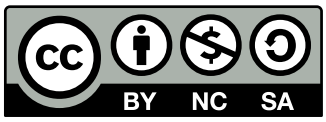
Aalto University
School of Electrical
Engineering

Master's Programme in Computer, Communication and Information Sciences

Throughput Optimized Channel Smoothing

Ruixin Xu

Master's Thesis
2025



Aalto University
School of Electrical
Engineering

Author Ruixin Xu

Title Throughput Optimized Channel Smoothing

Degree programme Computer, Communication and Information Sciences

Major Speech and Language Technology

Supervisor Prof. Sergiy A. Vorobyov

Advisors Dr. Jean-Luc Olives, Mr. Eeli Susan (MSc)

Collaborative partner Nokia Sol. & Networks Oy

Date September 2025 **Number of pages** 55 **Language** English

Abstract

Wireless communication systems have traditionally relied on carefully designed receiver algorithms, such as channel estimation, to mitigate impairments and ensure reliable data transmission. Although these traditional methods have been refined over decades, they often fall short when faced with the complexity of real-world wireless environments. In recent years, Artificial Intelligence (AI) has become a promising tool to enhance or even replace conventional receiver components. However, many AI-based methods in channel estimation still rely on distance-based loss functions for channel smoothing. These objectives may not correlate well with overall system performance and can lead to overfitting, especially when estimation errors have little impact on throughput.

This thesis investigates the use of AI for channel estimation, especially channel smoothing, in the 5G receiver chain with a particular focus on the Physical Uplink Shared Channel. A novel neural network is proposed that integrates Convolutional Neural Network and Residual neural Network, trained with a system-level, throughput-oriented objective function. By leveraging a fully differentiable receiver chain, the neural network is able to directly optimize decoding performance at the bit level. This method is compared against a conventional AI method using the same neural network architecture but optimized with a loss based on ideal channel conditions.

Both methods are trained under simulated 5G channel environments Clustered Delay Line (CDL) A/B/D and evaluate under CDL C/E, using the `Sionna` simulation framework. Results demonstrate that the throughput-optimized neural network consistently outperforms traditional least squares estimation in terms of Bit Error Rate and Block Error Rate, and in certain scenarios also surpasses the neural network trained with ideal channel conditions. Because ideal channel conditions are only available in simulation, these results highlight the practical value of learning directly for throughput and point toward a viable path for AI-enhanced receivers in real-world 5G deployments.

Keywords Channel Estimator, Neural Network, Throughput Optimization, Channel Smoothing

Preface

I want to thank Professor Sergiy A. Vorobyov and my instructors Dr. Jean-Luc Olives and Mr. Eeli Susan (MSc) for their guidance.

Otaniemi, September 2025

Ruixin Xu

Contents

Symbols and abbreviations

Abbreviations

3GPP	the 3rd Generation Partnership Project
AI	Artificial Intelligence
Adam	Adaptive Moment Estimation
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BLER	Block Error Rate
CDL	Clustered Delay Line
CDM	Code Division Multiplexing
CNN	Convolutional Neural Network
CP	Cyclic Prefix
CRC	Cyclic Redundancy Check
DMRS	DeModulation Reference Signals
DL	Deep Learning
eMBB	enhanced Mobile BroadBand
FDM	Frequency-Division Multiplexing
FNN	Feedforward Neural Network
GSMA	Global System for Mobile communications Association
IP	Internet Protocol
ISI	InterSymbol Interference
LDPC	Low-Density Parity-Check
LLRs	Log Likelihood Ratios
mMTC	massive Machine-Type Communications
MIMO	Multiple-Input Multiple-Output
MMSE	Minimum Mean Square Error
NR	New Radio
OCC	Orthogonal Cover Code
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
PUSCH	Physical Uplink Shared CHannel
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RB	Resource Block
RE	Resource Element
ReLU	Rectified Linear Unit
ResNet	Residual neural Network
UE	User Equipment
SCS	SubCarrier Spacing
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise Ratio
TDL	Tapped Delay Line
URLLC	Ultra-Reliable Low-Latency Communications

1 Introduction

With the rapid advancement of mobile communication technologies, 5G has emerged as an essential enabler of high-speed, low-latency, and highly reliable wireless communication, as demonstrated in several studies [1, 2, 3]. Sophisticated physical layer is key to the performance of 5G systems, in which channel impairments such as noise, fading, and interference are effectively mitigated to ensure robust and efficient data transmission [4].

Engineered receiver algorithms within several components, including channel estimation, are used to support reliable data transmission in 5G physical layer [5]. Although these algorithms have been refined over decades, they often fall short in capturing the full complexity of real-world wireless environments [6]. In recent years, artificial intelligence (AI) techniques have emerged as powerful tools to enhance or even replace some traditional receiver components. Leveraging their data-driven nature, AI offer the potential to significantly improve the system performance in dynamic and non-linear channel conditions commonly encountered in practical deployments [7].

Although AI techniques perform better then traditional components in the receiver algorithms [8], most AI methods primarily focus on minimizing errors within specific components of the receiver architecture [9], rather than addressing system-level performance. For example, AI technique in channel smoothing component often optimize distance-based metrics leading to suboptimal overall performance or increased neural network complexity. Moreover, strict reliance on such distance-based loss functions can result in overfitting [10], especially when the neural network pays attention to errors that have the least impact on system performance. Consequently, there is a clear need for a methodology that prioritizes throughput optimization over intermediate channel smoothing metrics.

This thesis aims to design and implement a throughput optimized AI method for channel smoothing which is involved in channel estimation, and compare it with a conventional AI method that uses the same neural network architecture but different optimization strategies. The neural network architecture, as referenced by [11], combines Convolutional Neural Network (CNN) and Residual neural Network (ResNet) to denoise raw channel estimates from a preceding component in channel estimation and interpolate all channel estimates for equalizer in 5G receiver. The optimization strategy used to update the learnable parameters of the throughput optimized AI method is based on the probability of predicted bits. This optimization at the system level can be achieved due to the fully differentiable receiver chain. Conversely, the optimization strategy for the conventional AI method is solely based on the distance between ideal channel conditions and the channel estimates.

This thesis focuses on the Physical Uplink Shared CHannel (PUSCH) within the physical layer of 5G systems, with particular emphasis on the receiver design. The study is focused on integrating neural networks into the receiver chain to enhance traditional receiver algorithms, rather than replacing the entire receiver with end-to-end AI architectures. Although the deployment scenarios are simulated, the proposed approach demonstrates strong potential for real-world application.

The rest of this thesis is organized as follows. Chapter 2 provides an introduction

to 5G New Radio (5G NR), outlining key concepts relevant to the thesis. Chapter 3 presents an overview of neural network architectures, with a particular focus on Feedforward Neural Network (FNN), CNN and ResNet. Chapter 4 reviews related work in the field of channel estimation and details the proposed methodology. Chapter 5 describes the data characteristics and the experiments. Chapter 6 offers a comprehensive analysis and discussion of the results. Finally, Chapter 7 concludes the thesis and proposes potential directions for future research.

2 5G New Radio

5G mobile communications marks a significant milestone in the evolution of wireless technology. It is driven by the need to support use cases with high requirements on data rates, latency, reliability, and connectivity. These use cases shown in Figure 1 include enhanced Mobile BroadBand (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine-Type Communications (mMTC) [12], posing unique challenges on the design of 5G physical layer.

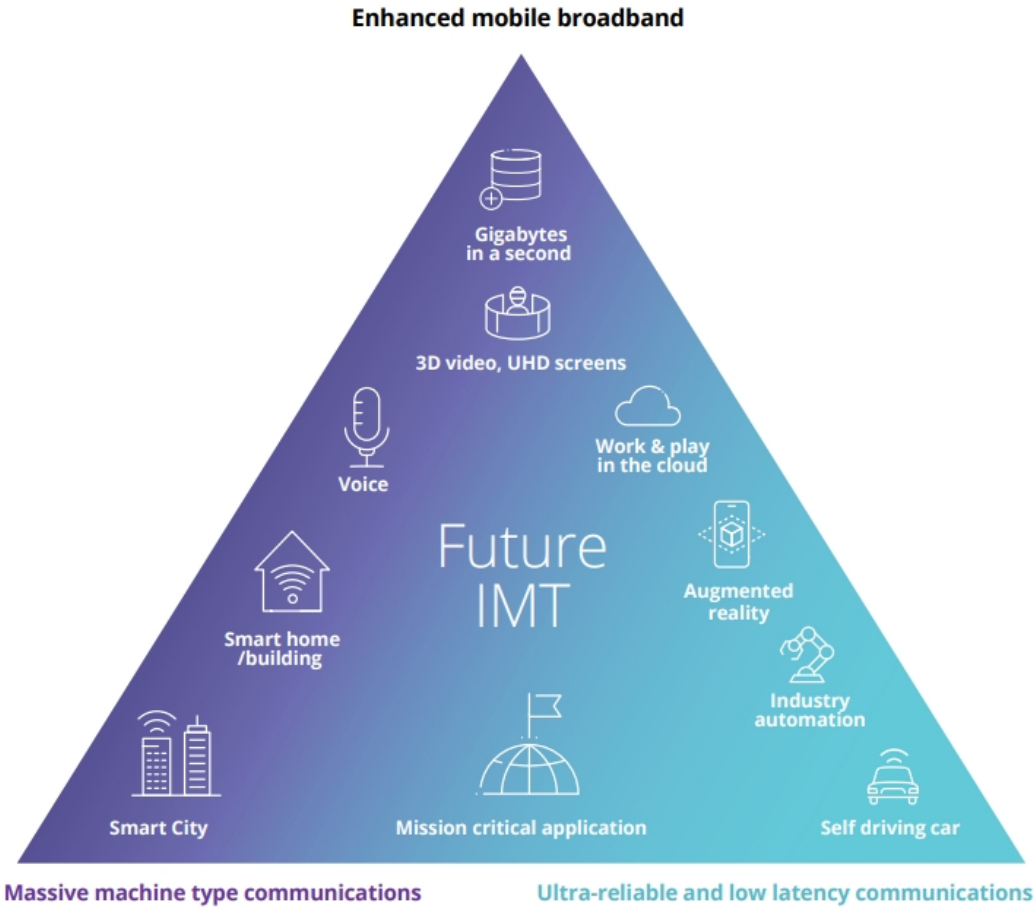


Figure 1: 5G Usage Scenarios [12].

5G physical layer covers critical concepts such as the spectrum, Orthogonal Frequency-Division Multiplexing (OFDM), and Resource Grid (RG). Additionally, 5G uplink transmission is introduced with an emphasis on the PUSCH and the role of DeModulation Reference Signals (DMRS). Finally, the conditions of propagation channel models are explored in simulation environments where we could achieve ideal channel conditions for channel estimation.

2.1 Spectrum in 5G

Wireless communication relies on the radio spectrum, which is divided into different frequency bands used to transmit data over the air [13]. This large range of frequency bands can be divided into Low bands, Mid bands, and High bands in 5G (See Figure 2).

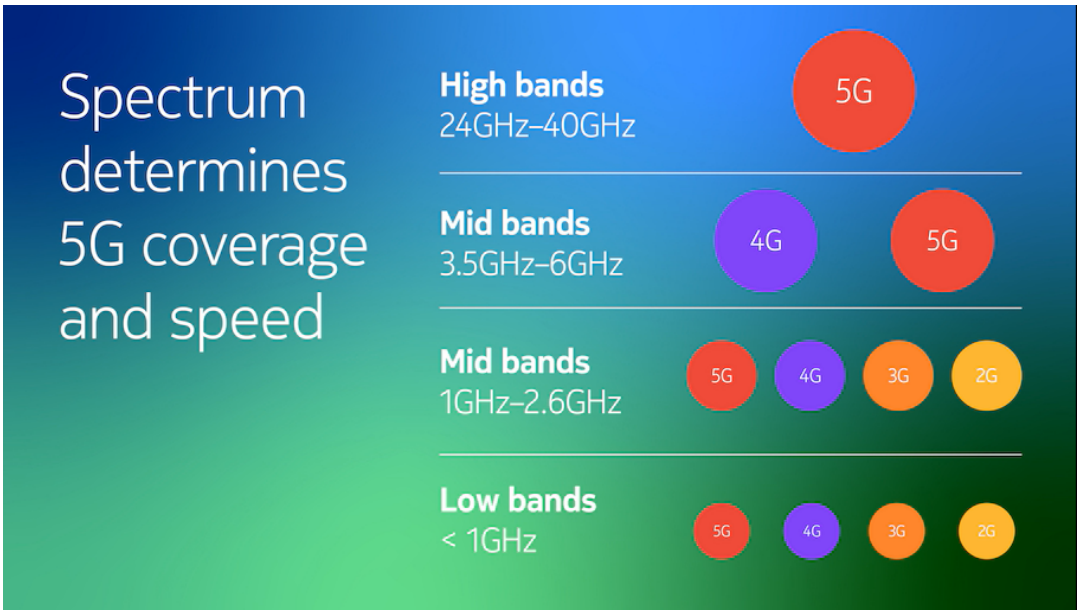


Figure 2: Frequency bands for 5G wireless communication [14].

Low band spectrum is typically below 1 GHz. It is suitable for rural and wide-area deployments with extensive coverage and strong penetration capabilities. Mid band spectrum tend to range from 1 GHz to 6GHz. It can carry plenty of data while traveling significant distances and make up the majority of the total frequencies assigned. As declared by Global System for Mobile communications Association (GSMA), the 3.3 GHz to 3.8 GHz range of the spectrum is ideal as many countries in the world have already designated it for 5G. Other mid band spectrum is in use as well. High band frequencies refer to the spectrum above 24 GHz, where the signal is called millimeter wave (mmWave). It provides extremely high data rates and capacity, enabling ultra-fast communication, but mmWave has limited penetration and shorter coverage ranges. Thus, advanced network planning and deployment of small cells is necessary to enhance connectivity.

The demand for eMBB, URLLC and mMTC requires well-suited spectrum allocation to maximize efficiency and minimize interference. Spectrum allocation influences key network performance factors, including data throughput, coverage, and interference mitigation.

2.2 Orthogonal Frequency Division Multiplexing

OFDM is a widely adopted multi-carrier modulation technique used in modern broadband communication systems, including 5G NR. It is a specialized form of

Frequency Division Multiplexing (FDM), designed to enhance spectral efficiency and data transmission rates. Unlike traditional FDM assigning non-overlapping frequency bands to different data streams, OFDM employs a large number of closely spaced orthogonal subcarriers within the same frequency channel.

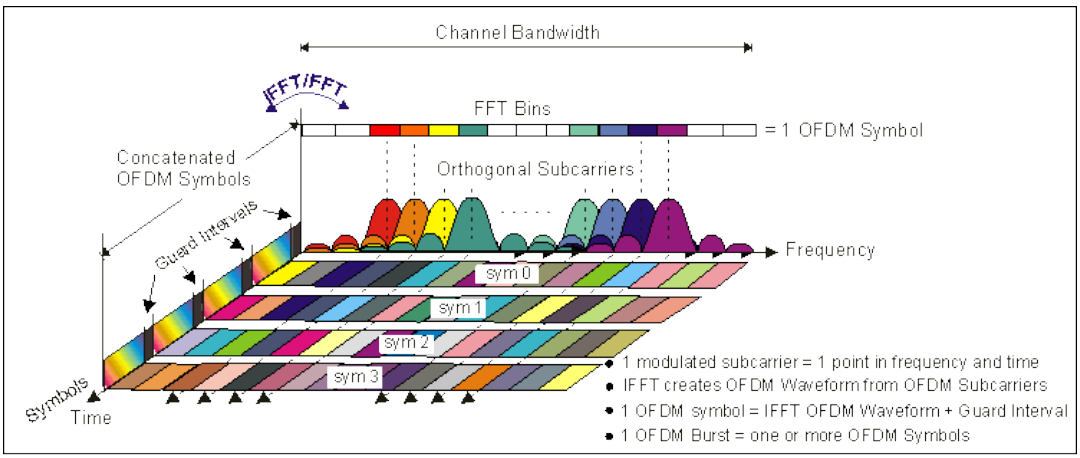


Figure 3: Frequency-time representation of an OFDM signal [15]. The horizontal axis represents frequency, while the vertical axis represents time. The time domain shows discrete OFDM symbols, each preceded by a CP to mitigate ISI. The frequency domain illustrates modulated subcarriers, each corresponding to a unique point in the time-frequency grid.

As illustrated in Figure 3, the OFDM signal spans both time and frequency domains. In the time domain, the signal is segmented into discrete OFDM symbols, each of which is preceded by a guard interval known as the Cyclic Prefix (CP). This CP can effectively mitigate InterSymbol Interference (ISI) where one symbol interferes with following symbols [16]. ISI arises when transmitted symbols overlap at the receiver due to delayed signal reflections caused by obstacles in the transmission path. These delays result in multiple versions of the transmitted signal arriving at different times, with some delayed copies of one symbol interfering with subsequent symbols. This temporal overlap compromises the receiver's ability to distinguish between adjacent symbols.

