Robust and Efficient CSI Compression Using CLLWCsiNet in Noisy MIMO Environments

Fardad Ansari*, Dr.Mahmood Mohassel Feghhi†
*Faculty of Electronic and Computer Engineering, †Faculty of Electronic and Computer Engineering,
*University of Tabriz, †University of Tabriz,
*f.ansari99@ms.tabrizu.ac.ir, †mohasselfeghhi@tabrizu.ac.ir

***Abstract*—Multiple-Input Multiple-Output (MIMO) systems play a crucial role in advancing 5G and 6G networks by improving spectral efficiency and data rates. In Frequency Division Duplex (FDD) systems, since the channel reciprocity does not hold, timely feedback of Channel State Information (CSI) from receiver to transmitter is essential to enable beamforming and precoding - which are fundamental techniques in MIMO systems. However, in massive MIMO configurations with large antenna arrays, this feedback mechanism introduces significant overhead to the system. Among different compression methods, deep learning based approaches outperform traditional compressive sensing techniques, but their large parameter count poses challenges for deployment on resource-limited devices, such as user equipment. Moreover, factors such as short channel coherence time and imperfect wireless channels demand more robust models. This paper introduces CLLWCsiNet (Convolutional Latent-space Low Weight CsiNet), a novel architecture that employs convolutional latent-space learning under imperfect channel conditions. CLLWCsiNet demonstrates superior predictive accuracy in noisy environments. We provide a comprehensive overview of previous models and introduce convolutional quantization techniques, which significantly reduce parameters and enhance computational efficiency. Our approach effectively compresses CSI while maintaining predictive accuracy, making it well-suited for deployment on resource-constrained user equipment. Simulations show that CLLWCsiNet achieves -11 dB Normalized Mean Square Error (NMSE) in low SNRs (under 20 dB) with only 70K parameters. Additionally, we evaluate the proposed model on the Quadriga dataset for the first time, and demonstrating CLLWCsiNet's robust performance across varying noise levels.**

***Index Terms*—Massive MIMO, CSI Feedback , deep learning, Convolutional Quantization, Dynamic Quantization.**

I. Introduction

Multiple-Input Multiple-Output (MIMO) systems play a pivotal role in the advancement of 5G and the forthcoming 6G networks by significantly enhancing spectral efficiency, data rates, and overall network capacity. [1] In Frequency Division Duplex (FDD) systems, accurate and timely feedback of channel state information (CSI) from the receiver to the transmitter is crucial for optimal performance. This feedback enables the transmitter to adapt its signaling strategies, thereby improving link reliability and throughput. However, the feedback process introduces substantial overhead, which can degrade system efficiency if not managed properly. Therefore, studying and mitigating feedback overhead is essential to fully realize the benefits of MIMO technology in next-generation wireless communications (Figure 1) [2].

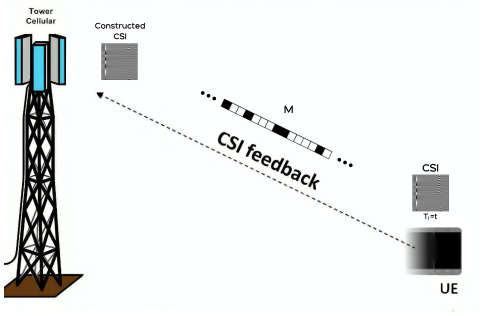

Fig. 1: Downlink From UE to Base-Station

CSI feedback compression is a critical component in MIMO Frequency Division Duplex (FDD) systems, particularly for enhancing spectral efficiency and network performance in 5G and emerging 6G networks. Traditional methods for CSI feedback compression, such as scalar and vector quantization [3], Principal Component Analysis (PCA) [4], and codebook-based techniques [5], have been widely utilized. These methods, while effective to some extent, often struggle with the trade-off between compression efficiency and the accuracy of CSI reconstruction. For instance, scalar and vector quantization reduce CSI dimensionality by mapping elements to predefined levels or codewords, whereas PCA retains the most significant components through orthogonal transformations. Codebook-based methods, which are integral to standards like LTE, involve selecting the closest codeword from a predefined set to represent the CSI. Despite their utility, these traditional methods may not fully capture the complex and dynamic nature of wireless channels.

