shown that promise for the case of single-user communications. Utilizing the strong correspondence between image classification and the tasks of beam prediction and user detection, a state-of-the-art deep learning model, like ResNet-18, trained for image classification could be fine-tuned to perform both tasks effectively. Both solutions need to be further developed and studied for dynamic environments with multiple users; they pose more difficult and realistic challenges to mmWave systems compared to those posed in the single-user scenarios considered in Chapter 5. Overall, if the results of that chapter are any indicator, vision-based approaches are definitely a strong contender for tackling problems related to link-blockage and beam selection.

Transmitter identification

Chapter 6 takes an important step towards addressing a critical concern about the practicality of the ViWiComm framework in real multiuser communication settings. It does so by: (i) defining the novel transmitter identification task, (ii) proposing a deep learning solution to the task, and (iii) building a development and benchmarking dataset of data samples obtained from real wireless environments. The development of the proposed DNN shows that images and mmWave beams could complement each other in terms of the information they convey about a wireless environment. The DNN utilizes those bimodal data to identify the wireless transmitter, and in doing so, it achieves an error margin of 14%. Such result is a clear indicator of the complemental nature of vision-wireless data, and it provides a vote of confidence in ViWiComm as a novel framework to address real high-frequency large-scale MIMO challenges.

The encouraging results reported in Chapter 6, however, should not deflect attention from how difficult transmitter identification is. Considering the relatively small size of the validation set compared to modern-day deep learning datasets, the 14% error margin is not expected to hold up as more data samples are added—especially if those samples are obtained using the same vision-wireless hardware but are coming for different wireless environments; the small dataset may not include cases that test beam-resolution limitation of the proposed identification approach. Beamforming codebooks, in general, have finite number of beams, indicating that their beams have finite spatial resolution. The limited resolution is a serious problem when two candidate transmitters fall within the same beam sector. The proposed DNN, or any ML algorithm for that matter, may not be able to differentiate those candidates using images and mmWave beams alone. Such limitation motivates further research on how to improve the identification performance, in particular, on what wireless data could be a better companion to the RGB images than mmWave beams. An answer could be found in sub-6 GHz channels, for they have two important features: (i) they are common companions to mmWave channels or beams (see Chapters 3 and 5); and (ii) sub-6 GHz channels are easier to obtain compared to mmWave channels or beams.

Main Takeaways

The main outcomes of Framework 3 can be summarized in the following points:

1. Framework 3 presents a new paradigm to wireless communications that could overcome the challenges with LOS-related challenges in high-frequency large-scale MIMO. It recognizes the fact that high-frequency communications are reliant on LOS much as computer vision. Therefore, it proposes to leverage the advances in deep learning and computer vision to equip high-frequency communication systems with *a sense of their surrounding*.

2. The framework presents a new challenge from a ML perspective; in order to provide a sense of surrounding to a wireless system, a ML algorithm needs to understand the content of visual data from the perspective of a wireless system, i.e., wireless-based scene understanding. This means the algorithm must be able to identify important objects in the environment; understand their role in the environment (e.g., transmitters, receivers, LOS blockages, signal scatterers, ...etc); and factor in the object type and role in the decision making process.

3. The transmitter identification task, introduced in Chapter 6, is a good example of a new ML challenge presented by Framework 3, and it is an approach to achieve wireless-base scene understanding. The task cannot be performed using visual or wireless data alone, and, as such, it mandates novel bimodal deep learning algorithms that can learn cross-modality patterns.

4. Transmitter identification in high-frequency large-scale MIMO system is not an easy task given the spatial information beamforming provides. This is corroborated by the experimental results of Chapter 6.6; they show that simple linear classifiers do not provide satisfactory performance on identifying transmitters out of a set of detected objects. This indicates that deep learning based algorithm are needed to effectively learn the identification task.

5. Robust transmitter identification likely requires the pairing of MIMO wireless channels, not beamforming vectors, with visual data. It is true that beamforming vectors encode explicit spatial information, yet they are limited by the size of the codebook and the resolution of the realized beams. On the other hand, wireless channels, especially sub-6 GHz channels, are more descriptive than beamforming vectors, i.e., have more information about signal propagation. Therefore, robust transmitter identification is expected to require deep learning algorithms that learn jointly from visual and sub-6 GHz MIMO channels.

REFERENCES

[1] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced BERT with disentangled attention,” *arXiv preprint arXiv:2006.03654*, 2020.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[3] J. Janai, F. Güney, A. Behl, A. Geiger *et al.*, “Computer vision for autonomous vehicles: Problems, datasets and state of the art,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1–308, 2020.

[4] J. Gao, M. Galley, and L. Li, “Neural Approaches to Conversational AI,” ser. SIGIR ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1371–1374. [Online]. Available: https://doi.org/10.1145/3209978.3210183

[5] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[10] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[11] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in *International conference on machine learning*, 2014, pp. 1764–1772.