The CP is constructed by copying the last segment of an OFDM symbol and appending it to the beginning of that symbol before transmission. This redundancy ensures that the delayed copies of a symbol that extend into the beginning of the next symbol are confined within the CP rather than corrupting the actual symbol data in the presence of multipath delays. Provided the maximum delay spread of the channel is less than or equal to the duration of the CP, ISI can be effectively eliminated. At the receiver, the CP is removed prior to demodulation. Thereby, it discards the portion of the signal potentially affected by interference, while preserving the integrity of the core OFDM symbol.

In the frequency domain, each OFDM symbol comprises multiple orthogonal subcarriers. Orthogonality ensures that the peak of one subcarrier coincides with the

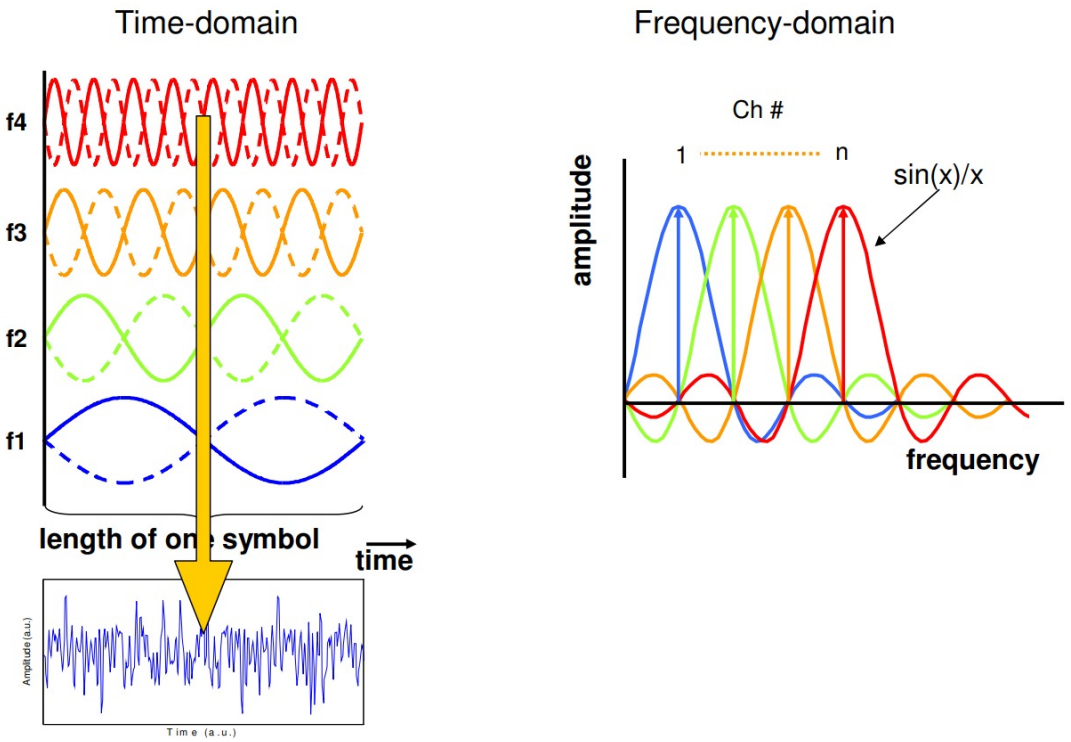


Figure 4: Illustration of orthogonality in OFDM systems [17]. The time-domain representation (left) shows multiple subcarriers (f1, f2, f3, f4) with different frequencies. The frequency-domain representation (right) demonstrates how the peak of each subcarrier aligns with the nulls of others, effectively eliminating ICI. This orthogonality enables efficient spectral resource utilization and facilitates implementation using the FFT and its inverse IFFT.

nulls of all other subcarriers, shown in Figure 4, effectively eliminating inter-carrier interference (ICI). This property enables efficient use of spectral resources and allows for straightforward implementation using the Fast Fourier Transform (FFT) and its inverse (IFFT).

The subcarriers in an OFDM system are modulated with data symbols, such as N-Quadrature Amplitude Modulation (QAM), resulting in a composite signal that is the sum of these modulated carriers. As depicted in the Figure 3, each modulated subcarrier corresponds to a unique point in the time-frequency grid, where the horizontal axis represents frequency and the vertical axis represents time. A single OFDM burst may consist of one or more such OFDM symbols, each occupying the entire channel bandwidth.

2.3 Numerology

In 5G NR, numerology is defined by the parameter μ as shown in detail in Table 1, which governs primarily SubCarrier Spacing (SCS). As illustrated in Figure 5, the hierarchical time-domain framework comprises frames, subframes, slots, and OFDM

μ	$\Delta f = 2^{\mu} \cdot 15$ [kHz]	**CP**
0	15	Normal
1	30	Normal
2	60	Normal, Extended
3	120	Normal
4	240	Normal

Table 1: Supported transmission numerologies.

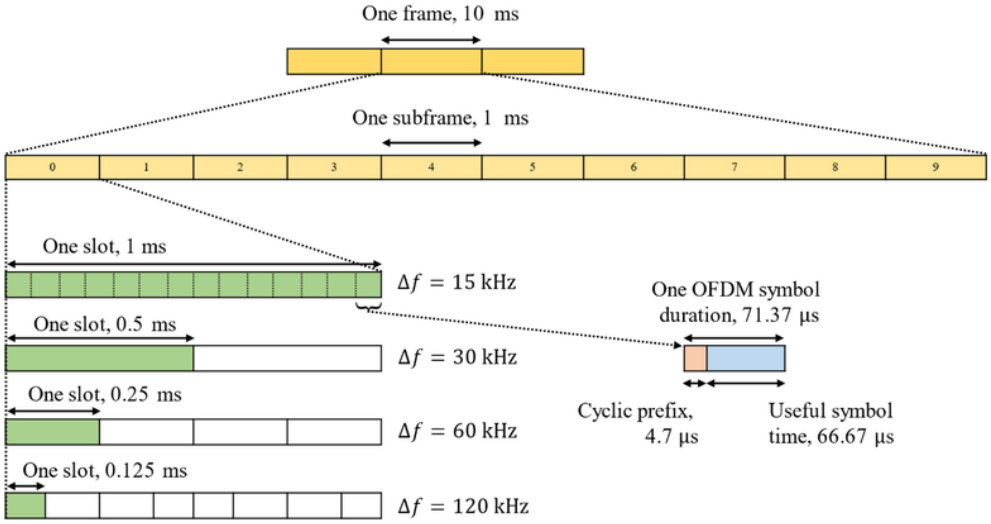


Figure 5: Time-domain structure of 5G NR illustrating the hierarchical framework of frames, subframes, slots, and OFDM symbols [18]. The figure shows how slot duration varies with SCS Δf, demonstrating durations of 1 ms for $\Delta f = 15$ kHz, 0.5 ms for $\Delta f = 30$ kHz, 0.25 ms for $\Delta f = 60$ kHz, and 0.125 ms for $\Delta f = 120$ kHz. Each slot typically contains 14 OFDM symbols, with a detailed view of the OFDM symbol duration, including a CP of 4.7 μs and a useful symbol time of 66.67 μs.

symbols. A frame spans 10 ms and is partitioned into ten 1 ms subframes. Each subframe is further divided into slots, whose duration inversely depends on the SCS Δf.

The slot duration scales as $T_{\text{slot}} = \frac{1}{2^{\mu}}$ ms. As shown in Figure 5, a slot tend to contains 14 OFDM symbols but the duration of a slot is different depending on various value of μ. Higher numerology correspond to increased subcarrier spacing. However, the slot duration is inversely proportional to the subcarrier spacing. This figure illustrates slot durations of 0.5 ms for $\mu = 1$ (30 kHz) , 0.25 ms for $\mu = 2$ (60 kHz) and 0.125 ms for $\mu = 3$ (120 kHz).

Each OFDM symbol comprises a CP and a useful symbol time. For $\Delta f = 15$ kHz, the symbol duration is 71.37 μs, with a CP of 4.7 μs and a useful time of 66.67 μs. Notably, $\mu = 2$ supports an extended CP to mitigate ISI in challenging channel conditions, though the default CP remains normal for most numerologies.

The flexibility in μ enables 5G to address diverse deployment scenarios [19]. Lower SCS such as 15 kHz, enhances robustness against delay spread in large cells, while higher SCS, such as 120 kHz and 240 kHz, reduces latency and supports high-frequency bands for ultra-reliable applications. Additionally, the 15 kHz subcarrier spacing ensures compatibility between 5G and 4G, as 4G only supported 15 kHz.

2.4 Resource Grid

The RG is a matrix representation of time-frequency resources over a specific bandwidth and time duration (one subframe). A Resource Block (RB) is a fundamental unit of resource allocation defined in the frequency domain. It comprises 12 consecutive subcarriers and is a little bit ambiguous in the time domain. Its minimum time duration

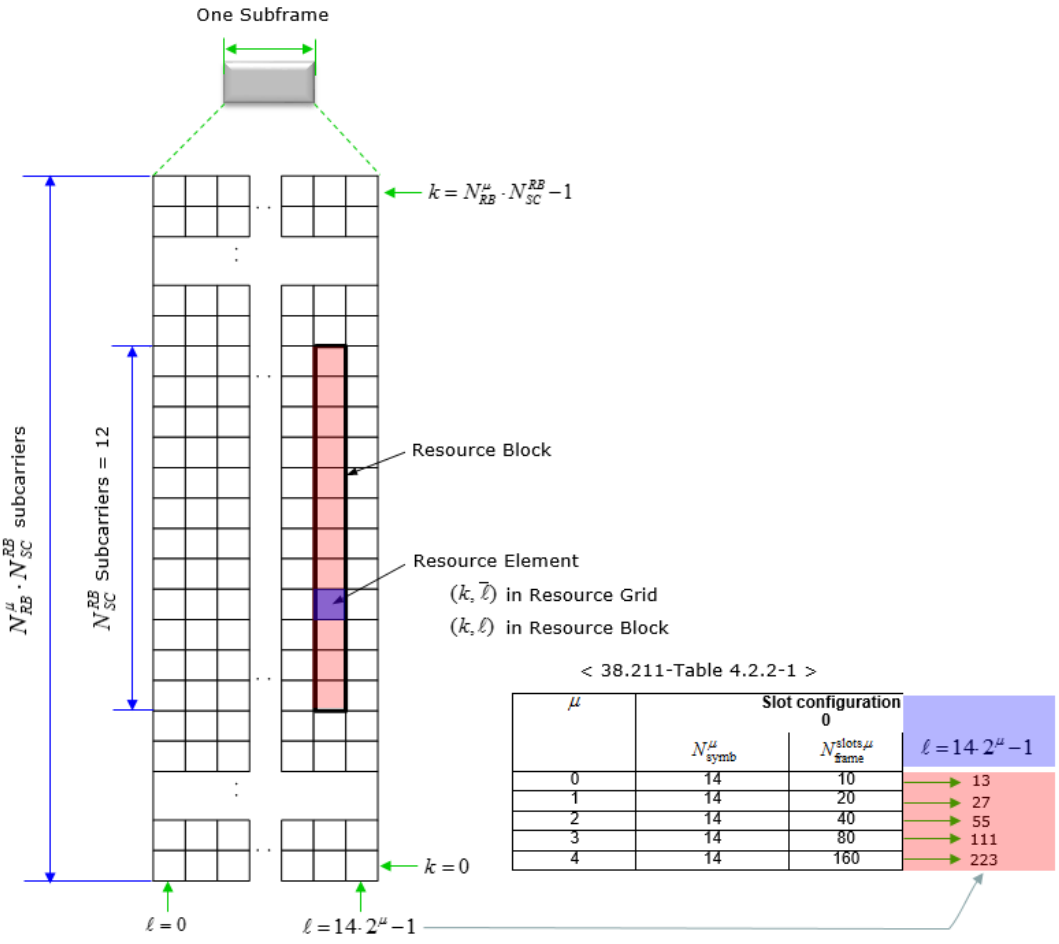


Figure 6: RG structure [20]. This figure illustrates the matrix representation of time-frequency resources within one subframe, highlighting the arrangement of RBs and REs. Each RB consists of 12 consecutive subcarriers, with the smallest time-frequency unit being the RE, defined by a subcarrier and an OFDM symbol index. The diagram also indicates the indexing of subcarriers and the configuration of slots based on different numerologies.

is only one OFDM symbol. As shown in Figure 6, each RB consists of resource elements (REs). The RB is indexed in frequency by k, where $k = 0$ corresponds to the lowest subcarrier index and $k = N^{\mu}_{\mathrm{RB}} \cdot N^{\mathrm{RB}}_{\mathrm{SC}} - 1$ corresponds to the highest. N^{μ}_{RB} denotes the number of RBs for specific numerology μ, and $N^{\mathrm{RB}}_{\mathrm{SC}} = 12$ is the number of subcarriers per RB.

A RE is the smallest unit of time-frequency resource and represents one subcarrier for one OFDM symbol, which refers to a QAM-modulated symbol carrying data or a pilot symbol. Each RE is uniquely identified by the pair (k, ℓ), where k denotes the subcarrier index and ℓ is the OFDM symbol index. Within a RB, the RE is located at position (k, ℓ).

2.5 Uplink Transmission and the Physical Uplink Shared Channel

PUSCH in 5G NR is responsible for the transmission of user data and uplink control information from UE to the base station (BS). As depicted in Figure 7, the PUSCH processing chain includes various modules that handle tasks such as modulation, coding, and multiple antenna processing, ensuring robust data transmission over wireless fading channels.

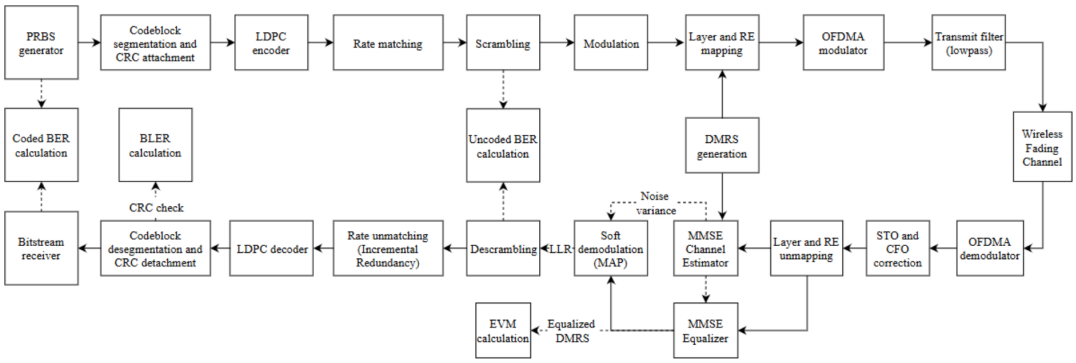


Figure 7: Diagram of the 5G NR PUSCH processing chain, illustrating the sequence of operations from the PRBS generation to the bitstream reception [21]. The chain includes key processes such as LDPC encoding, rate matching, modulation, and OFDMA modulation, followed by transmission through a wireless fading channel. The receiver side encompasses OFDMA demodulation, synchronization, equalization, and error correction, culminating in the calculation of BLER and BER metrics for performance evaluation.

The user data conveyed over the PUSCH refers to the actual content generated or requested by the end user. This may include, for example, Internet Protocol (IP) packets that encapsulate data from applications such as web browsing, video conferencing, email, or file uploads. For instance, when a user uploads a photo to a social media platform, the image data is packetized into IP packets, which are then transmitted over the air interface using the PUSCH.

In our simplified model of the full PUSCH chain, the PUSCH transmitter begins with a Pseudo-Random Binary Sequence (PRBS) generator, which generates test data that simulates the payload of real user data for system verification or simulation. The data undergoes codeblock segmentation with Cyclic Redundancy Check (CRC) attachment, enabling error detection at the receiver. Subsequently, Low-Density Parity-Check (LDPC) coding is applied for forward error correction, followed by rate matching to adapt the code rate to channel conditions and resource allocation. The encoded bits are scrambled to randomize interference and avoid spectral peaks. The scrambled bits are mapped to complex modulation symbols, such as Quadrature Phase Shift Keying (QPSK) through modulation. These symbols are distributed across layers which are spatial streams for Multiple-Input Multiple-Output (MIMO) and REs in the time-frequency grid. The OFDMA modulator transforms the symbols into time-domain signals using IFFT, while a lowpass transmit filter shapes the waveform to comply with spectral mask requirements.

The transmitted signal propagates through a wireless fading channel characterized by multipath propagation with noise. These impairments distort the signal amplitude and phase, necessitating robust synchronization and equalization at the receiver.

The received signal is processed by an OFDMA demodulator, which converts the time-domain waveform into frequency-domain symbols using FFT. Symbol Timing Offset (STO) and Carrier Frequency Offset (CFO) correction are used to align the time and frequency domains of the received signal. The symbols in the time-frequency grid are extracted and separated in layer and RE demapping. A Minimum Mean Square Error (MMSE) channel estimator computes the channel estimate, enabling the MMSE equalizer to mitigate ISI and recover the transmitted symbols. Soft demodulation using Maximum A Posteriori (MAP) detection computes Log-Likelihood Ratios (LLRs) for each bit, which are descrambled to reverse the transmitter's scrambling operation. The LLRs are fed to the LDPC decoder for error correction, followed by rate unmatching to reassemble codeblocks. A CRC check validates the integrity of the decoded data. Finally, BLock Error Rate (BLER) and Bit Error Rate (BER) metrics are calculated to evaluate link performance.