Recent advancements leverage modern deep learning techniques to enhance CSI feedback compression. Methods such as autoencoders, recurrent neural networks (RNNs), and convolutional neural networks (CNNs) have demonstrated superior performance by learning efficient representations of the CSI directly from data. Autoencoders compress the CSI into a lower-dimensional latent space and reconstruct it with minimal loss, while RNNs capture temporal correlations in time-varying channels. CNNs exploit the spatial structure of the CSI, making them particularly effective for large-scale MIMO systems. More sophisticated models like Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) provide robust compression by learning probabilistic

representations, offering resilience against noise and channel imperfections. Additionally, hybrid methods combining traditional techniques with deep learning models can further enhance performance by leveraging the strengths of both approaches.

For the first time, Wen et al. [6] introduced a deep learning solution for CSI feedback compression by employing an auto-encoder architecture. Their work serves as the foundation for numerous subsequent research efforts aiming to incorporate the behavior of wireless communication systems within MIMO systems using an auto-encoder architecture. In this setup, the encoder component assumes the role of the user equipment, while the decoder portion functions as the decoder. The random spatial characteristics of channel state information have been simulated using the COST2100 channel model [7]. The COST2100 dataset has been generated accordingly. Proposed architectures have been predominantly founded upon auto-encoder structures. Wen et al. [6] introduced CsiNet, an autoencoder that employs the Residual Network [8] concept in the decoder part. Wang et al [9]. introduced time-varying CSI Feedback, aiming to simulate the coherence time in the wireless channel attributed to user equipment mobility. They incorporated Long Short-Term Memory (LSTM) units into the decoder part of an auto-encoder. Lu et al. [10] implemented the LSTM architecture in both the encoder and decoder parts of their model. They opted for a neural network instead of a convolutional form for LSTM, which resulted in a doubling of parameters and complexity compared to CsiNet-LSTM [9], as reported. They emphasized memory for feature extraction module more than correlation by employing LSTM in parallel with a fully connected layer, akin to jump connections in residual networks [10]. Their proposed architecture, RecCsiNet [10], achieved superior results prior to considering time correlation, by employing LSTM for residual features instead of correlation features [9]. Li and Wu [11] extended the RecCsiNet model by separating the feature extraction process from feature compression and decompression. Their proposed model, Convlstm, significantly reduces the error by a considerable amount [11]. Liu et al. [12] emphasized coherence time simulation through LSTM and proposed a new architecture called Markovenet. In addition to their suggested architecture and a simple Markov model, they introduced a new convolutional latent space that significantly reduces the number of parameters and complexity of the autoencoder. They argue that instead of using fully connected layers to extract features, which may not be crucial in the random characteristics of the channel state information matrix, a convolutional approach is more appropriate, given that only adjacent elements in the channel state information matrix exhibit strong correlations. Mashhadi et al. [13] proposed the DeepCMC network, which is a fully convolutional neural network. Dimensional compression is achieved through upsampling and downsampling in pooling layers. To further compress the representation, they utilize a context-adaptive binary arithmetic coding scheme for the encoder side of the DeepCMC output codeword. Liang et al. [14] proposed a compressive sensing architecture named CSRenet, which employs compressive sensing on the encoder side and a deep learning-based method on the decoder side. Lu et al. [15] introduced the CRNet architecture, representing one of the pioneering studies utilizing small kernel sizes for CSI matrix feature granularity extraction. They employed a residual network architecture in both the encoder and decoder. Within the residual architecture, Lu et al. utilized different kernel sizes in different paths and suggested a multi-resolution block for both encoder and decoder. An important aspect of their research is that the performance of a single-resolution refine network can be enhanced by employing different kernel sizes. Guo et al. [16] proposed a multiple-rate architecture to address the challenge posed by short coherence time intervals, where CSI must be sufficiently compressed and the auto-encoder should dynamically adjust the compression ratio according to the environment. Their proposed multiple-rate framework is capable of compressing the CSI matrix at different compression ratios without altering the architecture or requiring a new training scheme. As a result, user equipment (UE) does not need to store parameters for different architectures. Guo et al. incorporated quantization and dequantization modules in the encoder and decoder, respectively. In contrast to Lu et al. [17], who used uniform quantization, Guo et al.[16] employed non-uniform quantization to generate binary streams at the output of the encoder. Additionally, they suggested using larger kernel sizes in both the encoder and decoder, as well as an expected residual configuration. These changes were applied to CsiNet, resulting in a new architecture called CsiNetPlus [16]. Yu et al. [18] employ the non-local block which enhance the accuracy with the help of long-range dependences. They inspired the non-local block from X et al. [19] which has been for sequence data denoising. Cai et al. [20] proposed a modified version of CsiNet called Attention-CsiNet. They replaced the fully connected layers in the encoder and decoder with LSTM modules to extract coherence among sub-carriers. Additionally, Cai et al. [20] introduced an Attention module for CSI feedback compression, inspired by the research of [21]. J et al. [21] proposed the squeeze-and-extraction network in the field of machine translation.