[12] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[14] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[15] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834–1853, 2018.

[16] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5G be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[17] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Deep learning for massive MIMO with 1-bit ADCs: When more antennas need fewer pilots," *IEEE Wireless Communications Letters*, vol. 9, no. 8, pp. 1273–1277, 2020.

[18] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep Learning Coordinated Beamforming for Highly-Mobile Millimeter Wave Systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[19] G. Charan, M. Alrabeiah, and A. Alkhateeb, "Vision-Aided 6G Wireless Communications: Blockage Prediction and Proactive Handoff," *arXiv preprint arXiv:2102.09527*, 2021.

[20] C. Wen, W. Shih, and S. Jin, "Deep Learning for Massive MIMO CSI Feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, 2018.

[21] X. Li, A. Alkhateeb, and C. Tepedelenlioğlu, "Generative adversarial estimation of channel covariance in vehicular millimeter wave systems," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 1572–1576.

[22] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.

[23] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[24] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[25] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle *et al.*, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, p. 153, 2007.

[26] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[27] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[30] Y. Bengio *et al.*, “Learning deep architectures for AI,” *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[31] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[32] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[33] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, “The expressive power of neural networks: A view from the width,” in *Advances in neural information processing systems*, 2017, pp. 6231–6239.

[34] D. Elbrächter, D. Perekrestenko, P. Grohs, and H. Bölcskei, “Deep neural network approximation theory,” *arXiv preprint arXiv:1901.02220*, 2019.

[35] D. Erhan, “Understanding deep architectures and the effect of unsupervised pretraining,” Ph.D. dissertation, 2011.

[36] Y. Bengio, O. Delalleau, and N. Le Roux, “The curse of highly variable functions for local kernel machines,” *Advances in neural information processing systems*, vol. 18, p. 107, 2006.

[37] M. Elsayed and M. Erol-Kantarci, “AI-Enabled Future Wireless Networks: Challenges, Opportunities, and Open Issues,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 70–77, 2019.

[38] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, “6G wireless networks: Vision, requirements, architecture, and key technologies,” *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019.

[39] E. Bjornson, E. G. Larsson, and T. L. Marzetta, “Massive MIMO: Ten myths and one critical question,” *IEEE Communications Magazine*, vol. 54, no. 2, pp. 114–123, 2016.

[40] R. W. Heath, N. Gonzalez-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.

[41] L. Sanguinetti, E. Björnson, and J. Hoydis, "Towards Massive MIMO 2.0: Understanding spatial correlation, interference suppression, and pilot contamination," *IEEE Transactions on Communications*, 2019.

[42] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond," *IEEE Access*, vol. 7, pp. 78 729–78 757, 2019.

[43] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *Journal of Communications and Networks*, vol. 15, no. 4, pp. 338–351, 2013.

[44] J. G. Andrews, T. Bai, M. Kulkarni, A. Alkhateeb, A. Gupta, and R. W. Heath Jr, "Modeling and Analyzing Millimeter Wave Cellular Systems," *submitted to IEEE Transactions on Communications, arXiv preprint arXiv:1605.04283*, 2016.

[45] A. Alkhateeb, I. Beltagy, and S. Alex, "Machine learning for reliable mmWave systems: Blockage prediction and proactive handoff," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 1055–1059.

[46] M. B. Mashhadi and D. Gündüz, "Deep learning for massive MIMO channel state acquisition and feedback," *Journal of the Indian Institute of Science*, vol. 100, no. 2, pp. 369–382, 2020.

[47] A. Adhikary, J. Nam, J. Ahn, and G. Caire, "Joint Spatial Division and Multiplexing The Large-Scale Array Regime," *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6441–6463, 2013.

[48] Y. Han, J. Lee, and D. J. Love, "Compressed Sensing-Aided Downlink Channel Training for FDD Massive MIMO Systems," *IEEE Transactions on Communications*, vol. 65, no. 7, pp. 2852–2862, 2017.

[49] M. B. Khalilsarai, S. Haghighatshoar, X. Yi, and G. Caire, "FDD massive MIMO via UL/DL channel covariance extrapolation and active channel sparsification," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 121–135, 2018.

[50] J. Flordelis, F. Rusek, F. Tufvesson, E. G. Larsson, and O. Edfors, "Massive MIMO Performance–TDD Versus FDD: What Do Measurements Say?" *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2247–2261, 2018.

[51] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-Free Massive MIMO: Uniformly great service for everyone," in *2015 IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2015, pp. 201–205.

[52] R. W. Heath Jr and A. Lozano, *Foundations of MIMO communication*. Cambridge University Press, 2018.