The PUSCH architecture delivers high spectral efficiency, low latency, and ultra-reliable performance. The equalizer and channel estimator in PUSCH optimize signal recovery in multipath environments, minimizing ISI while adapting to time-varying channel states. Flexible modulation schemes such as N-QAM, and MIMO layer mapping enable dynamic adaptation to varying throughput and Signal-to-Noise Ratio (SNR) demands. Additionally, BER and BLER metrics provide comprehensive diagnostics for evaluating modulation fidelity and link robustness.

2.6 Demodulation Reference Signal

The Demodulation Reference Signal (DMRS) primarily used for coherent demodulation of uplink transmissions. It facilitates accurate channel estimation, enabling the next-generation Node B (gNB) receiver to compensate for channel impairments such as fading and phase noise. Unlike downlink reference signals, the uplink DMRS is UE-specific and transmitted alongside uplink data on the PUSCH.

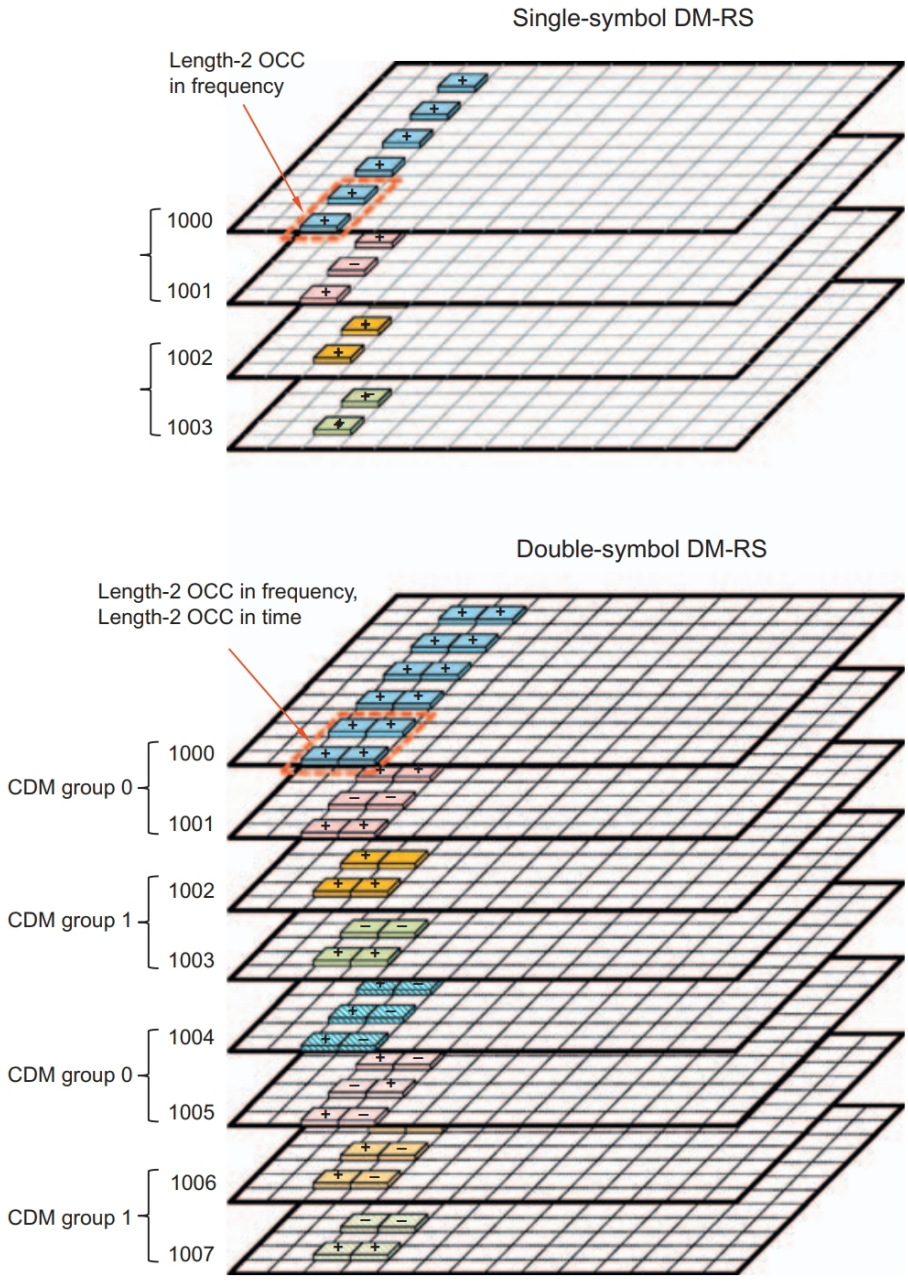


Figure 8: Illustration of Type 1 DMRS configuration, showing single-symbol and double-symbol DMRS arrangements with OCC applied in frequency and time domains [22]. The single-symbol DMRS utilizes length-2 OCC in frequency for code-division multiplexing across antenna ports, while the double-symbol DMRS applies OCC in both frequency and time, enhancing robustness against fading and Doppler effects.

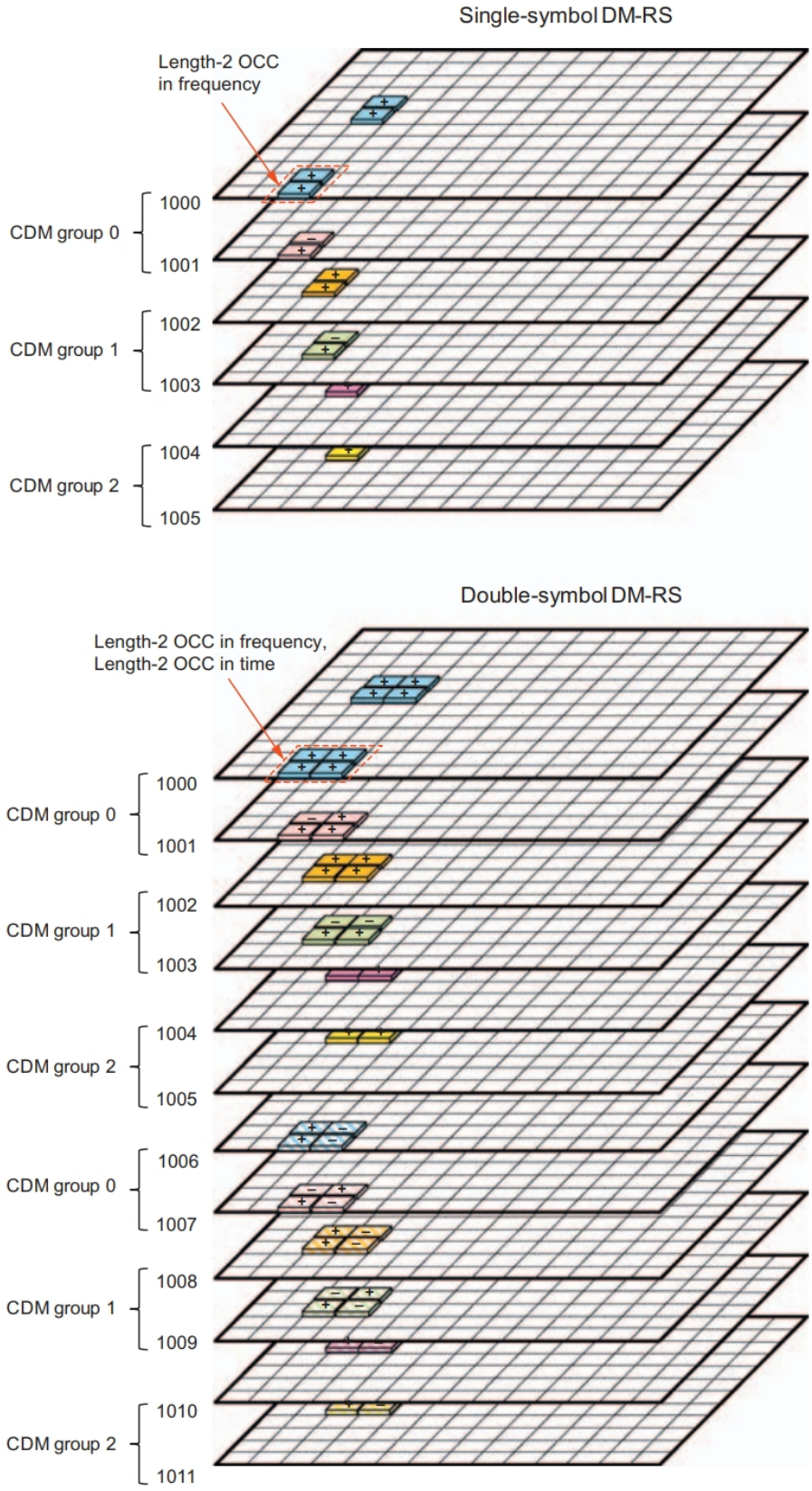


Figure 9: Depiction of Type 2 DMRS configuration, highlighting the use of single-symbol and double-symbol DMRS with frequency and time domain OCCs [22]. The single-symbol DMRS is designed for low overhead, suitable for stable channels, whereas the double-symbol DMRS offers improved performance in high-mobility scenarios by spanning two OFDM symbols.

The DMRS in uplink can be configured flexibly in terms of symbol duration, frequency domain spreading, and code-division multiplexing. Two typical configurations are single-symbol DMRS and double-symbol DMRS, each offering different trade-offs between overhead and channel estimation performance. The detailed structures of these configurations are depicted in the Figure 8 and 9, respectively.

In the single-symbol DMRS configuration (top part of Figure 8 and 9), the DMRS is transmitted in a single OFDM symbol. It utilizes orthogonal cover codes (OCCs) in the frequency domain to achieve code-division multiplexing across different antenna ports or layers as well. As shown, Code Division Multiplexing (CDM) groups, for example, group 0, group 1, and group 2 in the Figure 9 are mapped to specific REs within one symbol in each slot. A length-2 OCC is applied in frequency, spreading the reference signal across two subcarriers per group to ensure mutual orthogonality. This arrangement supports efficient multiplexing while minimizing inter-layer interference. The DMRS typically occupies one OFDM symbol out of a total of fourteen within a time slot when using a normal CP. This corresponds to approximately 7.14% of the OFDM symbols in a slot being allocated to reference signaling. Such a configuration introduces minimal overhead, making it particularly suitable for scenarios characterized by low-to-moderate user mobility and relatively stable wireless channels.

In contrast to the single-symbol case, the DMRS can also span two consecutive OFDM symbols. It applies OCC in both frequency and time domains in the double-symbol DMRS configuration (bottom part of each figure). Each CDM group in this configuration is mapped to specific sets of REs in both the frequency and time domains, providing enhanced robustness to fading and Doppler effects. This makes the double-symbol DMRS particularly suitable for high-mobility scenarios and cases.

2.7 Propagation Channels in 5G

Wireless communication systems rely on propagation channels to describe how radio signals travel from a transmitter to a receiver. In standardized system-level simulations, the 3rd Generation Partnership Project (3GPP) specifies two principal channel models capable of supporting wireless communication simulations: the Tapped Delay Line (TDL) and the Clustered Delay Line (CDL) models [23].

The TDL model characterizes the channel using a fixed number of delay taps, each associated with a power level. It is primarily intended for Single-Input Single-Output (SISO) scenarios. Conversely, the CDL model is designed to accommodate multiple antenna configurations in single-user contexts, making it particularly suitable for MIMO systems. Given that our experimental setup involves a single User Equipment (UE) having multiple antennas, the CDL model is adopted to capture the spatial characteristics intrinsic to such systems.

The CDL model [23] supports a broad frequency range, from 0.5 GHz to 100 GHz, and a maximum bandwidth of 2 GHz. It is especially well-suited for single-user MIMO scenarios. It represents the wireless channel as a set of multipath clusters which is shown in Figure 10. Each cluster is characterized by parameters such as path delays, Angle-of-Arrival (AoA), Angle-of-Departure (AoD), Doppler shifts, and relative power levels.

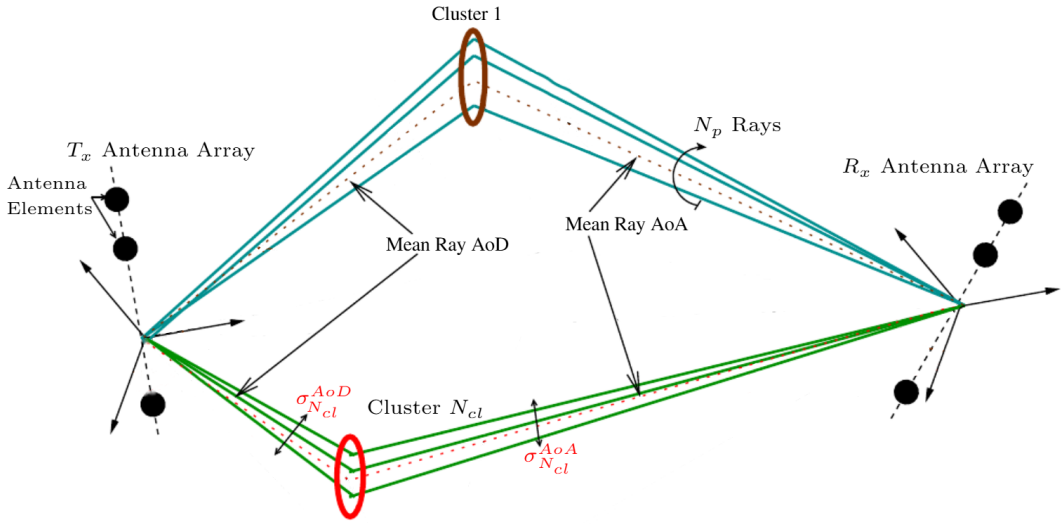


Figure 10: Spatial representation of N_{cl} clustered channel between the transmitter and receiver antenna arrays [24]. This figure illustrates the CDL model used for capturing spatial characteristics in single-user MIMO systems. Each cluster is characterized by path delays, AoA, AoD, Doppler shifts, and relative power levels, providing a comprehensive depiction of multipath propagation.

To capture diverse propagation environments, 3GPP defines five standardized CDL profiles. CDL A, CDL B, and CDL C are designed for Non-Line-Of-Sight (NLOS) scenarios, while CDL D and CDL E are tailored for Line-Of-Sight (LOS) conditions [23]. Moreover, the model allows for both delay and angular scaling to match target Root-Mean-Square (RMS) delay spreads and angular spreads, providing flexibility to adapt to different deployment environments. In LOS conditions, the Rician K-factor can be adjusted to reflect varying levels of dominant signal strength.

Channel coefficient generation within the CDL framework follows a structured methodology. The process begins with the generation of angular parameters (AoA, AoD) for each ray within a cluster which is shown in Figure 10, based on predefined angular spreads and offset distributions. These angles are coupled within each cluster to reflect spatial correlation. Each ray is also assigned a cross-polarization power ratio according to model-specific distributions. The final channel coefficients are computed following a standardized procedure that treats all clusters equally, without employing sub-clustering in the delay domain.

In addition to the propagation channel CDL model, Additive White Gaussian Noise (AWGN) is incorporated to simulate thermal noise at the receiver. AWGN is modeled as a zero-mean Gaussian random process with constant power spectral density, representing the ambient noise present in all practical wireless communication systems.

2.8 Performance Metrics and Evaluation in 5G

BER and BLER are two of widely used evaluation metrics in 5G. BER is defined as the ratio of the number of erroneous bits to the total number of transmitted bits. In contrast, BLER is the ratio of the number of erroneous blocks to the total number of transmitted blocks. Each block typically consists of multiple bits. BLER is measured after applying block-level error correction techniques which use LLR.

An example of such an error correction technique is the LDPC code, which is used to correct errors at the block level. Before the transmission, signals carrying data are divided into blocks, and each block is encoded using an LDPC encoder. After reception, LLR is computed for each bit within the received block. LLR provides likelihood information about each bit and serves as input to the LDPC decoder, which iteratively processes them to correct errors. The decoder uses the parity-check matrix of the LDPC code to identify and correct errors in the block, using the LLR values to make informed decisions regarding the most likely transmitted bits. The output of the LDPC decoder is a corrected block of bits, ideally with all errors removed. This corrected block is then used to compute the BLER.

A lower BLER indicates better performance of the error correction scheme, as fewer blocks are received with errors. BER, on the other hand, is the ratio of the number of erroneous bits to the total number of transmitted bits. It measures the quality of the transmission channel at the bit level. BER evaluates the raw error rate of individual bits, before any error correction is applied. A lower BER indicates a cleaner transmission channel with fewer bit errors, but it does not account for the improvements provided by block-level error correction techniques.

In the visualization of evaluation, BER and BLER tend to act as dependent variables that reflect the error performance of a receiver algorithmn, while the energy per bit to noise power spectral density ratio (E_b/N_0) and SNR serve as independent variables reflecting the simulated channel conditions.

The ratio E_b/N_0 provides a normalized measure of signal energy relative to the noise level. A higher E_b/N_0 shows that the energy per bit is high relative to the noise level, which generally implies better channel conditions and potentially lower BER and BLER values. While SNR characterizes the overall channel quality at the signal level, its relationship to E_b/N_0 depends on the modulation order and bit rate, given by

$$\frac{E_b}{N_0} = \frac{\text{SNR} \cdot B}{R_b}, \tag{1}$$

with B denoting the system bandwidth and R_b referring to the bit rate. This relationship allows SNR-based evaluations to be directly mapped to E_b/N_0, and subsequently to BER or BLER performance. The evaluation visualizations are typically presented as BER or BLER curves versus E_b/N_0 or SNR.