In this paper, we address the inefficiencies associated with the fully connected layers in many state-of-the-art models for CSI feedback compression. These layers often propagate redundant weights related to the zero parts of the channel state information, leading to a large number of parameters. This inefficiency is particularly problematic on User Equipment (UE), where resource constraints such as memory and computational power are critical factors. By focusing on quantization techniques, we aim to significantly reduce the parameter count and enhance the efficiency of CSI feedback compression models on UE.

We retrained the previously proposed models with a convolutional latent space on the COST2100 dataset [7], revealing that many designs fail in the absence of fully connected layers. To overcome this, we proposed a new architecture named CLLWCsiNet (Convolutional Latent-space Low-weight CsiNet), which achieves high accuracy without relying on fully connected layers and performs better with significantly fewer parameters. Furthermore, to ensure generalization, we evaluated these models on the Quadriga dataset [21], specifically

generated for outdoor scenarios. Notably, many models that exhibited high accuracy on the COST2100 dataset [7] failed when tested on the Quadriga dataset[21].

Our study also considers the imperfect channel conditions, a realistic scenario often neglected in favor of ideal conditions in wireless communication systems research. The proposed CLLWCsiNet model demonstrated performance comparable to CsiNet+DNNet [23], which has a significantly higher number of parameters. Importantly, CLLWCsiNet maintained high accuracy across varying noise power levels, showcasing robust performance. For the first time, we employed quantization not merely for fully connected layers or bit stream generation but also within the convolutional framework. This novel application of quantization is crucial for practical deployments, as it significantly reduces the computational burden and memory requirements, making the models more suitable for resource-constrained environments typical of wireless communication systems.

The importance of quantization in this context cannot be overstated. It allows for the compression of model parameters without significant loss of accuracy, facilitating the deployment of deep learning models in real-time and low-power scenarios. Models with a high number of parameters, while potentially more accurate, often face practical limitations in wireless communication systems due to their substantial resource demands, including computational power and energy consumption. By incorporating convolutional quantization, our proposed CLLWCsiNet model addresses these challenges, offering a more efficient and scalable solution for CSI feedback compression in next-generation wireless networks.

The following is a list of the contributions:

- To the best of our knowledge, this is the first time that dynamic quantization has been explicitly applied to all layers of a neural network for CSI compression tasks. Previous research has primarily focused on network binarization in fully connected layers within the latent space, aiming to enable quantization from a communication system design perspective. However, this concept can be extended to all layers, which can significantly reduce network complexity, albeit potentially at the expense of accuracy.
  Bit-level neural networks are a crucial subset of network quantization, enabling the deployment of deep learning models on devices with constrained resources, such as those found in wireless communication systems. It is important to distinguish between network quantization and codeword quantization, which involves larger compression representations as proposed in communication system designs.
- Unlike most research in this domain, our focus is on imperfect channels characterized by additive white Gaussian noise (AWGN) in wireless communication systems. Previous studies that consider imperfect channels include [23] and [20]. In [23], a DNNet was proposed, which introduces high complexity even in low dimensions, despite the training process being conducted in two stages. Researchers in [20] proposed a variational auto-encoder for imperfect channels, but they did not report numerical results. In contrast, we propose a fully convolutional neural network (CNN) with a latent space alongside a simple subtraction denoising convolution filter for stabilizing accuracy and denoising, respectively. Our architecture is trained in a single stage, unlike the approach in [23]. Our results demonstrate the state-of-the-art performance of this denoising network compared to [23].
- We propose a convolutional latent space low-light CSINet named CLLWCsiNet, which achieves high accuracy compared to architectures that use fully connected layers. Previous architectures rely heavily on fully connected layers, which introduce redundant zero weights and significantly increase the model's parameter count by approximately 2 million parameters. This increase in parameters leads to high computational costs Floating Point Operations (FLOPs) and storage requirements, making them impractical for user equipment with limited GPU power and storage capacity, especially in scenarios involving low temporal CSI information and CPU-based mass-produced user devices. Furthermore, most of these architectures are designed under ideal channel conditions.