[53] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed Channel Sensing: A New Approach to Estimating Sparse Multipath Channels," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1058–1076, 2010.

[54] Y. Ding and B. D. Rao, "Dictionary Learning-Based Sparse Channel Representation and Estimation for FDD Massive MIMO Systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5437–5451, 2018.

[55] X. Rao and V. K. N. Lau, "Distributed Compressive CSIT Estimation and Feedback for FDD Multi-User Massive MIMO Systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3261–3271, 2014.

[56] F. Rottenberg, R. Wang, J. Zhang, and A. F. Molisch, "Channel extrapolation in FDD massive MIMO: Theoretical analysis and numerical validation," *arXiv preprint arXiv:1902.06844*, 2019.

[57] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved Handover Through Dual Connectivity in 5G mmWave Mobile Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 2069–2084, 2017.

[58] M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Multi-connectivity in 5G mmWave cellular networks," in *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2016, pp. 1–7.

[59] D. Aziz, J. Gebert, A. Ambrosy, H. Bakker, and H. Halbauer, "Architecture Approaches for 5G Millimeter Wave Access Assisted by 5G Low-Band Using Multi-Connectivity," in *2016 IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[60] N. H. Mahmood and H. Alves, "Dynamic Multi-Connectivity Activation for Ultra-Reliable and Low-Latency Communication," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, 2019, pp. 112–116.

[61] X. Li and A. Alkhateeb, "Deep Learning for Direct Hybrid Precoding in Millimeter Wave Massive MIMO Systems," *CoRR*, vol. abs/1905.13212, 2019. [Online]. Available: http://arxiv.org/abs/1905.13212

[62] J. Choi, W. Lee, J. Lee, J. Lee, and S. Kim, "Deep Learning Based NLOS Identification With Commodity WLAN Devices," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3295–3303, 2018.

[63] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling Large Intelligent Surfaces With Compressive Sensing and Deep Learning," *IEEE Access*, vol. 9, pp. 44 304–44 321, 2021.

[64] T. Nishio, H. Okamoto, K. Nakashima, Y. Koda, K. Yamamoto, M. Morikura, Y. Asai, and R. Miyatake, "Proactive Received Power Prediction Using Machine Learning and Depth Images for mmWave Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2413–2427, 2019.

[65] M. Alrabeiah, Y. Zhang, and A. Alkhateeb, "Neural Networks Based Beam Codebooks: Learning mmWave Massive MIMO Beams that Adapt to Deployment and Hardware," *arXiv preprint arXiv:2006.14501*, 2020.

[66] Y. Wang, A. Klautau, M. Ribero, A. C. K. Soong, and R. W. Heath, "MmWave Vehicular Beam Selection With Situational Awareness Using Machine Learning," *IEEE Access*, vol. 7, pp. 87 479–87 493, 2019.

[67] Y. Oguma, T. Nishio, K. Yamamoto, and M. Morikura, "Proactive handover based on human blockage prediction using RGB-D cameras for mmWave communications," *IEICE Transactions on Communications*, vol. 99, no. 8, pp. 1734–1744, 2016.

[68] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. of Information Theory and Applications Workshop (ITA)*, 2019, pp. 1–8.

[69] D. Vasisht, S. Kumar, H. Rahul, and D. Katabi, "Eliminating channel feedback in next-generation cellular networks," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 398–411.

[70] F. Rottenberg, R. Wang, J. Zhang, and A. F. Molisch, "Channel extrapolation in FDD massive MIMO: Theoretical analysis and numerical validation," *arXiv preprint arXiv:1902.06844*, 2019.

[71] A. Ali, N. González-Prelcic, and R. W. Heath, "Millimeter wave beam-selection using out-of-band spatial information," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1038–1052, 2017.

[72] F. Maschietti, D. Gesbert, and P. de Kerret, "Coordinated Beam Selection in Millimeter Wave Multi-User MIMO Using Out-of-Band Information," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[73] y. b. Ian goodfellow and aaron courville, "Deep learning," 2018.

[74] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson, "Deep Convolutional Neural Networks for Massive MIMO Fingerprint-Based Positioning," in *28th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2017.* IEEE–Institute of Electrical and Electronics Engineers Inc., 2018.

[75] V. Savic and E. G. Larsson, "Fingerprinting-based positioning in distributed massive MIMO systems," in *2015 IEEE 82nd vehicular technology conference (VTC2015-Fall).* IEEE, 2015, pp. 1–5.

[76] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs," *Computer Networks*, vol. 47, no. 6, pp. 825–845, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128604002610

[77] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 763–776, 2016.

[78] K. Hornik, M. Stinchcombe, H. White *et al.*, “Multilayer feedforward networks are universal approximators.” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[79] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, “Efficient backprop,” in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[80] M. Alrabeiah and A. Alkhateeb, “Deep learning for mmWave beam and blockage prediction using sub-6 GHz channels,” *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, 2020.