3 Neural Network Background

Neural networks form a class of AI algorithms designed to emulate the intricate operations of the human brain. One common use case is to be powerful function approximators to simulate the relationship between features and labels. It makes a machine learn from data through a training stage which include validation, and evaluates the performance of this machine through testing stage. The training process adjusts the neural network's internal parameters, such as weights and biases, to make the predictions approximate the target labels. Conversely, the testing process evaluates the neural network's predictive ability on data not encountered during training.

In contrast to the neural network's internal parameters, hyperparameters define the configuration of neural network training. One of the hyperparameters is the number of epochs, which shows the number of iterative cycles used in training. During each epoch, all samples in the training are propagated forward through the neural network layers to produce predictions. These predictions are compared with the corresponding target labels and then a loss function is used to show the quantified distance between them. The parameters of the neural network are updated by minimizing this loss with the optimization algorithms which tend to be a variant of stochastic gradient descent. Gradients are computed by the backpropagation algorithm via the chain rule, which facilitate the end to end learning process.

Validation is used for hyperparameter tuning and early detection of overfitting or underfitting behaviors. However, since the signals in this thesis are randoAIy simulated, and the quantity of data is thus unbounded, the likelihood of overfitting is reduced.

After training, the neural network is evaluated with the testing dataset. This phase assesses the neural network's generalization performance on unseen data. BER and BLER serve as the evaluation metrics to estimate the neural network performance in this thesis.

3.1 Artificial Neuron and Activation Functions

The artificial neuron is also called node or unit, and it constitutes the fundamental computational building block of a neural network. The artificial neuron was first introduced in [25], where it was illustrated as a mathematical representation of a biological neuron to provide a binary threshold-based mechanism for simulating basic logical operations. Later, [26] enhanced it by incorporating adjustable weights and a learning algorithm, which makes many future advancements possible.

Mathematically, the artificial neuron operates by receiving multiple features, which are then associated with specific weights and aggregated through a weighted sum. A bias is typically added to it for the node to represent a broader class of functions. This linear combination can be transformed by a non-linear activation function to produce the final output of the artificial neuron. Formally, this operation can be described as

$$y = \phi\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2}$$

,where x_i denotes the i^{th} input feature, w_i denotes the corresponding weight, b is bias, $\phi(\cdot)$ is the activation function, and y is the output of the artificial neuron.

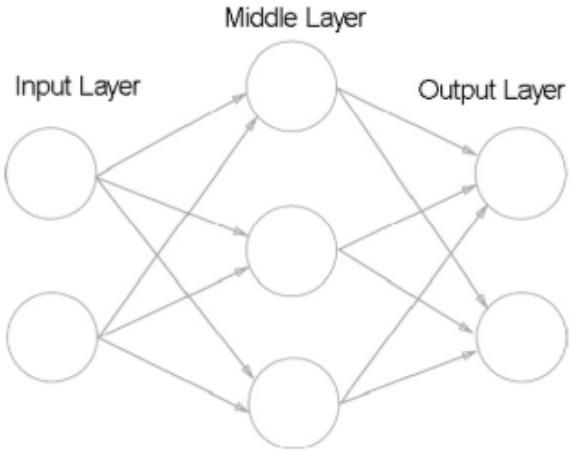


Figure 11: Neural network architecture demonstrating the modular arrangement of artificial neurons into input, hidden, and output layers [27].

The design of the artificial neuron permits modular structure. Multiple nodes can be organized into layers to form complex neural network structures. As illustrated in Figure 11, nodes are arranged sequentially into input, middle (hidden), and output layers, allowing for hierarchical feature learning. Each artificial neuron in a hidden layer processes the outputs from the previous layer, enabling the neural network to learn increasingly abstract representations of the features.

Activation functions are responsible for introducing non-linearity into the neural network. They enable the neural networks to learn non-linear patterns within data. Without activation functions, a neural network composed of nodes in multiple layers is equivalent to a linear neural network with only one layer, lacking the representational capacity needed for complex learning tasks [28]. Universal Approximation Theorem [29] supports it, which states that a feedforward neural network with at least one hidden layer and non-linear activation functions can approximate any continuous function using a sufficient number of neurons. This theoretical result confirms that neural networks, when equipped with non-linear activation, have the potential to represent highly complex relationships between inputs and outputs.

Historically, the step function was one of the earliest activation functions. However, due to its non-differentiability, it is unsuitable for modern learning algorithms that rely on gradient-based optimization. Then, the sigmoid function [30], which is defined by

$$\sigma(z) = 1/(1 + e^{-z}), \tag{3}$$

became a widely adopted alternative [31]. However, although the sigmoid function is differentiable, it suffer the non zero centered problem and is prone to the vanishing gradient problem, where the derivatives of the activation functions become progressively smaller during backpropagation [32]. The hyperbolic tangent (Tanh) function [30] was introduced to offer a zero-centered output in the range $(-1, 1)$, but

still exhibits vanishing gradients under certain conditions. Then, the rectified linear unit (ReLU) [30], defined as

$$\mathrm{ReLU}(z) = \max(0, z) \tag{4}$$

was introduced. It mitigates the vanishing gradient problem for positive inputs, making it particularly effective for deep networks. Nonetheless, ReLU can cause some artificial neurons to become inactive, leading to the "dying ReLU" problem. To address this, various modifications such as Leaky ReLU [33], Parametric ReLU (PReLU) [34], and Exponential Linear Units (ELU) [35] have been proposed. These variants aim to maintain non-zero gradients for negative inputs, thereby ensuring more robust training dynamics.

3.2 Loss Function and Optimization Function

The loss function guides the learning process by providing an error that needs to be minimized by the neural network, which is achieved through updating the neural network's parameters. Common loss functions vary depending on the tasks.

For regression problems, two loss functions are widely used. One is the Mean Squared Error (MSE) defined as

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{5}$$

and the other is the Mean Absolute Error (MAE), defined as

$$\mathcal{L}_{\mathrm{MAE}} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|. \tag{6}$$

In 5 and 6, y_i and $\hat{y}_i$ represent the true label and the predicted value, respectively, while N is the number of samples in the dataset.

MSE corresponds to the ℓ_2 norm, penalizing larger errors more heavily due to the squaring operation, making it sensitive to outliers. It is commonly used when large errors should be strongly discouraged. In contrast, MAE is defined using the ℓ_1 norm. It treats all errors equally, regardless of their magnitude, making it more robust to outliers but potentially slower to converge during training. The choice between MSE and MAE often depends on the specific characteristics of the task. For smoother predictions and when large errors are especially undesirable, MSE is typically preferred.

For classification tasks, the Cross-Entropy Loss is often employed, especially when the predictions are probabilities after a Softmax activation. It is given as

$$\mathcal{L}_{\mathrm{CE}} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{7}$$

where C is the number of classes.

The most common selection of optimization techniques is based on gradient descent. One of them is Stochastic Gradient Descent (SGD) [36] [37] which updates parameters according to the gradient of the loss function with respect to each parameter. Given a hyperparameter learning rate η, the parameter update rule is illustrated in Equation 8.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}, \tag{8}$$

where θ denotes the parameters and $\nabla_\theta \mathcal{L}$ is the gradient. Variants of SGD, such as Momentum, RMSprop, and Adam [38], have been developed to improve convergence speed and stability. Adam, in particular, combines the advantages of adaptive learning rates and momentum, making it highly effective and popular in many neural network applications.

3.3 Basic Neural Network Architectures

3.3.1 Convolutional Neural Network

The CNN is another foundational neural network. It is designed specifically for processing data with a grid-like topology, such as images. Figure 12 presents an example of the CNN architecture.

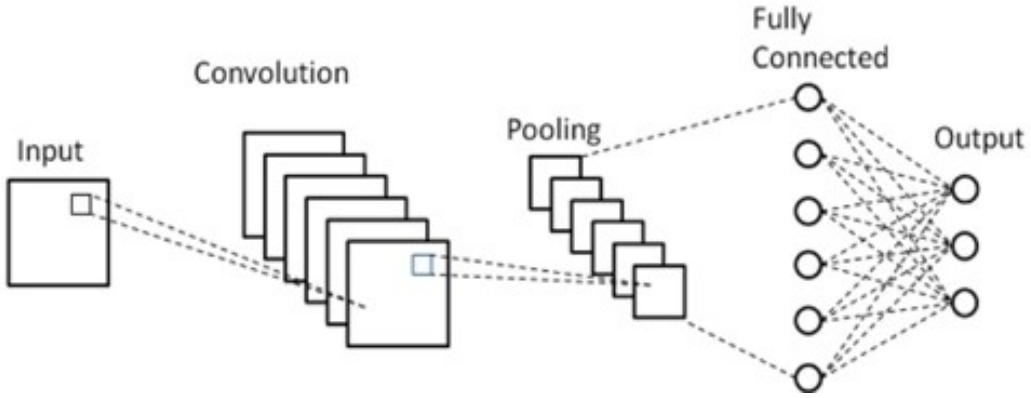


Figure 12: General architecture of a CNN, consisting of an input layer, convolutional and pooling layers for feature extraction, and fully connected layers for classification [39].

Images are used as data to illustrate the architecture of CNN here. Convolution is applied to create feature maps by performing convolution operations via filters. Pooling can be used after convolution to reduce dimensions of feature maps while preserving the important information. Fully connected tend to be applied at the end of CNN to transform the flattened feature maps into outputs, where the number of outputs corresponds to the number of labels. The features to the CNN is typically a multidimensional tensor, as shown in Figure 13. Its dimension is (C, H, W), where C, H and W denote the number of channels, height and width.

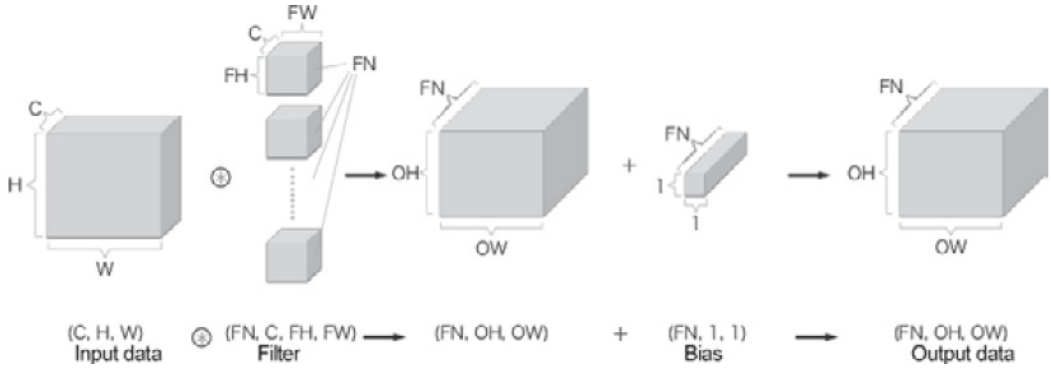


Figure 13: Illustration of the convolution operation in a CNN [27]. Input data with dimensions is processed using a set of filters , producing output feature maps of size . A bias term is added before applying non-linear activation functions.

A set of learnable filters, which are also called kernels, are applied to the features with the shape (C, FH, FW) each, where FH and FW are the filter height and width, respectively. Thus, the full set of filters is denoted with the shape (FN, C, FH, FW), where FN is the number of filters applied. These filters are convolved with the features using the convolution operator, producing an intermediate tensor with the shape (FN, OH, OW), where OH and OW represent the height and width determined by the convolutional stride and padding parameters.

Following the convolution operation, a bias with the shape $(FN, 1, 1)$ is added via broadcasting to help the neural network improve the representational capacity. The output of this process is a tensor with the shape (FN, OH, OW), constituting the transformed feature maps. Following the convolutional layer, a non-linear activation function, which tend to be ReLU, is applied to introduce non-linearity into the neural network.

3.3.2 Residual neural Network

Neural networks encounter challenges in optimization, especially the vanishing gradient problem when the depth of neural network is quite large. In this problem, increasing the number of layers beyond a certain point leads to a decline in training accuracy. It contradicts the expected behavior that deeper networks should perform at least as well as shallower ones. ResNet is a neural network introduced in [40] to solve this problem by incorporating residual learning into the network architecture.

Instead of attempting to learn an underlying mapping directly, a ResNet block learns the residual function with respect to the input [40]. Figure 14 illustrates the structure of a typical residual block, in which x denote the input to this block. Rather than directly learning a mapping $H(x)$, the residual block aims to learn a residual function $F(x) = H(x) - x$, thereby reformulating the mapping as

$$H(x) = F(x) + x. \tag{9}$$

This formulation is implemented by introducing an identity connection that bypasses

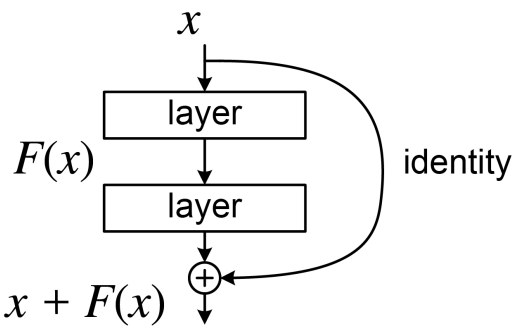


Figure 14: Structure of a Residual Block in ResNet [40]. This block demonstrates the concept of residual learning, where the input (x) is directly added to the output of the residual function ($F(x)$) through an identity connection. This design facilitates improved gradient flow during backpropagation, addressing the vanishing gradient problem in deep networks.

one or more layers to enable the input x to be added directly to the output of the residual function $F(x)$.

These residual connections could improve the gradient flow during backpropagation, since the identity connections enable gradients to propagate more directly through the network. Thus, it mitigates the vanishing gradient problem that typically hampers the training of very deep architectures. This enhancement allows networks with residual blocks to converge faster and more reliably than their plain counterparts.

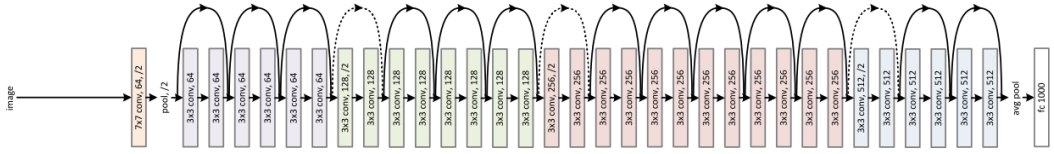


Figure 15: Architecture of a ResNet Model [40]. The diagram illustrates a sequence of residual blocks, each comprising convolutional layers, batch normalization, and activation functions. The identity connections in each block enable the direct addition of the input to the output, enhancing training efficiency and convergence.

The architecture of a ResNet is composed of multiple residual blocks, as shown in Figure 15. They are stacked sequentially to form a deep computational graph. Each residual block typically consists of two or more convolutional layers, followed by batch normalization and non-linear activation functions. The key characteristic of the ResNet's architecture is the identity connection in each block. This architectural design allows for the construction of neural networks with greater depth than previous neural networks, enabling the extraction of complex features with improved performance.

4 Methodology

4.1 Related Work

Over the past few years, AI techniques have emerged as powerful tools to enhance or replace the classical channel estimation methods such as Least Squares (LS) and MMSE. Although these traditional approaches are analytically tractable, they often struggle to adapt to the complex and non-linear conditions in wireless channels in 5G. Neural networks, particularly convolutional and recurrent architectures, have shown significant promise in capturing both spatial and temporal dependencies in channel conditions, particularly under pilot scarce or high mobility scenarios [41].

The CNN is effective for channel estimation due to its ability to exploit local spatial correlations in time-frequency grids. In [42], a CNN-based approach to estimate cascaded 5G channels was introduced, showing that CNN outperforms classical estimators in dense multipath scenarios. Similarly, [43] proposed an enhanced CNN framework for MIMO–OFDM systems with polar-coded transmission over realistic 5G channels, achieving notable accuracy improvements.

To further improve depth and representational power, ResNet has been incorporated in recent works. Particularly, [44] demonstrated the effectiveness of 2D/3D ResNet architectures for channel estimation in 4G and 5G NR systems. Moreover, [45] advanced this direction by combining ResNet and U-Net in a two-stage estimation scheme, optimized for mmWave massive MIMO settings.

While CNN and ResNet architectures excel at spatial interpolation, they do not inherently account for temporal correlation. Long Short-Term Memory (LSTM) networks are designed precisely for sequential dependencies. For example, [46] showed that a bi-directional LSTM can denoise time-varying TDL-C channels in massive-MIMO OFDM and outperform static estimators. Beyond the physical layer, residual LSTM neural networks have improved mobility management by forecasting future user positions [47] and enabled intelligent traffic steering in Open RAN [48].

Other innovations have been proposed to enhance channel estimation, which include SRGAN-based estimation [49] and compressed sensing approaches [50], revealing a broad spectrum of DL architectures applicable to the channel estimation problem. Moreover, [51] proposed Channelformer, a Transformer-based neural network employing self-attention to capture global dependencies in the channel matrix, and achieving real-time adaptability via online training. While Transformers show promise, they often come with high computational overhead, making CNN and ResNet hybrids more suitable for low-latency applications.