II. System Model

A. System Configuration

Like most scenarios adopts a single user massive MIMO system in a single cell, in which the base station equips $N_t > 1$ transmitting antennas as well as the UE with a single receiving antennas. An orthogonal frequency division multiplexing (OFDM) system with N_c sub-carriers for FDD downlink massive MIMO is examined. In the OFDM-MIMO system considered in this work, the subcarriers are assumed to maintain orthogonality.

B. Channel Model

The time-domain channel impulse response between the i-th transmit antenna and UE follows Rayleigh fading:

$$h_i(\tau) = \sum_{l=1}^{L} \alpha_{i,l}\delta(\tau - \tau_l), \quad \alpha_{i,l} \sim \mathcal{CN}(0, \sigma_l^2)$$

where $\alpha_{i,l}$ is the complex gain and τ_l is the delay of the l-th path. The frequency-domain channel matrix for subcarrier n is:

$$\mathbf{H}_n = [H_{1,n}, H_{2,n}, \ldots, H_{N_t,n}] \in \mathbb{C}^{1\times N_t}$$

with elements derived from:

$$H_{i,n} = \sum_{l=1}^{L} \alpha_{i,l} e^{-j2\pi n\tau_l/N_c}$$

C. Pilot Transmission

We will assume base station sends orthogonal pilots for each subcarrier to the UE $n = 1, ..., N_c$: let pilot matrix $\mathbf{X}_{p,n} \in \mathbb{C}^{N_t\times T_p}$ be orthogonal ($T_p \geq N_t$). We will assume on n^{th} sub-carriers the received signal at UE will be equal to:

$$\mathbf{Y}_{p,n} = \mathbf{H}_n\mathbf{X}_{p,n} + \mathbf{Z}_n$$

where $\mathbf{Y}_{p,n} \in \mathbb{C}^{1\times T_p}$ ($N_r = 1$) is received pilot at n^{th} subcarrier, $\mathbf{H}_n \in \mathbb{C}^{1\times N_t}$ is the MIMO channel matrix for subcarrier n and $\mathbf{Z}_n \in \mathbb{C}^{1\times T_p}$ is AWGN noise with variance σ^2.

D. Channel Estimation

Let's assume UE uses Least Squares (LS) to estimate $\mathbf{H}_n$:

$$\hat{\mathbf{H}}_n = \mathbf{Y}_{p,n}\mathbf{X}_{p,n}^{\dagger}$$

where $\hat{\mathbf{H}}_n \in \mathbb{C}^{1\times N_t}$ and $\mathbf{X}_{p,n}^{\dagger} = \mathbf{X}_{p,n}^{H}(\mathbf{X}_{p,n}\mathbf{X}_{p,n}^{H})^{-1}$. The pilot matrix $\mathbf{X}_{p,n}$ satisfies $\mathbf{X}_{p,n}\mathbf{X}_{p,n}^{H} = \mathbf{I}_{N_t}$ for $T_p \geq N_t$.