[81] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic Differentiation in PyTorch,” in *NIPS Autodiff Workshop*, 2017.

[82] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[83] M. Alrabeiah. (2020). [Online]. Available: https://github.com/malrabeiah/VABT/tree/master/beam_only

[84] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[85] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, “NR: The New 5G Radio Access Technology,” *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.

[86] N. Gonzalez-Prelcic, A. Ali, V. Va, and R. W. Heath, “Millimeter-Wave Communication with Out-of-Band Information,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 140–146, 2017.

[87] Remcom. Wireless insite. [Online]. Available: http://www.remcom.com/wireless-insite

[88] M. Hashemi, C. E. Koksal, and N. B. Shroff, “Out-of-Band Millimeter Wave Beamforming and Communications to Achieve Low Latency and High Energy Efficiency in 5G Systems,” *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 875–888, 2018.

[89] M. Peter, K. Sakaguchi, S. Jaeckel, S. Wu, M. Nekovee, J. Medbo, K. Haneda, S. Nguyen, R. Naderpour, J. Vehmas *et al.*, “Measurement campaigns and initial channel models for preferred suitable frequency ranges,” *Deliverable D2*, vol. 1, p. 160, 2016.

[90] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, “Steering with eyes closed: Mm-Wave beam steering without in-band measurement,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2416–2424.

[91] D. Burghal, R. Wang, and A. F. Molisch, “Deep Learning and Gaussian Process based Band Assignment in Dual Band Systems,” *arXiv e-prints*, p. arXiv:1902.10890, 2019.

[92] F. B. Mismar, A. AlAmmouri, A. Alkhateeb, J. G. Andrews, and B. L. Evans, “Deep Learning Predictive Band Switching inWireless Networks,” *submitted to IEEE Transactions on Wireless Communications, arXiv preprint*, 2019.

[93] O. Semiari, W. Saad, and M. Bennis, “Joint Millimeter Wave and Microwave Resources Allocation in Cellular Networks With Dual-Mode Base Stations,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4802–4816, 2017.

[94] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, “Non-Line-of-Sight Identification and Mitigation Using Received Signal Strength,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1689–1702, 2015.

[95] A. Ali, N. Gonzlez-Prelcic, and R. W. Heath, “Spatial Covariance Estimation for Millimeter Wave Hybrid Systems using Out-of-Band Information,” *IEEE Transactions on Wireless Communications*, pp. 1–1, 2019.

[96] X. Li and A. Alkhateeb, “Deep learning for direct hybrid precoding in millimeter wave massive MIMO systems,” *arXiv preprint arXiv:1905.13212*, 2019.

[97] Junyi Wang, Zhou Lan, Chang-woo Pyo, T. Baykas, Chin-sean Sum, M. A. Rahman, Jing Gao, R. Funada, F. Kojima, H. Harada, and S. Kato, “Beam codebook based beamforming protocol for multi-Gbps millimeter-wave WPAN systems,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1390–1399, 2009.

[98] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh, “Millimeter Wave Beamforming for Wireless Backhaul and Access in Small Cell Networks,” *IEEE Transactions on Communications*, vol. 61, no. 10, pp. 4391–4403, 2013.

[99] W. Kotzé, “Lattice morphisms, sobriety, and Urysohn lemmas,” in *Applications of category theory to fuzzy subsets*. Springer, 1992, pp. 257–274.

[100] S. S. Haykin, S. S. Haykin, S. S. Haykin, and S. S. Haykin, *Neural networks and learning machines*. Pearson Upper Saddle River, NJ, USA:, 2009, vol. 3.

[101] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[102] M. Telgarsky, “Benefits of depth in neural networks,” *arXiv preprint arXiv:1602.04485*, 2016.

[103] M. Alrabeiah. (2020). [Online]. Available: https://github.com/malrabeiah/Sub6-Preds-mmWave

[104] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *J. Mach. Learn. Res.*, vol. 11, p. 3371–3408, Dec. 2010.

[105] D. Vasisht, S. Kumar, H. Rahul, and D. Katabi, "Eliminating channel feedback in next-generation cellular networks," in *Proceedings of the 2016 ACM SIGCOMM Conference*, 2016, pp. 398–411.

[106] M. Alrabeiah and A. Alkhateeb, "Deep learning for TDD and FDD massive MIMO: Mapping channels in space and frequency," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1465–1470.

[107] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[108] D. J. De Waal, *Matrix-Valued Distributions*. American Cancer Society, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0471667196.ess1565.pub2

[109] A. Leon-Garcia, "Probability, statistics, and random processes for electrical engineering," 2017.