Compared to the most commonly used AI-based methods in channel estimation, which focus on minimizing the MSE between predicted and ideal channel conditions, several recent studies have emphasized throughput as a critical optimization objective in 5G systems. The throughput-oriented design seeks not just to reconstruct channel conditions accurately, but also enhances end-to-end communication performance. For example, [52] employed AI techniques to predict LTE and 5G network throughput, aiding in dynamic resource allocation and improving QoS provisioning. Similarly, [53] has proposed a data-driven framework for downlink throughput prediction across

4G-LTE and 5G networks, demonstrating improved accuracy through feature-rich modeling of network states. More granular approaches, such as the LSTM-based method in [54], modeled user traffic patterns to anticipate future data rates, enabling proactive scheduling and traffic steering. These works illustrate a growing trend toward aligning AI-driven neural networks with system-level metrics, moving beyond traditional estimation errors to optimize the metrics that matter most in deployment, particularly throughput.

The work in this thesis takes a novel approach by embedding throughput directly into the training objective of a CNN-based ResNet neural network. Rather than using traditional loss functions between predicted and ideal channel conditions, soft demapper's outputs LLRs are incorporated into the throughput optimized loss function. While channel smoothing has not been trained before with LLR-based loss functions, this loss function has been succesfully explored by DeepRx [55] and `Sionna` tutorials [56]. This loss function allows the neural network to iteratively refine its channel estimates based on the throughput, which could improve performance in the system level and enable training to be done on data collected from real BSs.

4.2 Signal Transmission and Reception

4.2.1 Signal Transmission

As illustrated in Figure 16, the payload is generated by UE itself and transmitted on uplink channels, such as PUSCH. The binary payload can be represented as a bitstream

$$\mathbf{b} = [b_1, b_2, \ldots, b_n] \tag{10}$$

which can be modulated, for example, using 16-QAM in modulation block, where each complex-valued symbol represents 4 bits,

$$\mathbf{X} = [X_1, X_2, \ldots, X_k], \quad \text{where } k = \frac{n}{4}. \tag{11}$$

Each symbol X_i is drawn from the 16-QAM constellation. For a single transmit antenna ($N_{\text{tx}} = 1$), the modulated symbols are directly mapped to one transmission layer. These symbols are placed into a time-frequency RG $\in \mathbb{C}^{N_{\text{tx}} \times N_{\text{layers}} \times N_{\text{symb}} \times N_{\text{subcarriers}}}$.

In this thesis, 14 OFDM symbols per slot and 96 subcarriers per OFDM symbol are configured. A pilot signal DMRS is inserted into the third OFDM symbol across all subcarriers to enable channel estimation. To simulate the characteristics of 5G pilot features, a pilot mask $\mathbf{M}$ is applied to DMRS, which is illustrated in detial in subsection 5.2.1. This mask ensures that the pilot values alternate across subcarriers, enhancing the signal's ability to carry more data, which is a key characteristic of 5G technology. The pilot values are defined as

$$X_p[i] = \begin{cases} \frac{1+j}{\sqrt{2}}, & \text{for even } i \\ 0, & \text{for odd } i, \end{cases} \tag{12}$$

where $j = \sqrt{-1}$.

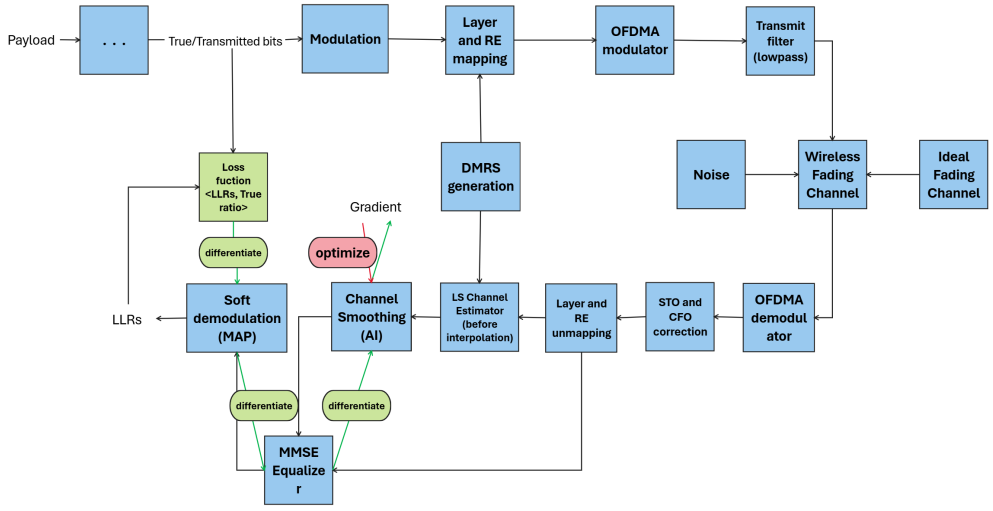


Figure 16: Diagram of the signal processing chain for uplink transmission in the thesis, illustrating the generation and modulation of the payload, layer and resource element mapping, and modulation. The process includes DMRS generation for channel estimation, followed by transmission through a wireless fading channel with noise. The received signal undergoes demodulation, channel smoothing, and equalization, with feedback loops for optimization and soft demodulation.

The RG undergoes OFDMA modulation via an IFFT, yielding the time-domain signal

$$x(t) = \sum_{f} X e^{j2\pi ft}. \tag{13}$$

Thus, the received signal $y(t)$ is aligned when the transmitted signal $x(t)$ is filtered and transmitted over a wireless channel modeled using the CDL and AWGN as

$$y(t) = \sum_{i=1}^{N} h_i(t) \cdot x(t - \tau_i) + n(t). \tag{14}$$

where $h_i(t)$ and τ_i denote the gain and delay of the i-th path, and $n(t)$ refer to the noise.

4.2.2 5G Receiver Architecture

At the receiver stage, it first synchronizes using STO and CFO correction. The time-domain signal is then transformed back into the frequency domain using the FFT as

$$Y = \int_{-\infty}^{\infty} y(t) e^{-j2\pi ft}\, dt \tag{15}$$

Channel estimation which includes raw channel estimation and channel smoothing is based on

$$Y = H \cdot X + N, \tag{16}$$

where H refer to ideal fading channel conditions, X is transmitted signals, N is Gaussian noise following $\mathcal{N}(0, \sigma^2)$, and Y is the received signal in frequency domain. LS method is used for raw channel estimation $\hat{H}_{\text{LS}}$ via minimizing the LS cost function

$$J(\hat{H}_{\text{LS}}) = \|Y_p - \hat{H}_{\text{LS}} X_p\|^2. \tag{17}$$

Through setting the cost function's derivative equal to zero, $\hat{H}_{\text{LS}}$ is obtained with pilot transmitted symbol X_p and pilot received symbol Y_p, given as

$$\hat{H}_{\text{LS}} = X_p^{-1} Y_p. \tag{18}$$

This raw estimate $\hat{H}_{\text{LS}}$ is noisy and discrete in time and frequency domain. In channel smoothing, a neural network is applied to smooth it, given as

$$\hat{H} = f(W\hat{H}_{\text{LS}} + B) \tag{19}$$

where W and B are learnable parameters and f is the activation function. The channel estimate $\hat{H}$ is then used in equalization. MMSE equalization is applied to estimate the transmitted signal carrying data X_d from the received signal carrying data Y_d according to the equation:

$$\hat{X}_d = G_{\text{MMSE}} Y_d \tag{20}$$

where G_{MMSE} can be derived from minimizing the MSE $\|G_{\text{MMSE}} Y_d - X_d\|^2$ via setting its derivative with repect to G_{MMSE} to zero. Thus, G_{MMSE} can be found to be

$$G_{\text{MMSE}} = \hat{H}^H \left(\hat{H}\hat{H}^H + \frac{\sigma^2 I}{P} \right)^{-1}. \tag{21}$$

After setting 16-QAM modulation under Gaussian noise $N(0, \sigma^2)$ in this thesis, the LLR is computed by MAP estimation as

$$LLR(b_m) = \log \left(\frac{P(b_m = 1 \mid \hat{X}_d)}{P(b_m = 0 \mid \hat{X}_d)} \right). \tag{22}$$

This provides LLR for the loss function in the thesis to update the parameters W and B in channel smoothing. This loss function is defined as Binary Cross-Entropy (BCE) fuction between LLR which is $\mathbb{R}^N$ and true coded bits $\mathbf{b} \in \{0, 1\}^N$ as follows.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{m=1}^{N} [b_m \log \sigma(LLR(b_m)) + (1 - b_m) \log(1 - \sigma(LLR(b_m)))] \tag{23}$$

where $\sigma(\cdot)$ is the sigmoid function that maps LLRs to bit probabilities.

The BCE loss function is differentiable, allowing the gradients to be computed with respect to the neural network parameters W and B in the channel smoothing neural network. The gradients are computed using the chain rule as follows:

$$\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial W} = \frac{\partial \mathcal{L}_{\text{BCE}}}{\partial LLR} \cdot \frac{\partial LLR}{\partial \hat{X}_d} \cdot \frac{\partial \hat{X}_d}{\partial \hat{H}} \cdot \frac{\partial \hat{H}}{\partial W}; \tag{24}$$

$$\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial B} = \frac{\partial \mathcal{L}_{\text{BCE}}}{\partial LLR} \cdot \frac{\partial LLR}{\partial \hat{X}_d} \cdot \frac{\partial \hat{X}_d}{\partial \hat{H}} \cdot \frac{\partial \hat{H}}{\partial B}. \tag{25}$$

4.3 Key Architectures

In order to comply with 5G standards, the thesis is developed using `Sionna` simulator [56], which is an open-source library for research on communication systems developed by NVIDIA, and with `TensorFlow` for building neural network. `Sionna` simulator provides a modular framework for simulating various components of the 5G NR physical layer, in alignment with the 3GPP specifications. Key features used in the thesis are supported by `Sionna` include stream management, OFDM RG generation, antenna array modeling and channel modeling.

4.3.1 Proposed Neural Network Architecture

As illustrated in Figure 17, the proposed neural network combines CNN and ResNet to interpolate and denoise raw channel estimate obtained using LS. This neural network

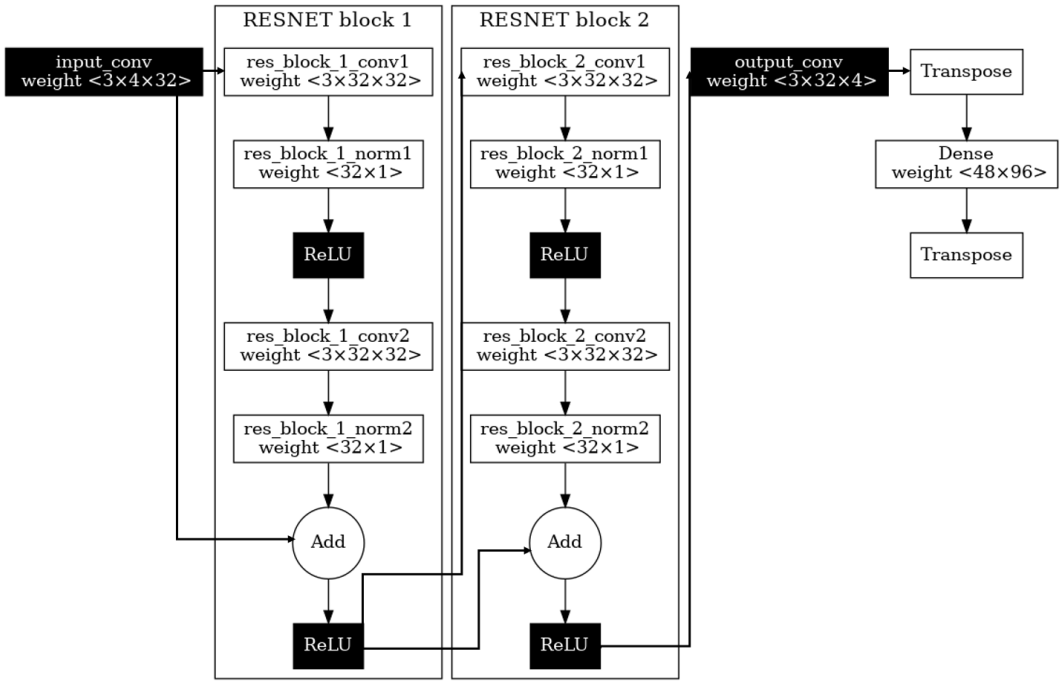


Figure 17: Proposed neural network architecture combining CNN and ResNet for channel interpolation and denoising. The network processes a tensor input of LS channel estimates through convolutional layers and residual blocks, enhancing feature extraction and preserving input information. The architecture includes a final dense layer for subcarrier interpolation, outputting a refined complex-valued channel estimate.

is applied for channel smoothing to improve the raw channel estimate.

The input to the neural network is a tensor $\mathcal{H}_{\text{LS}} \in R^{B\times T\times \frac{F}{2}\times 2L}$, where B denotes the batch size, T corresponds to the time, $\frac{F}{2}$ refers to the number of subcarriers with values in a pilot symbol, and the final dimension distinguishes the real and imaginary components of the LS channel estimate across layers L. These components are concatenated along the channel axis to form a real-valued representation suitable for processing by convolution.

The input firstly goes through a convolutional layer with 32 filters and kernel size 3, which acts across the L layer axis to extract features. Its output is processed sequentially by two residual blocks, each of which contains convolutional layers with ReLU activations and skip connections to preserve input information. After residual processing, the output passes through another convolutional layer that return the layers to $2L$, while keeping the real and imaginary parts of the complex form of channel estimates separate. Then, the resulting tensor is transposed and passed through a FNN that performs subcarrier interpolation, transforming $\frac{F}{2}$ into F which represents the number of subcarriers.

Finally, interpolation is applied by repeating the channel estimate from this neural network based on the observed pilot signals. The estimate $\hat{\mathcal{H}}_{\text{AI}} \in R^{B\times T\times F\times 2L}$ is the output of this neural network, which is then reshaped and split into real and imaginary components $\Re\{\hat{\mathcal{H}}_{\text{AI}}\}$ and $\Im\{\hat{\mathcal{H}}_{\text{AI}}\}$, respectively. These are combined to form the complex-valued channel estimate as

$$\hat{\mathbf{H}} = \Re\{\hat{\mathcal{H}}_{\text{AI}}\} + j \cdot \Im\{\hat{\mathcal{H}}_{\text{AI}}\}. \tag{26}$$

The refined channel estimate $\hat{\mathbf{H}}$ is then reshaped and permuted to match the format required by the equalization process.

4.3.2 Backpropagation through classical components

Compared with the traditional training strategy that solely focuses on reconstructing channel conditions by minimizing the error between predicted channel conditions from the neural network and ideal ones, the key optimization strategy in this thesis is training the neural network based on end-to-end system-level metrics to maximize throughput. The neural network is trained by minimizing the loss between the LLRs and the original coded bits. Unlike traditional optimization strategy that relies on ideal channel conditions which are only achievable in simulation environments, the necessary elements for the throughput optimized strategy are readily available in real-life scenarios.

The loss function for the throughput optimized strategy is defined as the BCE $\mathcal{L}_{\text{BCE}}$ between the predicted LLR vector and the true coded bit vector. Thanks to the differentiability of the components following the channel smoothing, the gradients can propagate from soft output decoding back to the channel estimation. The gradients of this loss are computed with respect to the trainable parameters of the neural network, which are shown in (24) and (25). This strategy enables this CNN-based ResNet neural network to learn optimal channel conditions based on throughput rather than a specific distance-based error calculation about channel conditions.

4.4 Training and Testing Design

4.4.1 Training Setup

The training environment is configured to use a single GPU for efficient resource usage. The training pipeline is made deterministic by setting fixed random seeds.

As explained in the previous section, two neural networks, which share the same neural network architecture, but use different optimization strategies, are developed for this thesis. One is throughput optimized channel smoothing neural network based on predicted LLRs and true coded bits. The other one is optimized by minimizing the distance between channel estimate and ideal channel conditions. The latter is most common strategy [57]. Both neural networks are integrated into the 5G receiver algorithms provided by the `Sionna` library, leveraging `TensorFlow` as the backend.

For both neural networks, the input is the raw channel estimate from the LS channel estimator without interpolation, which will be illustrated in detail in section 5.2.2. Specifically, the raw channel estimate is preprocessed by removing pilot signals corresponding to masked indices. The complex-valued channel tensor is separated into its real and imaginary components, which are concatenated to form a real-valued input. This preprocessed raw channel estimate is then passed to the respective CNN-based ResNet neural network.

4.4.2 Throughput Optimized Neural Network Training

The first neural network is reffered to the throughput optimized neural network which is trained to maximize decoding throughput. In this throughput optimized neural network, the channel smoothing, equalizer and soft demodulation are integrated together, resulting in the generation of LLRs. The CNN-based ResNet neural network architecture in channel smoothing has been illustrated in detail in Section 4.3.1.

During training, binary information bits are generated and encoded using a 5G LDPC encoder. The encoded bits are modulated using 64-QAM and mapped to an OFDM RG. The modulated signals are transmitted through a randoAIy selected CDL fading channel model from CDL A, CDL B and CDL D, and noise is added according to the SNR specified by E_b/N_0. The raw channel estimate is calculated by LS and sent to this throughput optimized neural network to generate LLRs.