E. CSI Feedback

For each subcarrier estimated CSI matrices is $\{\hat{\mathbf{H}}_1, ..., \hat{\mathbf{H}}_{N_c}\}$. For simplicity we stack and reshape the channel matrix into:

$$H_{\text{freq}} = \left[\hat{\mathbf{H}}_1^H, \hat{\mathbf{H}}_2^H, ..., \hat{\mathbf{H}}_{N_c}^H\right]^H_{N_c\times N_t}$$

$H_{\text{freq}} \in \mathbb{C}^{N_c\times N_t}$ is the stacked CSI matrix across subcarriers. Evidently, $H_{\text{freq}} \in \mathbb{C}^{N_c\times N_t}$ has $2 \times N_c \times N_t$ float numbers, which is too big for direct feedback in a massive MIMO system and this feedback is required for creating the precoding to be built by base station.

H_{freq} is the spatial-frequency domain equivalent of the CSI. We use the 2D discrete Fourier transform (DFT) and transfer H_{freq} into the angle-delay domain as follows to extract the CSI features:

$$\tilde{\mathbf{H}} = \mathbf{F}_d H_{\text{freq}} \mathbf{F}_a^H$$

where $\mathbf{F}_d \in \mathbb{C}^{N_c\times N_c}$ is delay DFT and $\mathbf{F}_a \in \mathbb{C}^{N_t\times N_t}$ is angular DFT and both of them are unitary DFT matrices.

CSI shows sparsity in the delay domain, with $\tilde{\mathbf{H}}$ having significant values only in the first N_i rows because the time of arrival (TOA) between multipaths is limited in duration. Like aforementioned researches, we select first N_i rows of $\tilde{\mathbf{H}}$ to create a new channel matrix $\tilde{\mathbf{H}}_f$ as

$$\tilde{\mathbf{H}}_f = \left[\tilde{\mathbf{H}}\right]_{1:N_i}$$

$\tilde{\mathbf{H}}_f$ as is still too heavy for feedback, though, because in a massive MIMO system, N_t is a big number. Our goal is to further compress matrix $\tilde{\mathbf{H}}_f$ in order to minimize the weight of the feedback.

F. Deep Learning Compression

In an effort to lower transmission overheads, DL-based algorithms have recently been used in CSI feedback. An encoder at UE first compresses the CSI data into a codeword:

$$s = f_{\text{enc}}(\tilde{\mathbf{H}}_f)$$

and after that, a feedback channel is used to send the codeword to the BS. A decoder at the BS can recreate the CSI as:

$$\hat{\tilde{\mathbf{H}}}_f = f_{\text{dec}}(s)$$

The compression ratio can be obtained as:

$$\gamma = \frac{N_s}{2 \times N_i \times N_t}$$

where γ, N_s, N_i and N_t are compression ratio, codeword length, selected rows from $\tilde{H}$ and number of transmit antenna.

III. Trade of Between Accuracy and Resources

CNNs have demonstrated superior performance in CSI compression tasks compared to compressive sensing methods. While numerous architectures have been proposed, deploying these models on mobile hardware presents significant challenges. Consequently, it becomes imperative to strike a balance between accuracy and resource constraints when considering the practical application of CSI feedback compression on user equipment with limited resources. Most research in CSI feedback compression focuses on various designs of auto-encoders. A significant portion of the parameters and FLOPs in auto-encoder designs is attributed to the fully connected layer. Considering the CsiNet [6] architecture, initial measurements indicate a computational load of 5.6 million FLOPs and 2.1 million trainable parameters. Upon selective removal of the fully connected layers within the architecture, these metrics were reassessed, revealing a notable reduction. Specifically, the FLOPs decreased to 3.5 million, while the number of trainable parameters plummeted to 3400.

This analysis underscores a substantial optimization opportunity through the elimination of fully connected layers. Notably, the removal of these layers resulted in a reduction of approximately 37.5% in FLOPs, indicating a significant computational burden alleviation. Moreover, the parameter count experienced a staggering decrease of over 99.8%, illuminating the dominant role played by fully connected layers in parameter-heavy neural network architectures such as CsiNet [6].

Additionally, kernel size in CNNs structure stands out as a crucial factor influencing the computational complexity of CNNs within the CsiNet [6] architecture. Initially set at (3,3), altering the kernel size to (7,7) prompts a significant shift in computational requirements. Specifically, this adjustment results in a notable increase, with FLOPs soaring to 20.3 million and the number of parameters remaining at 2.1 million.