[110] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[111] M. Alrabeiah. (2020, June) Statistical channel prediction: A regression and clustering approaches. [Online]. Available: https://github.com/malrabeiah/stat_ch_pred

[112] S. Eckelmann, T. Trautmann, H. Ußler, B. Reichelt, and O. Michler, "V2V-Communication, LiDAR System and Positioning Sensors for Future Fusion Algorithms in Connected Vehicles," *Transportation Research Procedia*, vol. 27, pp. 69 – 76, 2017, 20th EURO Working Group on Transportation Meeting, EWGT 2017, 4-6 September 2017, Budapest, Hungary.

[113] K. Nakashima, Y. Koda, K. Yamamoto, H. Okamoto, T. Nishio, M. Morikura, Y. Asai, and R. Miyatake, "Impact of Input Data Size on Received Power Prediction Using Depth Images for mm Wave Communications," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Aug 2018, pp. 1–5.

[114] A. Klautau, N. González-Prelcic, and R. W. Heath, "LIDAR Data for Deep Learning-Based mmWave Beam-Selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, June 2019. [Online]. Available: https://www.lasse.ufpa.br/raymobtime/

[115] A. Ali, N. G. Prelcic, R. W. Heath, and A. Ghosh, "Leveraging Sensing at the Infrastructure for mmWave Communication," *ArXiv*, vol. abs/1911.09796, 2019.

[116] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, “Millimeter wave base stations with cameras: Vision-aided beam and blockage prediction,” in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[117] [Online]. Available: http://www.viwi-dataset.net

[118] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, “Deep learning coordinated beamforming for highly-mobile millimeter wave systems,” *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[119] V. M. De Pinho, M. L. R. De Campos, L. U. Garcia, and D. Popescu, “Vision-Aided Radio: User Identity Match in Radio and Video Domains Using Machine Learning,” *IEEE Access*, vol. 8, pp. 209 619–209 629, 2020.

[120] H. Zou, J. Yang, H. Prasanna Das, H. Liu, Y. Zhou, and C. J. Spanos, “WiFi and vision multimodal learning for accurate and robust device-free human activity recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[121] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, “Through-wall human pose estimation using radio signals,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7356–7365.

[122] T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, “Making the invisible visible: Action recognition through walls and occlusions,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 872–881.

[123] A. Alahi, A. Haque, and L. Fei-Fei, “RGB-W: When Vision Meets Wireless,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3289–3297.

[124] A. Alkhateeb, G. Leus, and R. W. Heath, “Limited Feedback Hybrid Precoding for Multi-User Millimeter Wave Systems,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6481–6494, 2015.

[125] T. Moon, J. Gaun, and H. Hassanieh, “Online Millimeter Wave Phased Array Calibration Based on Channel Estimation,” in *2019 IEEE 37th VLSI Test Symposium (VTS)*, 2019, pp. 1–6.

[126] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[127] R. Girshick, “Fast R-CNN,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[128] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