The loss function used is the bitwise BCE, computed between the predicted LLRs and the transmitted codeword bits. Gradients are backpropagated through the entire throughput optimized neural network, including the soft demodulation and equalizer, to update the CNN-based ResNet neural network's parameters. This enables the neural network to implicitly learn channel features that contribute most to correct decoding, thus enhancing throughput.

4.4.3 Ideal Channel Based Neural Network Training

The second neural network is the ideal channel based neural network which has similar CNN-based ResNet architecture, but does not include the equalizer or soft

demodulation modules. Its objective is to reconstruct the ideal frequency-domain channel conditions from the discrete and noisy raw channel estimate.

The training procedure is similar to that of the throughput optimized neural network, but its output is channel estimate which is sent to the equalizer followed by channel smoothing.

The loss function employed for this neural network is the MSE between the predicted and ideal channel conditions. Unlike the throughput optimized neural network, the optimization here is focused on improving the fidelity of channel estimation rather than the decoding performance.

4.4.4 Testing Procedure

The evaluation of the proposed throughput optimized neural network and ideal channel based neural network is conducted using the `Sionna` link-level simulator, employing an OFDM system under standardized 3GPP channel models CDL C and CDL E. The performance metrics used are the BLER and BER.

The test receiver is evaluated using four different configurations: *Ideal*, *LS Estimation*, *LLR-based AI CE*, and *Ideal Channel-based AI CE*. *Ideal* serves as the ideal benchmark. In this configuration, the receiver is provided with the ideal channel conditions, and it gives an upper bound for performance. *LS Estimation* is the baseline configuration. It employs a LS channel estimator without interpolation, representing a simple yet widely used technique. *LLR-based AI CE* corresponds to the throughput optimized neural network. It visualizes the neural network trained to maximize throughput using LLRs as outputs. *Ideal Channel-based AI CE* corresponds to the ideal channel-based neural network. It shows the neural network trained using ideal channel conditions, which is a popular approach in AI-based methods.

The simulations are performed across a wide range of E_b/N_0, specifically from $-5dB$ to $15dB$, with each point averaging results over up to 100 block errors or a maximum of 100 iterations. The batch size remains consistent across tests to ensure comparability. For each configuration, the BER and BLER curves are recorded and used to evaluate the robustness and decoding accuracy of the system under varying channel and noise conditions.

Furthermore, the generalization capability of the trained neural networks is analyzed using channel model CDL C and CDL E with multiple SNR points. This allows a comparative visualization of channel estimates after AI-based channel smoothing against ideal channel conditions and raw channel estimate, highlighting the performance advantage of AI-based channel smoothing in both high and low SNR scenarios.

5 Data and Experiments

5.1 Data

The simulation is based on a 1×2 system, with one transmitter antenna and two receiver antennas, representative of a simple 5G uplink configuration. This setup consists of a single UE equipped with one transmit antenna and a BS with two receive antennas. To reflect realistic transmission conditions, the input bit stream is randomly generated prior to modulation and transmission.

Within the OFDM RG, both signals carrying data and DMRS are mapped onto PRBs comprising 96 subcarriers. These reference signals serve as the basis for channel estimation at the receiver. Due to their sparse distribution in time and frequency, interpolation is required to estimate the channel conditions over the full grid.

To emulate realistic propagation environments, the simulator applies 3GPP CDL channel models to the transmitted signals. Each simulated CDL channel model also incorporates AWGN, where noise power is sampled from a Gaussian distribution to vary signal quality across runs.

At the receiver, the transmitted and received DMRS are extracted for raw channel estimation. The transmitted DMRS denotes the known reference signal sent by the UE, unaffected by channel or noise, while the received DMRS represents the corresponding observation after propagation, subject to distortion from multipath fading and AWGN.

To estimate the channel conditions, the LS is employed given as

$$\hat{H}_{\text{LS}} = X_p^{-1} Y_p, \tag{27}$$

This raw channel estimate $\hat{H}_{\text{LS}}$ serves as input to both neural networks. Additionally, H, which represents the ideal channel conditions, is recorded from the simulator after the channel propagation. It serves as the reference target in the loss function to optimize the ideal channel based neural network.

5.1.1 Scenario Parameters

Table 2 outlines the detailed configuration of the simulation environment. This simulation focuses on the PUSCH in the FR1 frequency band using time division duplexing (TDD) at a carrier frequency of 3.5 GHz. A total bandwidth of 100 MHz is allocated, and the subcarrier spacing is set to 30 kHz, resulting in 14 OFDM symbols per slot, consistent with 3GPP numerology.

One CDM group without data is used, and the DMRS is transmitted on antenna ports 0 and 2. The modulation is set to 256-QAM, and the Modulation and Coding Scheme (MCS) index is selected as 15 to maximize throughput and enhance transmission accuracy through high-order modulation. Sounding Reference Signals (SRS) and Phase Tracking Reference Signals (PTRS) are both disabled to focus on the core smoothing mechanism.

Two neural networks are trained using UE motion at a speed of 10 m/s, but evaluation is carried out across four different mobility scenarios to test generalizability. Scenario 1 involves walking at a minimum speed of 1.4 meters per second. Scenario 2

Parameter	**Values**
Data Source	`Sionna`
TX Antennas	1TX
Frequency band	FR1, TDD
Simulation length	1 slot
Physical channel	PUSCH
# of UEs	1
PRBs	8
PRB offset	8
RX Antennas	2RX
Carrier frequency	3.5 GHz
# of layers	1
Band Width	100 MHz
Subcarrier Spacing	30 kHz
DMRS type	1
# of OFDM Symbols per slot	14
# of DMRS Symbols/position	1/2
DMRS Length	1
DMRS index	[2]
# of CDM Groups w/o data (value)	1
DMRS port allocation	(0,2), pilot pattern
Modulation Table	256QAM
MCS Index / Modulation scheme	(15)
SRS	SRS off
PTRS	PTRS off
Delay spread (nano sec)	100
Time Offset	0 ns
Frequency Offset	0 Hz

Table 2: Simulation Parameters.

consists of biking at a minimum speed of 5 meters per second. Scenario 3 includes traveling by tram at a minimum speed of 8 meters per second. Scenario 4 involves metro travel at a minimum speed of 11 meters per second.

The primary goal of the evaluation scenarios is to check how well the throughput-oriented optimization strategies generalize to various channel dynamics introduced by different user velocities, compared with the traditional LS estimation method and the ideal channel conditions based neural network.

5.1.2 Dataset Parameters

Dataset	Parameters	Values
Training	Channel models	CDL A, CDL B, CDL D
	Velocity	36 km/h
	SNR	16 dB
Testing (Evaluation)	Channel models	CDL C, CDL E
	Velocity	5.04, 18, 28.8, 39.6 km/h
	SNR	[-5, 15] dB

Table 3: Parameter ranges used for dataset generation

Table 3 summarizes the key parameters used during the signal generation process, including the channel models, user velocities, and SNR settings.

For training, the dataset is constructed using three distinct CDL channel models: CDL A, CDL B, and CDL D. These channel models are selected to capture a mix of LoS and NLoS propagation effects, which are essential for learning generalized channel conditions. The UE velocity during training is fixed at 36 km/h, corresponding to a moderate mobility condition. A single SNR value of 16 dB is chosen to provide a balance between noise robustness and signal clarity, thereby enabling stable learning across diverse channel realizations.

The testing dataset is designed to rigorously evaluate the generalization ability of the trained neural networks. It employs two unseen channel models, CDL C and CDL E, to ensure the neural networks are not overfitted. The evaluation covers a wide range of mobility scenarios with UE velocities set at 5.04 km/h (walking), 18 km/h (biking), 28.8 km/h (tram), and 39.6 km/h (metro). This setup allows for performance analysis under different Doppler and fading conditions. The SNR for testing spans from –5 dB to 15 dB, to test the robustness of the neural networks under both noisy and clear transmission conditions.

Each input sample provided to the neural networks consists of the real and imaginary components of the raw channel estimate. These components are concatenated along the layer dimension, preserving the spatial and temporal structure of the input while maintaining compatibility with the neural network architecture. This representation enables the neural networks to learn complex-valued channel patterns while maintaining a real-valued computational framework suitable for deep learning.

5.2 Exploratory Data Analysis

5.2.1 DeModulation Reference Signals

In this thesis, single-symbol DMRS is employed with Type 1 configuration, conforming to the 3GPP specification. Figure 18 illustrates the specific allocation of DMRS in a RG, where the pilot signals are interleaved across the subcarriers and positioned in the third OFDM symbol of each slot.

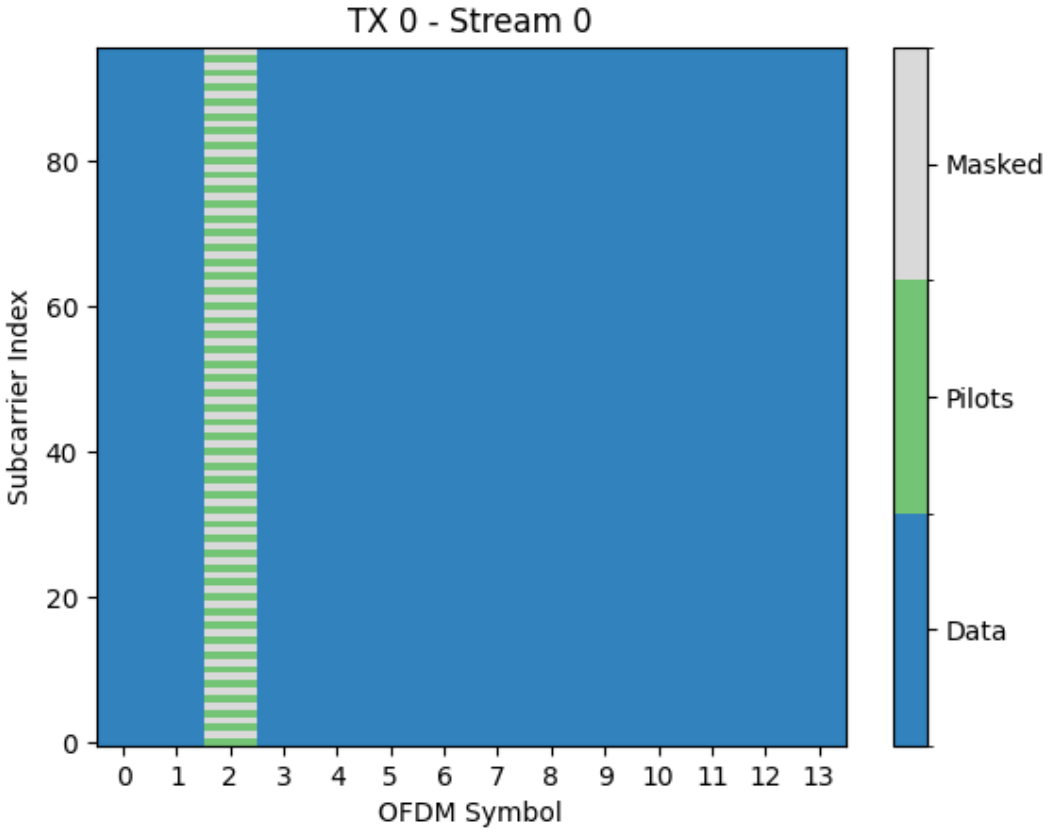


Figure 18: Allocation of single-symbol DMRS in a RG for Type 1 configuration. The DMRS is interleaved across subcarriers and positioned in the third OFDM symbol of each slot, optimizing channel estimation by providing representative channel conditions across the bandwidth while maintaining spectral efficiency.

In the frequency domain, DMRS is placed on every alternate subcarrier index within each RB to maximize throughput and support channel estimation. This structured pattern makes channel estimation more efficient by applying interpolation between each two pilot signals. The interleaving of DMRS provides representative channel conditions across the full bandwidth while maintaining a sparse overhead.

Temporally, DMRS is positioned in the front OFDM symbol of each slot within the 14-symbol frame. This supports early channel estimation in the transmission timeline, which could enhance the receiver's ability to adapt to rapid channel fluctuations, particularly in high-mobility scenarios.

The placement of DMRS in the thesis represents a deliberate design trade-off between estimation accuracy and spectral efficiency. While the sparse pilot allocation may impose challenges for traditional channel estimation, it can effectively increase throughput and is feasible to be solved by using AI to enhance or replace traditional channel estimation methods. In this context, the sparsity of DMRS compels the AI to infer robust representations of the channel model, thereby aligning the dataset design with the operational realities of 5G systems.

5.2.2 Channel Plots

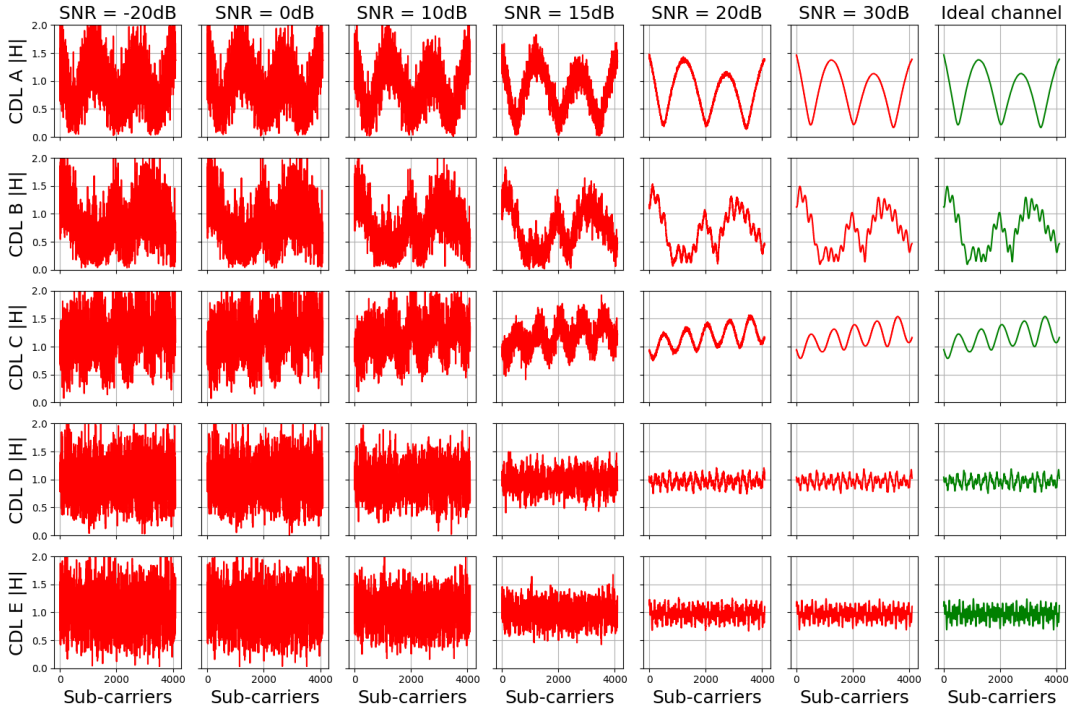


Figure 19: Channel estimates $\hat{H}_{\text{LS}}$ under varying SNR conditions compared to ideal channel conditions. This figure demonstrates the impact of SNR on the channel estimation across different CDL models (A to E), highlighting the transition from noise-dominated estimates at low SNR to more accurate multipath representations at higher SNR levels. The $\hat{H}_{\text{LS}}$ in CDL D/E remain more concentrated around 1, compared to the $\hat{H}_{\text{LS}}$ in CDL A/B/C

To further understand the characteristics of the wireless channel models in the thesis, visual analyses are conducted on both the channel conditions and their corresponding statistical distributions. These analyses, presented in Figures 19 and 20, encompass multiple SNR levels and channel models as defined by the 3GPP from CDL A to CDL E.