This shift underscores the profound impact of kernel size on computational demands within CNNs architectures. Notably, enlarging the kernel size to (7,7) yields a substantial rise in FLOPs, escalating by approximately 362%. Furthermore, the adjustment marginally affects the parameter count, increasing it by a modest1.44%.

These findings underscore the intricate interplay between kernel size and computational complexity in CNNs architectures such as CsiNet [6]. Such insights are valuable for optimizing network design to achieve desired performance outcomes while managing computational resources effectively.

CsiNet [6] achieves a parameter count under 1 million, while ACRNet [24] and CRNet [15] achieve superior Normalized Mean Square Error (NMSE) performance while maintaining a parameter count similar to CsiNet [6]. DS-NLCsiNet [18] achieves an NMSE near -17dB with nearly 10 million FLOPs and a parameter count similar to CsiNet. In contrast, Deep Decoder [25] and TransNet [26] have over 50 million FLOPs and 25% and 30% more parameters, respectively, compared to CsiNet (Figure 2 and 3).

Many researchers primarily focus on increasing accuracy without considering the practical deployment of their proposed architectures. However, beyond addressing the imperfect

channel, a challenge for evaluating the real performance of deep learning versus compressive sensing methods lies in considering storage constraints. Lu et al. [17] discuss the flexible deployment of their proposed architecture by adjusting kernel sizes for the convolutional neural network part of the auto-encoder and implementing network binarization for the fully connected layer. Lu et al. [17] also propose feature quantization for bitstream generation, replacing the feedback of floating-point numbers, which may not be feasible in digital communication systems.

As highlighted by Lu et al. [17], adjusting kernel sizes can enhance spatial correlation extraction, particularly not beneficial for CSI matrices characterized by inherent randomness due to wireless channel conditions. Furthermore, Lu et al.'s introduction[17] of network binarization represents a novel approach, marking the first application of binary neural networks [27] in this domain.

Many studies in this field prioritize optimizing the NMSE, overlooking the practical implications of models in real-world scenarios involving imperfect channels in wireless communication. Factors such as short-term channel correlation and resource constraints are frequently overlooked, despite their considerable dependency on model runtime. The runtime of most models is heavily reliant on the Graphics Processing Unit (GPU) of the hardware on which these models are evaluated, and this runtime is proportional to the FLOPs. Most proposed models exhibit higher NMSE when considering a snapshot of CSI and assuming zero mobility of User Equipment (UE). To address this gap, we evaluate several state-of-the-art proposed models using the QuadriGa dataset [22]. Since most models are trained on the COST2100 [7] channel model, it's essential to consider its implications for generalization to real-world scenarios. We generate CSI samples while considering the mobility of UE, and we examine snapshots at various time intervals to assess the generalization capabilities of the proposed models. We quantified the number of FLOPs and parameters required for User Equipment (UE) in Table I. Notably, most proposed models demand at least one million parameters, emphasizing the need to consider resource limitations and computational efficiency alongside NMSE optimization in model design. One of our proposed approaches to mitigate the high parameter count is to utilize Convolutional LatentSpace (CL), acknowledging that it may result in a slight loss of accuracy. If a significant decrease in accuracy occurs, it will demonstrate the dependence of these models on the fully connected network.

TABLE I: Methods Parameters

Methods	Params	
	UE Params	UE FLOP
CsiNet	1052K	1094k
CRNet	1049K	1235K
ACRNet-1x	1049K	1235K
CsiNetPlus	1049K	1462K
TransNet	2381K	1054K

A. Convolutional LatenSpace

Liu et al. [17] employed a convolutional latent space in lieu of a fully connected layer, a departure from conventional auto-encoder architectures. Typically, fully connected neural networks constitute a significant component of such architectures, dominating in terms of both FLOPs and parameter count. Notably, the accuracy of the model is heavily influenced by the weights of these fully connected layers. However, considering that the CSI matrix exhibits significant correlation primarily with adjacent elements, a substantial portion of the parameters within the fully connected layers may be redundant, contributing to excessive FLOPs and parameter counts. This redundancy is particularly problematic in digital communication systems, which often face constraints such as time delays and storage limitations. Thus, achieving high accuracy at the expense of an inflated parameter count is impractical in such contexts.