APPENDIX A

BIJECTIVENESS OF CHANNEL-TO-POSITION FUNCTION

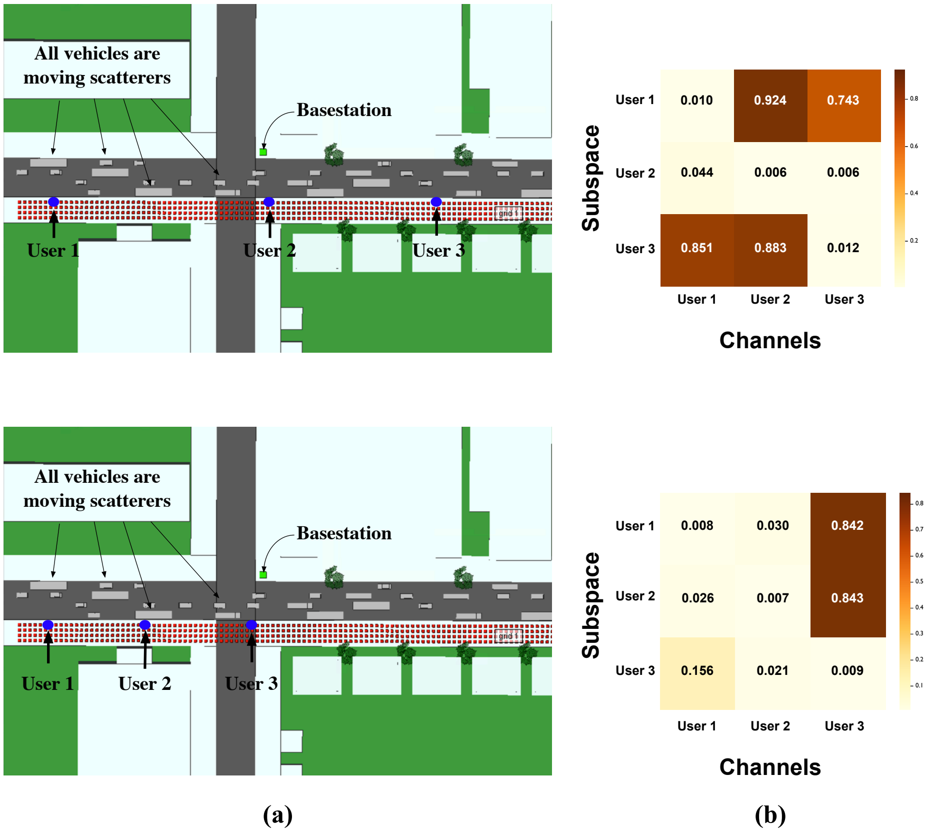


Figure A.1: (a) Top-views of from the "O2_dyn_3p4" scenario showing the positions of selected users and basestation. (b) two heat-maps for the NMSE of the three users and their subspaces.

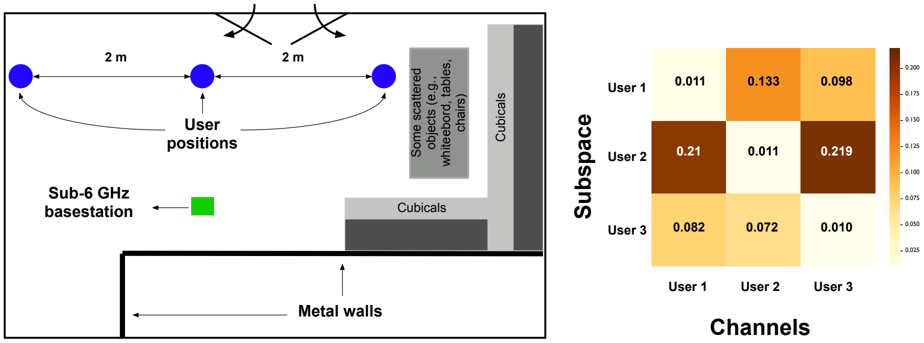
Despite the fact that I do not have a mathematical proof for its existence, I think bijectiveness is a reasonable assumption for two reasons, which I explore below.

A.1 Empirical Study:

I have developed an empirical study to measure bijectiveness in realistic wireless environments (whether with stationary or dynamic scatterers). That study is inspired by the work on joint spatial division and multiplexing in [47] The authors there posit that the massive MIMO M-dimensional channel vector of a user is very likely to have a low dimensional structure, i.e., lives in a low-dimensional subspace in the M-dimensional vector space. They verify that proposition in their experimental results by showing that a 128 dimensional channel vector could be estimated using the channel covariance and 30 to 40 pilots—as briefly discussed in Comment 2.

Proposed measure of bijectiveness:

Given their findings, I think a good measure of bijectiveness (a soft measure) could be obtained by identifying channel subspaces for different users and measuring how much



(a) **(b)**

Figure A.2: (a) A schematic of the lab where the real channel measurements are collected showing the positions of the users and the basestation. (b) a heatmap for the NMSE of the three users and their subspaces.

overlap there is between those subspaces. This overlap is measured using the following four steps:

1. For the u-th user at position $\mathbf{x}_u \in \mathbb{R}^3$, estimate its channel covariance using N channel samples

$$\mathbf{C}_u = \frac{1}{N-1} \sum_{n=1}^{N} \left(\mathbf{h}_\mathbf{n}^{(\mathbf{u})} - \boldsymbol{\mu}\right) \left(\mathbf{h}_\mathbf{n}^{(\mathbf{u})} - \boldsymbol{\mu}\right)^H, \tag{A.1}$$

   where $\mathbf{h}_u^{(u)} \in \mathbb{C}^{M \times 1}$ is the n-th channel vector of the u-th user at all M basestation antennas, and $\boldsymbol{\mu} = (1/N) \sum_{n=1}^{N} \mathbf{h}_n^{(u)}$ is the estimated channel mean.

2. Apply eigen-decomposition on the covariance $\mathbf{C}_u$, and identify the 99%-energy subspace. This is done by sorting the eigne values in a descending order and finding the $r(< M)$ eigen-values that satisfy

$$\frac{\sum_{m=1}^{r} \lambda_m}{\sum_{m=1}^{M} \lambda_m} \geq 0.99, \tag{A.2}$$

   where λ_m is the m-th eigen-value. The eigen-vectors associated with the r eignevalues define the basis matrix $\tilde{\mathbf{V}}_u \in \mathbb{C}^{M \times r}$ of the 99%-subspace of user u.