Figure 19 illustrates channel estimate $\hat{H}_{\text{LS}}$ across frequency for a range of SNR values, beginning from severely degraded conditions (−20 dB) and extending to high-quality transmissions (30 dB), and the ideal channel conditions. At low SNR

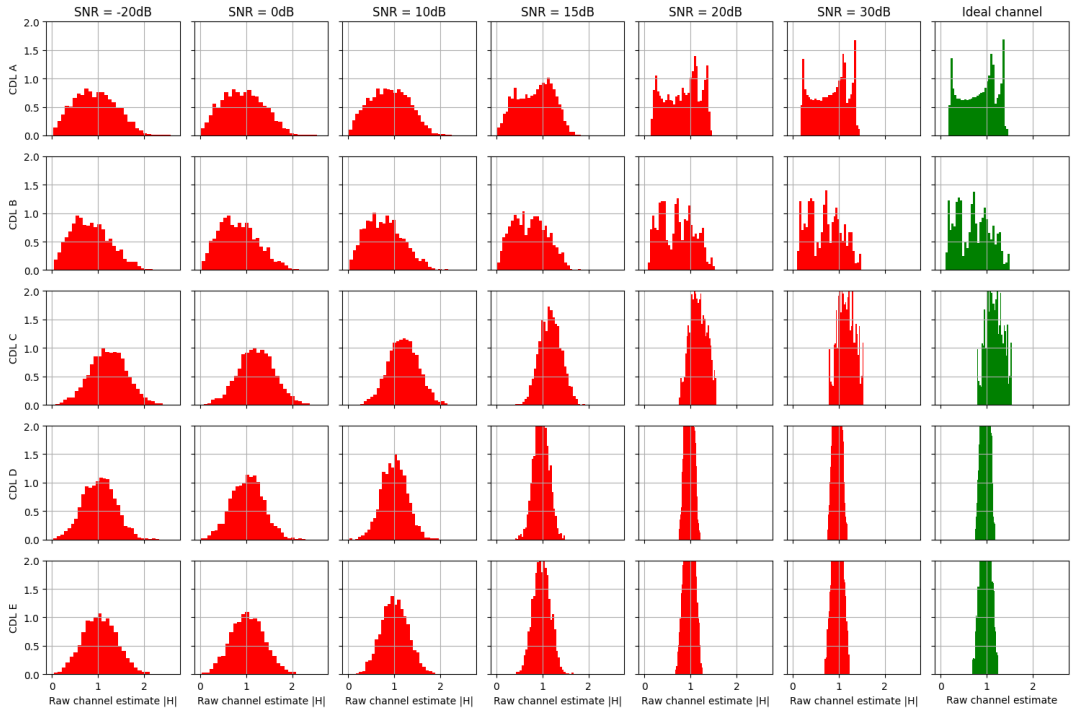


Figure 20: Distribution of LS channel estimates under different SNR conditions compared with ideal channel conditions. This figure presents histograms illustrating how the quality of channel estimates evolves with SNR, showing broader distributions at low SNR due to noise and more peaked distributions at higher SNR, especially in CDL D and E which have a direct path.

levels, particularly in CDL D and CDL E, the magnitude plots are dominated by noise, masking the underlying channel structure. As the SNR increases, the noise influence diminishes, progressively unveiling the channel's multipath characteristics. For example, in CDL A, the effect of frequency selective fading becomes apparent with increasing SNR, approximating the ideal channel conditions in the high-SNR plot. In contrast, the $\hat{H}_{\text{LS}}$ in CDL D and CDL E remain more concentrated around 1, attributable to the direct path of channel models.

Complementing the magnitude plots, Figure 20 displays histograms of the LS channel estimate across different SNR values and ideal channel conditions. These distributions reveal how the channel estimation quality evolves with SNR. Under low SNR, the distributions resemble broadened Rayleigh-like profiles, consistent with the dominance of noise and random fading effects. As the SNR improves, the distributions narrow and become more peaked, indicating increased fidelity in the channel estimates. Notably, channels modeled by CDL D and CDL E demonstrate tighter and more symmetric distributions at higher SNRs, reflecting simpler propagation conditions. Conversely, CDL A, CDL B and CDL C exhibit broader distributions even under favorable SNR, underscoring their representation of more challenging propagation scenarios.

5.3 Experiments Setting

The experiments are designed to evaluate the performance of throughput-optimized neural network, compared with the ideal channel conditions based neural network and the conventional LS estimation method in various scenarios. The core objective is to assess whether throughput-oriented optimization strategy is feasible for neural networks to improve the accuracy of signal transmission and ultimately maximize throughput to replace the optimization strategy based on ideal channel conditions, which is hard to apply in real life.

5.3.1 Evaluation Scenarios

To ensure robust generalization, both neural networks are evaluated exclusively on channel conditions not seen during training. Specifically, the Clustered Delay Line models CDL C and CDL E are selected as test-only channel models. CDL C simulates challenging NLoS conditions, while CDL E represents environments with a dominant LoS component.

The evaluation is conducted across four distinct mobility scenarios, each representing a different user velocity and thus different levels of Doppler spread and channel time-variation. Using CDL C and CDL E channel models together in every test scenario ensures diversity in channel complexity, irrespective of user speed.

Scenario 1 Walking (1.4 m/s) This scenario simulates low-mobility conditions, such as pedestrian usage in urban or suburban environments. The slow user movement leads to minimal Doppler effects, resulting in a relatively time-invariant channel. This case provides a baseline to examine how well the neural networks perform when the channel varies only gradually over time.

Scenario 2 Biking (5 m/s) In this scenario, the user velocity represents moderate mobility, typical of scenarios such as cycling in urban areas or moderate-speed movement of small vehicles. The Doppler spread increases compared to the walking case, introducing more rapid changes in the channel. This tests the neural network's ability to track moderately time-varying channel conditions.

Scenario 3 Tram (8 m/s) This higher mobility case corresponds to a user inside a tram or light rail transit. The channel varies significantly over time due to both mobility and more complex urban propagation, including reflections and diffraction. This scenario challenges the neural network's robustness against faster fading.

Scenario 4 Metro (11 m/s) This scenario captures high-mobility conditions, such as metro or train movement in densely built-up areas or tunnels. The Doppler shift is pronounced, and the channel coherence time is short. This represents one of the most challenging environments for channel estimation and is used to assess the upper limits of the smoothing neural networks' adaptability.

5.3.2 Performance Visualization

To isolate the contribution of the channel smoothing, all other receiver components are held constant across experiments. Performance is evaluated in terms of BER and BLER across a range of SNR values from −5 dB to 15 dB. Each data point is averaged over a maximum of 100 iterations.

The proposed throughput-optimized neural network and the ideal channel conditions based neural network are compared against the *Ideal* and conventional LS channel estimator. These channel estimation methods are described in detail in section 4, with the Ideal as comparative references to highlight their respective trade-offs and strengths.

By encompassing a broad spectrum of realistic mobility scenarios and propagation dynamics, this experimental setup provides a robust framework for assessing the real-world applicability and effectiveness of AI-based channel smoothing techniques in 5G communication systems.

6 Analysis of the Results

The traditional and AI-based channel estimators provide various visualizations of evaluation results. Specifically, the evaluation focuses on two different optimization techniques for the same neural network architecture. One is trained to minimize error relative to ideal channel conditions, and the other is trained end-to-end to optimize decoding reliability using LLRs. The conventional LS channel estimator serves as the baseline for these two AI channel estimators. The visualizations exhibit channel estimate, BER, and BLER values for various scenarios which include Walking (1.4 m/s), Biking (5 m/s), Tram (8 m/s), and Metro (11 m/s), tested over CDL C and CDL E channel models.

6.1 Channel Estimate Visualization

Figure 21 provides a qualitative comparison of the channel estimates generated by each channel estimator in two channel models. Although it displays only one representative example, it shows essential differences in performance between each estimator in the frequency domain.

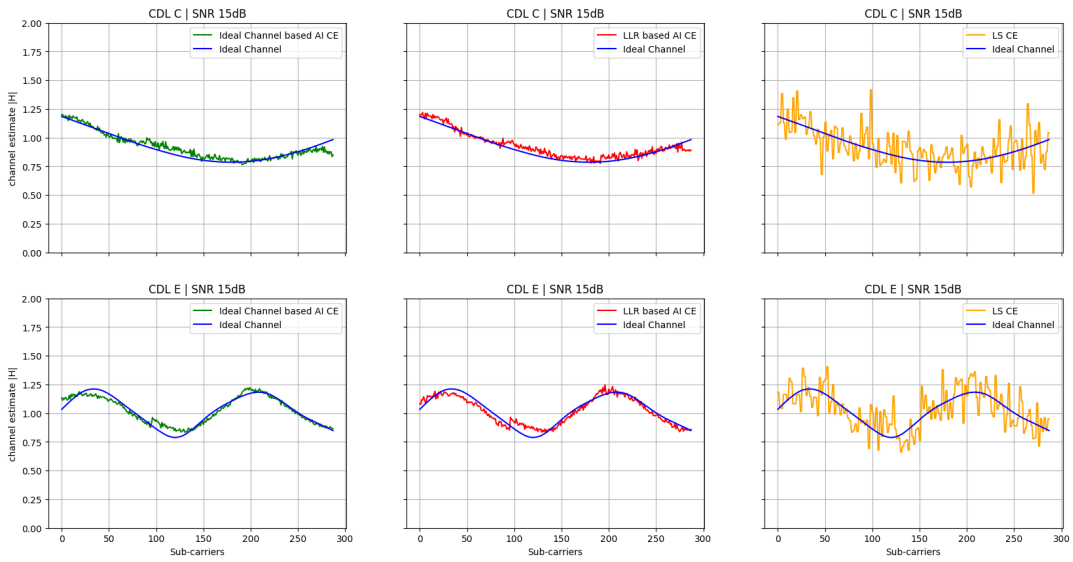


Figure 21: Comparison of channel estimates from different estimators across two channel models in the frequency domain. The LS channel estimator exhibits significant noise and irregularity, while the ideal channel-based AI estimator closely tracks the true channel with effective noise suppression. The throughput-optimized AI estimator produces smoother estimates that emphasize decoding performance over precise channel reconstruction.

The LS channel estimator produces a visibly noisy and uneven profile, making it difficult to discern the true underlying channel conditions. This jaggedness is a hallmark of estimation errors in noisy environments, where LS fails to adequately suppress noise. In contrast, the ideal channel conditions based AI estimator closely follows

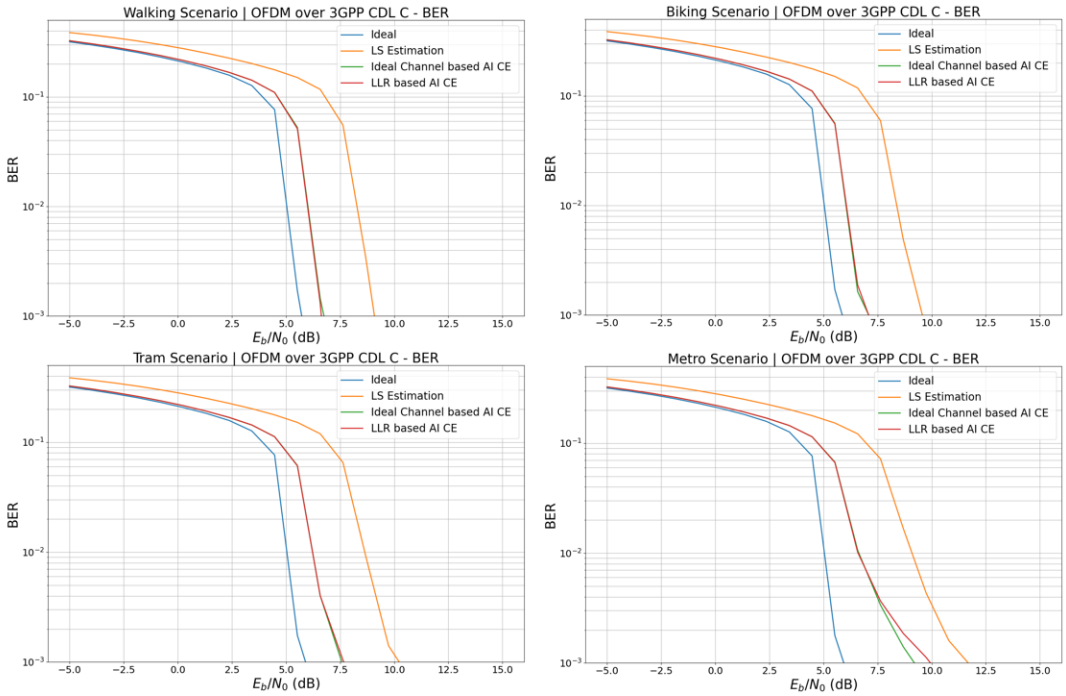


Figure 22: BER performance across different mobility scenarios using the 3GPP CDL C channel model. The figure compares the performance of LS estimation, ideal channel-based AI estimation, and LLR-based AI estimation, highlighting the superior performance of AI methods, particularly in low-mobility scenarios such as walking and biking, where they achieve lower BER at lower E_b/N_0 values.

the ideal channel trace. It demonstrates excellent noise suppression capability while retaining the fine structure of the channel conditions. It achieves this by minimizing the squared error between the channel estimate and ideal channel conditions, making it particularly well suited for producing visually accurate estimates.

The throughput-optimized AI estimator shows a similar but different channel estimate against the ideal channel conditions based AI estimator. Its estimate is slightly less precise, especially around rapid variations in the channel. However, this is because the neural network in this channel estimator is trained using LLRs from the MAP, shaping its estimate to maximize end-to-end performance rather than estimation accuracy. As a result, it intentionally smooths out high-frequency fluctuations that may confuse the decoder, even if those fluctuations are present in the ideal channel. Thus, it captures the general channel shape while filtering out noise that could impair decoding.

6.2 Performance across CDL C and CDL E

The BER curves in Figures 22 and 23 show consistent trends. In low-mobility scenarios like walking and biking, the throughput-optimized estimator outperforms both the LS and ideal channel-based estimators. Both AI methods achieve slightly lower BER

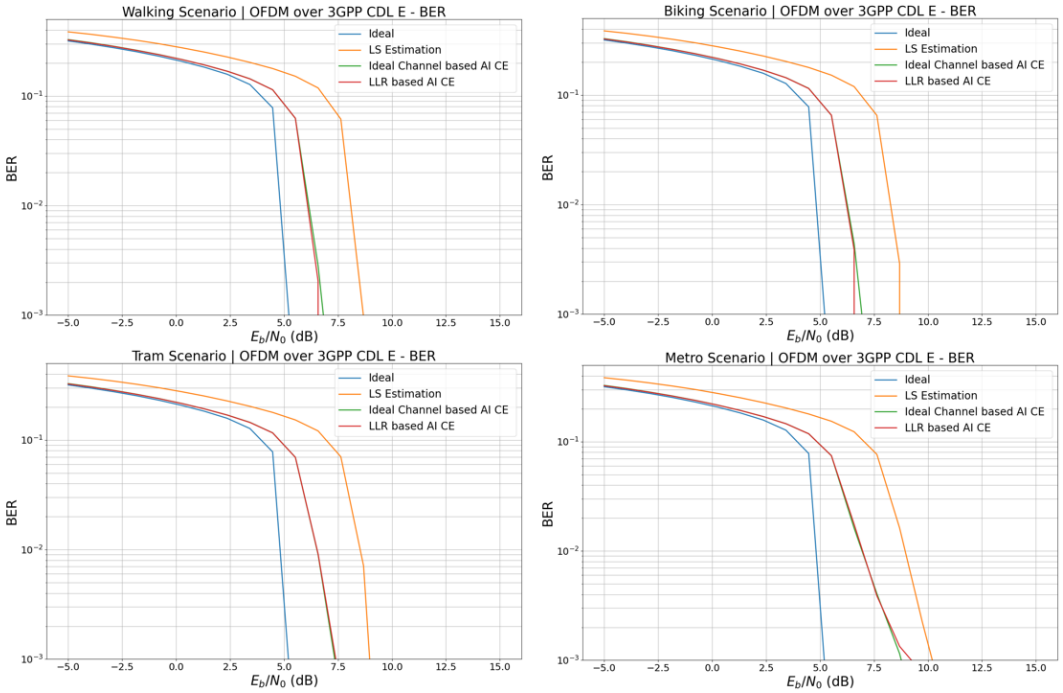


Figure 23: BER performance across different mobility scenarios using the 3GPP CDL E channel model. The figure demonstrates the performance shift favouring the ideal channel-based AI estimator in high-mobility scenarios, such as tram and metro, where it maintains lower BER across a range of E_b/N_0 values.

when the E_b/N_0 is in the $5 - 7.5$ dB range, while the LS curve shows a much more gradual decline, only achieving acceptable BER values at higher E_b/N_0 (> 8 dB). For example, in CDL C for the biking scenario, the LLR-based method achieves a BER of around 10^{-3} at roughly 7 dB, while LS remains above 10^{-2} until 8 dB.

As the UE speed increases, in the tram and metro scenarios, the ideal channel-based neural network gradually surpasses the throughput-optimized one. In CDL E under the metro condition, the LLR-based estimator plateaus around 10^{-3} in the 8.75 to 9.5 dB range, while the ideal neural network continues to decline faster, reaching 10^{-3} by about 8.5 dB. LS estimation struggles to go to 10^{-3} even at 10 dB, underlining its inability to cope with high Doppler and fast fading.

The performance shift from the LLR-based to the ideal channel-based neural network at higher UE velocities indicates that the structural fidelity of the channel estimate becomes increasingly important. The ideal neural network better captures the time-frequency correlations across subcarriers and symbols, which is especially beneficial in scenarios with rapidly varying channels.

Interestingly, in CDL E which includes a strong LoS path, the LLR-based estimator appears to hold its ground longer than in CDL C. It benefits from the predictable component in the channel, allowing its smoothed representation to remain effective for decoding. This suggests that the throughput-optimized estimator generalizes well in structured fading environments, despite being trained on different CDL models.