3. For another u'-th user at a different position $\mathbf{x}_{u'} \in \mathbb{R}^3$ in the same wireless environment, project its N channels—assuming the same number of channel samples to that of the u-th user—onto the subspace defined by $\tilde{\mathbf{V}}_u$. Formally, this is given by

$$\tilde{\mathbf{h}}_n^{(u')} = \tilde{\mathbf{V}}_u \tilde{\mathbf{V}}_u^H \mathbf{h}_n^{(u')}. \tag{A.3}$$

4. Compute the average Normalized Mean Squared Error (NMSE) between projected channels and original ones as follows

$$\text{NMSE} = \frac{1}{N} \sum_{n=1}^{N} \frac{||\tilde{\mathbf{h}}_n^{(u')} - \mathbf{h}_n^{(u')}||_2^2}{||\tilde{\mathbf{h}}_n^{(u')}||_2^2}. \tag{A.4}$$

where $||.||_2$ is the Euclidean norm.

The NMSE in Equation A.4 could serve as a measure of how similar the channels of user u' at position $\mathbf{x}_{u'}$ to those of user u at position $\mathbf{x}_u$, i.e., measuring how much overlap there is between the channel subspaces of the two users. When the NMSE is quite large, it, then, points to the likelihood that the channels of user u are quite different from those of user u'. It is important to note here that this measure of bijectivenes is not claimed to be definitive. It rather provides a sense of how different the channels of two users are.

Experimental settings and findings:

To perform my study, I collect two different sets of channels, namely $\mathcal{D}_1$ and $\mathcal{D}_2$. The first has synthetic channels obtained from a DeepMIMO scenario while the other has real channel measurements obtained in the lab. The DeepMIMO "O2_dyn_3p4" scenario is used, which represents a wireless environment with dynamic scatterers and stationary users and basestation. The scenario deploys a basestation with 64-element Uniform Linear Array (ULA) operating at 3.4 GHz and has a grid of 760 users that are uniformly spaced (2 meters between any two users). Two top-views from that scenario are shown in Figure A.1-(a). The other set of channels ($\mathcal{D}_2$) is constructed of channels obtained in an indoor environment. The basestation in this case deploys 4-element ULA and 64 subcarriers, and the user assumes one of three position that are 2 meters apart, see Figure A.2-(a).

Using the two sets, I apply the proposed measure in Section A.1 and plot the results in the form of error heatmaps as shown in Figures A.1-(b) and A.2-(b). The heatmaps, in general, show some interesting results. Using the synthetic data, the top heatmap in Figure A.1-(b) shows that channels of the three users should be very different. For example, User 1 subspace, which has a rank of $r = 9$, achieves 0.01 NMSE with its own channels (projecting its own channels onto its 99%-subspace) while it achieves NMSE of 0.74 with User 3. This suggests that User 1's channels live in a different subspace to that of User 2 and 3. However, the subspace of User 2, which has a rank of $r = 29$, records the same NMSE with User 3 to that of its own channels, i.e., NMSE = 0.006. For this specific case, the two users' channels should not be expected to be quite. different. Similar results are expected to be obtained with real channel measurements, and this is due to the heatmap shown in Figure A.2-(b). Overall, it indicates that users' channels have some differences, yet they are not quite as distinct as the differences observed in Figure A.1. This could be attributed to several factors, three of the most important ones are:

1. The nature of the wireless environment. One is an indoor environments while the other is outdoor.

2. The nature of the two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$. The former has synthetic data samples while the other has real measurements.

3. The spacing between the users in the two figures. The indoor environment does not allow for wide spacing such as that in the outdoor environment. Maximum distance between two users is $\approx 5\ m$ in the lab.

APPENDIX B

NOTES ABOUT FRAMEWORKS 1 AND 2

This appendix attempts to present a couple of notes on the relation between coherence time and bandwidth and the proposed deterministic- and statistical-channel prediction frameworks. Both are general in the sense that they transcend the concepts of coherence time or bandwidth. I will try to explain this in the following.

B.1 Coherence time

The deterministic channel prediction framework is general in the sense that it applies equally to stationary wireless environments (i.e., static scatterers and users) and dynamic ones (i.e., moving scatterers and users). For stationary environments, the wireless channel between a user at a certain position and a set of antennas does not change over time (i.e., it is time invariant). Hence, the change in the channel is completely defined by the user and antenna set positions in the environment, and this is captured in the deterministic channel-prediction framework, see Section 2.4 in the comprehensive monograph. For dynamic environments, the bijectiveness assumption could be extended to time-varying channels; if the position-to-channel function (Equation 2.3, Section 2.4.2 in the monograph) remains bijective over time, the essence of deterministic channel prediction holds, and at any time instance t, the channel-to-channel function (Equation 2.6, Proposition 1 in the monograph) could be shown to exist. Hence, one could say that the deterministic prediction framework is not restricted by coherence time.

Similar thing could be said about the statistical channel prediction framework, for it, in essence, attempts to learn a conditional covariance (Equation 4.4, Section 4.5.1 in the monograph) that characterizes the large-scale statistics (large-scale fading) of the channel between a user and a set of antennas [47]. Such large-scale statistics are not restricted by coherence time.

B.2 Coherence bandwidth

Whether we assume a stationary or a dynamic environment, deterministic channel prediction targets learning the channel-to-channel function (Equation 2.6, Proposition 1 in the monograph) that governs the relation between two sets of channels. The target function could be broken down into two components. The first component is the channel-to-position function (Equation 2.5, Section 2.4.2 in the monograph) which describes the relation between the observed channel at the first set of antennas and frequency f_1 and a user at some position $\mathbf{x}_u$. The second component is the position-to-channel (Equation 2.4, Section 2.4.2 in the monograph) function describing the relation between the same user position and the channel at the second set of antennas and another frequency f_2. Both components are functions of their own channel parameters like positions, delays, paths, angles, path gains,..etc. Hence, learning the composition function means that the machine learning algorithm implicitly models the two components, and by doing so, the algorithm learns the environment response to signals propagating at two different frequencies, i.e., f_1 and f_2. This indicates that the algorithm is not restricted by the coherence bandwidth at either of the two frequencies. A good evidence for that is the study case presented in Chapter 3 on predicting mmWave beams from sub-6 GHz channels. As for the statistical prediction framework, it learns large-scale statistics of the channel between a user and a set of antennas, and, therefore, it is not restricted by coherence bandwidth.

APPENDIX C

RELATION BETWEEN SAMPLE COVARIANCE AND CONDITIONAL COVARIANCE

When $\boldsymbol{\mu}_{\mathbf{h}_{DL}|\mathbf{h}_{UL}} = 0$, the conditional covariance defined in 4.4 could be decomposed into real and imaginary parts as follows

$$\mathbf{C} = \mathbf{C}_r + j\mathbf{C}_{im}, \tag{C.1}$$

where:

$$\mathbf{C}_r = \begin{bmatrix} \mathbb{E}[(h_1^r)^2] & \dots & \mathbb{E}[h_1^r h_{M_2}^r] + \mathbb{E}[h_1^{im} h_{M_2}^{im}] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[h_1^r h_{M_2}^r] + \mathbb{E}[h_1^{im} h_{M_2}^{im}] & \dots & \mathbb{E}[(h_{M_2}^r)^2] \end{bmatrix} \tag{C.2}$$

$$\mathbf{C}_{im} = \begin{bmatrix} \mathbb{E}[(h_1^{im})^2] & \dots & \mathbb{E}[h_1^{im} h_{M_2}^r] - \mathbb{E}[h_1^r h_{M_2}^{im}] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[h_1^{im} h_{M_2}^r] - \mathbb{E}[h_1^r h_{M_2}^{im}] & \dots & \mathbb{E}[(h_{M_2}^r)^2] \end{bmatrix}. \tag{C.3}$$

On the other hand, the conditional sample covariance could be expanded as follows

$$\mathbb{E}[\mathbf{C}_s|\tilde{\mathbf{h}}_{UL}] = \begin{bmatrix} \mathbb{E}[(h_1^r)^2] & \dots & \mathbb{E}[h_1^r h_{M_2}^r] & \mathbb{E}[h_1^r h_1^{im}] & \dots & \mathbb{E}[h_1^r h_{M_2}^{im}] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}[h_{M_2}^r h_1^r] & \dots & \mathbb{E}[(h_{M_2}^r)^2] & \mathbb{E}[h_{M_2}^r h_1^{im}] & \dots & \mathbb{E}[h_{M_2}^r h_{M_2}^{im}] \\ \mathbb{E}[h_1^{im} h_1^r] & \dots & \mathbb{E}[h_1^{im} h_{M_2}^r] & \mathbb{E}[(h_1^{im})^2] & \dots & \mathbb{E}[h_1^{im} h_{M_2}^{im}] \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbb{E}[h_{M_2}^{im} h_1^r] & \dots & \mathbb{E}[h_{M_2}^{im} h_{M_2}^r] & \mathbb{E}[h_{M_2}^{im} h_1^{im}] & \dots & \mathbb{E}[(h_{M_2}^{im})^2], \end{bmatrix} \tag{C.4}$$

given that $\boldsymbol{\mu}_{\tilde{\mathbf{h}}_{DL}|\tilde{\mathbf{h}}_{UL}} = 0$, as well. From (C.2), (C.3), and (C.4), one could readily see that every element in both $\mathbf{C}_r$ and $\mathbf{C}_{im}$ could be constructed from those of $\mathbb{E}[\mathbf{C}_s|\tilde{\mathbf{h}}_{UL}]$.