The BLER results in Figures 24 and 25 offer deeper insight into decoding stability

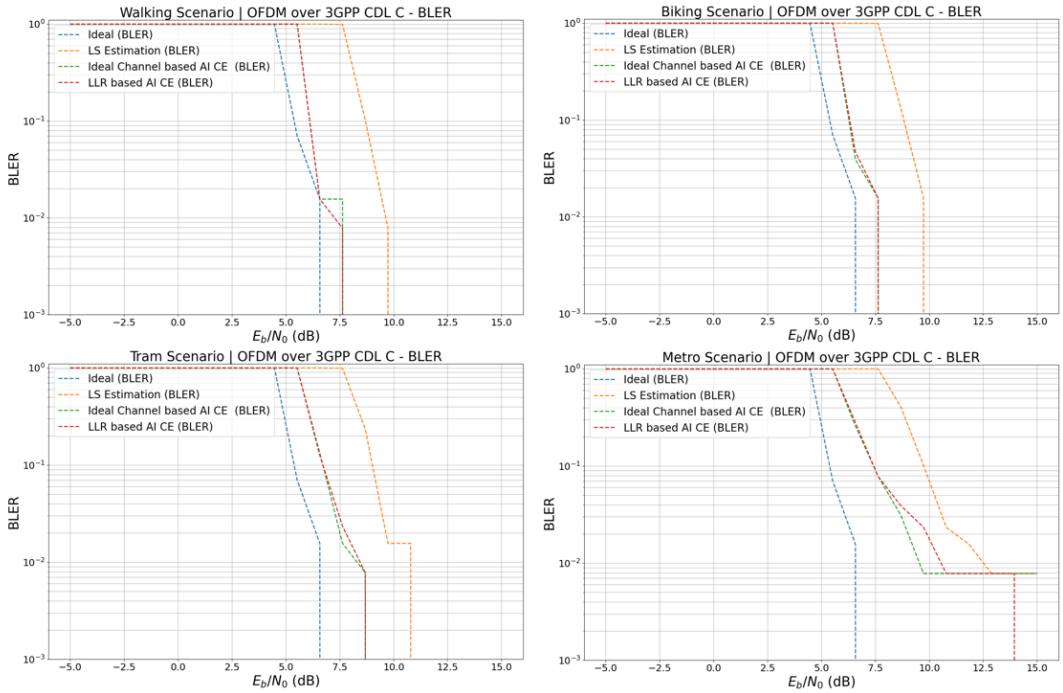


Figure 24: BLER performance across different mobility scenarios using the 3GPP CDL C channel model. This figure illustrates the effectiveness of AI-based estimators in achieving lower BLER at reduced E_b/N_0 levels compared to LS estimation, especially in low-mobility conditions.

across mobility levels. In the walking scenario, both AI-based estimators quickly reach BLER values below 10^{-1} at E_b/N_0 as low as 6.5 dB, while the LS method requires 8–9 dB. In the biking scenario, the LLR-based estimator achieves BLER values close to 10^{-3} around 7 dB.

In high-mobility cases like tram and metro, differences become clearer. In CDL C tram conditions, the LS channel estimator fails to reach BLER $< 10^{-1}$ until beyond 8.5 dB, while both neural networks achieve this threshold around 6.5–7.5 dB. Under metro conditions in CDL E, the ideal channel-based neural network maintains lower BLER across all E_b/N_0 values, particularly excelling between 7.5 and 10 dB. The LLR-based estimator still tracks well and remains usable, though its curve begins to flatten, suggesting slightly reduced reliability as channel dynamics intensify.

Across all conditions, the AI-based estimators outperform the LS method consistently. The throughput-optimized neural network converges faster in low-mobility and lightly faded channels, while the ideal channel-based neural network retains an edge under severe fading. In real-world terms, this means that LLR-based smoothing offers quicker convergence and more robustness at moderate speeds, while the ideal neural network offers greater performance stability at the edge of receiver sensitivity.

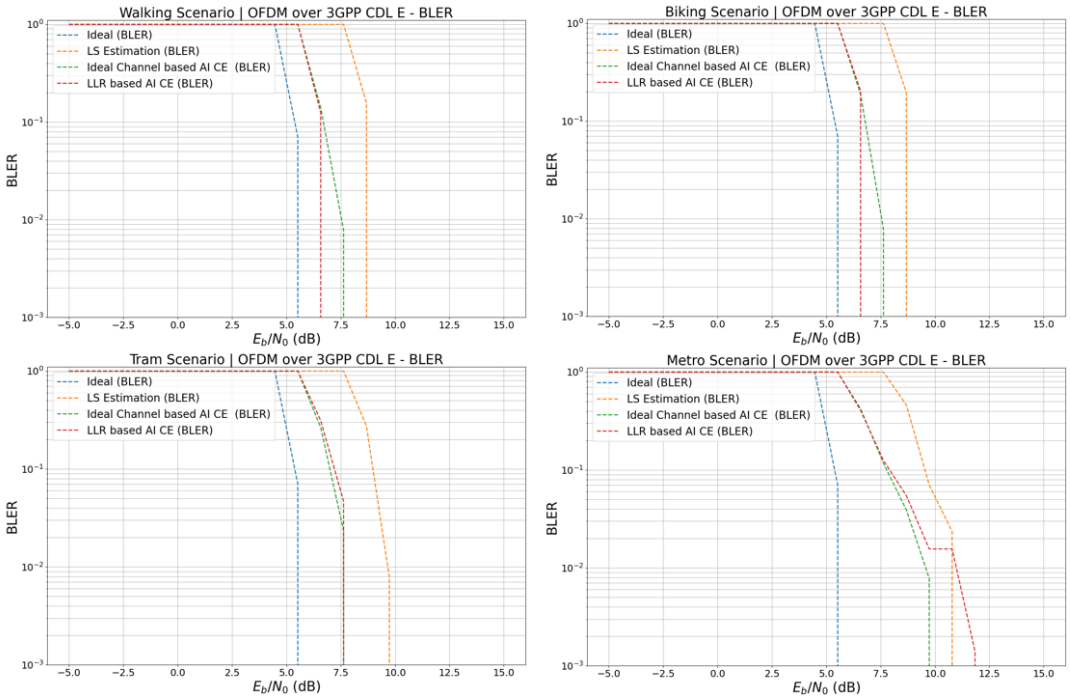


Figure 25: BLER performance across different mobility scenarios using the 3GPP CDL E channel model. The figure highlights the robust performance of AI-based estimators, particularly the ideal channel-based AI estimator, in achieving lower BLER in high-mobility scenarios, outperforming LS estimation across all E_b/N_0 values.

7 Conclusions and Future Work

7.1 Conclusions

According to the thesis objective, the performance of throughput-optimized channel smoothing has been investigated, and compared to that of the ideal channel conditions based channel smoothing and the traditional LS channel estimator.

The results demonstrate that the AI-based channel estimator improves substantially over the traditional LS channel estimator. Across all tested scenarios, the estimators with AI channel smoothing outperformed LS channel estimator, often by a wide margin in both BER and BLER. This confirms that AI channel smoothing has real potential to enhance the robustness and reliability of 5G receivers.

The differences between these two AI channel smoothing methods lead to a more nuanced conclusion. The throughput-optimized AI method, trained on LLRs, has showed its strengths especially at lower UE speed. In walking and biking scenarios, it consistently delivers the lowest BER and often meets the target BLER threshold of 10^{-1} with lower required E_b/N_0 value than the other methods.

However, when the UE speed increased beyond 5 m/s, the performance shifts. In higher-mobility scenarios like tram or metro conditions, the neural network trained on ideal channels becomes more effective. Its focus on preserving the fine structure of the channel allows it to better adapt to the rapid variations in the signal, helping it outperform the LLR-based neural network. This suggests that while throughput-optimized channel smoothing is more suitable for general purpose, ideal channel based method still has value when channel dynamics become more extreme.

Channel type also has an impact. The LLR-based neural network shows stronger relative performance in CDL E than in CDL C, which is because CDL E includes a strong LoS path that the neural network can rely on to anchor its smoothing, helping it maintain decoding performance even when the channel is highly scattered. Still, in CDL E at high mobility, the ideal channel-based neural network ultimately shown more resilient.

Given that the data required by the throughput-optimized neural network is readily available in real-world scenarios, it stands as a strong candidate for practical deployment in typical mobility environments. This neural network offers decoding performance comparable to that of the ideal neural network while being trained without access to ideal channel conditions. It performs particularly well at the 10^{-1} BLER threshold, which makes it suitable for real-world systems where that metric often defines acceptable quality.

7.2 Future Work

A natural next step is to transmit the proposed throughput-oriented optimization strategy for neural network in channel smoothing from simulation to real world deployment. In practice, this could involve collecting received channel estimates and transmitted bits from a live system to train the model and evaluate its impact on throughput. While this thesis relies on the `Sionna` simulator, which provides a controlled environment with

simplified channel models, the real wireless medium is considerably more complex due to factors such as interference, mobility, and multi-antenna processing [58].

Deploying the proposed optimization strategy in channel estimation in a real 5G system raises several challenges. The training process is a bit slower than with ideal channels, since the equalization is computationally expensive. Additionally, the practical receiver chain must accounts for interference from neighboring UEs and BSs, which significantly affects channel estimation and decoding performance. Addressing these challenges is essential for bringing throughput-oriented optimization strategy to practical use in next-generation wireless networks.

References

[1] F. Rinaldi, A. Raschella, and S. Pizzi, "5G NR system design: A concise survey of key features and capabilities," *Wireless Networks*, vol. 27, no. 8, pp. 5173–5188, 2021.

[2] S. Parkvall, Y. Blankenship, R. Blasco, E. Dahlman, G. Fodor, S. Grant, E. Stare, and M. Stattin, "5G NR release 16: Start of the 5G evolution," *IEEE Communications Standards Magazine*, vol. 4, no. 4, pp. 56–63, 2020.

[3] X. Lin, "An overview of 5G advanced evolution in 3GPP release 18," *IEEE Communications Standards Magazine*, vol. 6, no. 3, pp. 77–83, 2022.

[4] R. Verdecia-Peña, R. Oliveira, and J. I. Alonso, "5G wireless channel estimation: Addressing PHY-layer impairments through model-based deep learning," in *2024 IEEE 22nd Mediterranean Electrotechnical Conference (MELECON)*, 2024, pp. 532–537.

[5] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, "Channel estimation for OFDM," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1891–1908, 2014.

[6] A. S. M. Mohammed, A. I. A. Taman, A. M. Hassan, and A. Zekry, "Deep learning channel estimation for OFDM 5G systems with different channel models," *Wireless Personal Communications*, vol. 128, no. 4, pp. 2891–2912, 2023.

[7] J. Bian, C.-X. Wang, X. Gao, X. You, and M. Zhang, "A general 3D non-stationary wireless channel model for 5G and beyond," *IEEE Trans. Wireless Commun.*, vol. 20, no. 5, pp. 3211–3224, 2021.

[8] C.-X. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G. Larsson, "Artificial intelligence enabled wireless networking for 5G and beyond: Recent advances and future challenges," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 16–23, 2020.

[9] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[10] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, E. A. Chavez-Urbiola, and J. A. Romero-Gonzalez, "Loss functions and metrics in deep learning," *arXiv preprint arXiv:2307.02694*, 2023.

[11] NVIDIA, "CUDA-Accelerated RAN: Channel Estimation," 2023.

[12] C. for Communications Regulation, "Radio Spectrum Management Strategy Statement 2022 to 2024 (Design Version) ComReg 21/136," 2022.

[13] R. Liu, Y. Ma, X. Zhang, and Y. Gao, "Deep learning-based spectrum sensing in space-air-ground integrated networks," *Journal of Communications and Information Networks*, vol. 6, no. 1, pp. 82–90, 2021.

[14] Nokia, "Spectrum Bands in a 5G World," 2023.

[15] K. Technologies, "OFDM Basic Principles Overview."

[16] S. Chen and C. Zhu, "ICI and ISI analysis and mitigation for ofdm systems with insufficient cyclic prefix in time-varying channels," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 78–83, 2004.

[17] G. Zhang, M. De Leenheer, A. Morea, and B. Mukherjee, "A survey on OFDM-based elastic core optical networking," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 65–87, 2013.

[18] B. Liu, X. Lyu, and W. Fan, "Analysis of 5G signal for radar application," in *Journal of Physics: Conference Series*, vol. 2356, no. 1. IOP Publishing, 2022, p. 012027.

[19] A. A. Zaidi, R. Baldemair, V. Moles-Cases, N. He, K. Werner, and A. Cedergren, "OFDM numerology design for 5G new radio to support IoT, eMBB, and MBSFN," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 78–83, 2018.

[20] ShareTechnote, "5G Resource Grid."

[21] G. Cisek and T. P. Zieliński, "Prototyping software transceiver for the 5G new radio physical uplink shared channel," in *2019 Signal Processing Symposium (SPSympo)*, 2019, pp. 150–155.

[22] E. Dahlman, S. Parkvall, and J. Sköld, "Chapter 9 - transport-channel processing," in *5G NR: the Next Generation Wireless Access Technology*. Academic Press, 2018, pp. 153–182.

[23] J. Meredith, "Study on channel model for frequency spectrum above 6 ghz," *3GPP TR 38.900, Jun, Tech. Rep.*, 2016.

[24] I. A. Hemadeh, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, "Millimeter-wave communications: Physical channel models, design considerations, antenna constructions, and link-budget," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 870–913, 2018.

[25] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[26] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[27] K. Saitoh, *Deep learning from the basics: Python and deep learning: Theory and implementation*. Packt Publishing Ltd, 2021.

[28] G. E. Hinton, "Learning multiple layers of representation," *Trends in cognitive sciences*, vol. 11, no. 10, pp. 428–434, 2007.

[29] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature machine intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[30] M. Kaloev and G. Krastev, "Comparative analysis of activation functions used in the hidden layers of deep neural networks," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1–5.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[32] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, 1994.

[33] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learning*, vol. 30. Atlanta, GA, 2013, p. 3.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1026–1034.

[35] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, vol. 4, no. 5, p. 11, 2015.

[36] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[37] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, p. 12, 1991.

[38] D. Kinga and J. B. Adam, "A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, vol. 5, no. 6, 2015.

[39] E. J. Rhee, "A deep learning approach for classification of cloud image patches on small datasets," *Information and Communication Convergence Engineering*, 2018.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vision and Pattern Recognition*, 2016, pp. 770–778.

[41] W. Jiang and H. D. Schotten, "Neural network-based fading channel prediction: A comprehensive overview," *IEEE Access*, vol. 7, pp. 118 112–118 124, 2019.

[42] F. D. Coutinho, H. S. Silva, P. Georgieva, and A. S. Oliveira, "5G cascaded channel estimation using convolutional neural networks," *Digital Signal Processing*, vol. 126, p. 103483, 2022.

[43] M. Meenalakshmi, S. Chaturvedi, and V. K. Dwivedi, "Enhancing channel estimation accuracy in polar-coded MIMO–OFDM systems via cnn with 5G channel models," *AEU-International Journal of Electronics and Communications*, vol. 173, p. 155016, 2024.

[44] V. S. Usatyuk and S. I. Egorov, "2D/3D ResNet deep neural network for 4G and 5G NR wireless channel estimation," in *2023 25th International Conference on Digital Signal Processing and its Applications (DSPA)*, 2023, pp. 1–4.

[45] J. Zhao, Y. Wu, Q. Zhang, and J. Liao, "Two-stage channel estimation for mmWave massive MIMO systems based on ResNet-UNet," *IEEE Systems Journal*, vol. 17, no. 3, pp. 4291–4300, 2023.

[46] R. Shankar, "Bi-directional LSTM based channel estimation in 5G massive MIMO OFDM systems over TDL-C model with rayleigh fading distribution," *International Journal of Communication Systems*, vol. 36, no. 16, p. e5585, 2023.

[47] A. Baz, J. Logeshwaran, Y. Natarajan, and S. K. Patel, "Enhancing mobility management in 5G networks using deep residual LSTM model," *Applied Soft Computing*, vol. 165, p. 112103, 2024.

[48] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent traffic steering in beyond 5G open RAN based on LSTM traffic prediction," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7727–7742, 2023.

[49] S. Zhao, Y. Fang, and L. Qiu, "Deep learning-based channel estimation with SRGAN in OFDM systems," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[50] W. Tong, W. Xu, F. Wang, J. Shang, M. Pan, and J. Lin, "Deep learning compressed sensing-based beamspace channel estimation in mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 11, no. 9, pp. 1935–1939, 2022.

[51] D. Luan and J. S. Thompson, "Channelformer: Attention based neural solution for wireless channel estimation and effective online training," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6562–6577, 2023.

[52] D. Minovski, N. Ögren, K. Mitra, and C. Åhlund, "Throughput prediction using machine learning in LTE and 5G networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 3, pp. 1825–1840, 2023.

[53] A. Al-Thaedan, Z. Shakir, A. Y. Mjhool, R. Alsabah, A. Al-Sabbagh, F. Nembhard, and M. Salah, "A machine learning framework for predicting downlink throughput in 4G-LTE/5G cellular networks," *International Journal of Information Technology*, vol. 16, no. 2, pp. 651–657, 2024.

[54] L. Li and T. Ye, "Research on throughput prediction of 5G network based on LSTM," *Intelligent and Converged Networks*, vol. 3, no. 2, pp. 217–227, 2022.

[55] M. Honkala, D. Korpi, and J. M. J. Huttunen, "Deeprx: Fully convolutional deep learning receiver," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3925–3940, 2021.

[56] NVIDIA, "Sionna: End-to-end communications library," 2025.

[57] L. Deneire, P. Vandenameele, L. van der Perre, B. Gyselinckx, and M. Engels, "A low-complexity ML channel estimator for OFDM," *IEEE Trans. Commun.*, vol. 51, no. 2, pp. 135–140, 2003.

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