By substituting the fully connected layer with a convolutional latent space, a significant drop in NMSE of up to 50% was observed. We selected several of the proposed models that exhibited the highest accuracy and removed the fully connected network. Instead, we designed and trained a convolutional latent space network for them. We trained these networks alongside our proposed model from scratch with their own settings. The results are presented in Table III.

As shown in Table III, CLLWCsiNet achieves a performance of -15 dB, whereas CsiNet, with nearly 2 million parameters, achieves -17 dB. Without a fully connected layer, CsiNet's [6] accuracy drops to -9.72 dB. ACRNet1x [24] (with an expansion factor of 1) also has nearly 2 million parameters and achieves -27 dB, but without a fully connected layer, its performance decreases to -8.76 dB. It is evident that the high accuracy of both ACRNet1x [24] and CRNet [15] is heavily dependent on the presence of fully connected layers.

B. Proposed Model: CLLWCsiNet

The proposed model, CLLWCsiNet, introduces a novel approach to compressing and refining CSI feedback for MIMO systems, particularly under the constraints of 5G and 6G networks. This model utilizes convolutional latent-space to reduce parameters while maintaining predictive NMSE in noisy channels, making it suitable for resource-constrained user equipment.

Model Architecture: The CLLWCsiNet model is an extension of the CsiNet model with several key enhancements:

1) Convolutional Latent-Space: : This technique significantly reduces the number of parameters compared to traditional fully connected layers. This reduction is achieved by compressing the feature maps into a lower-dimensional space using convolutional layers, thus minimizing the computational overhead (Figure 4). The input CSI matrix has dimensions 2 $\times$ 32 $\times$ 32, which is passed through 3 different convolutional filters with kernel sizes of (7,7), (5,5), and (3,3). Following the convolutional factorization approach proposed in [15], the convolution kernel is split into two parts: (1,a) and (a,1), where 'a' is the kernel size. By factoring the convolution kernel in this manner, the model complexity is reduced while

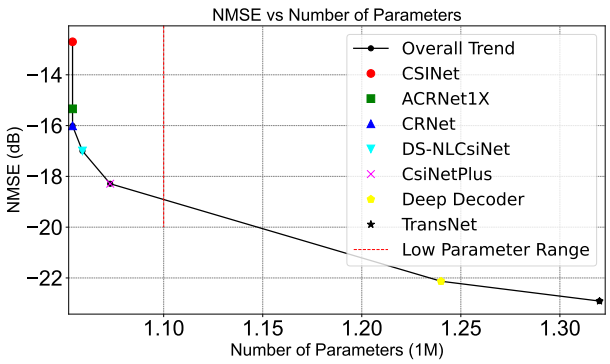

Fig. 2: NMSE vs Number of Parameters

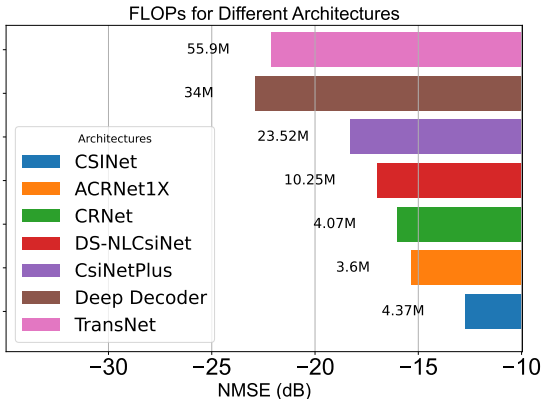

Fig. 3: NMSE vs FLOPs

TABLE II: Architecture Design for Different Compression Ratios

CR	Code word dimension	Convolution Filters	Kernel Sizes	Codeword dimension
4	512	64, 32, 16	$(1, 7)$	$16 \times 1 \times 32$
8	256	64, 32, 16, 8	$(1, 7)$	$8 \times 1 \times 32$
16	128	64, 32, 16, 4	$(1, 7)$	$4 \times 1 \times 32$

preserving the expressive power of the original convolution operation. This factorization technique has been shown to improve the efficiency of the neural network architecture without significantly sacrificing its performance.

2) Encoder Compression: After the initial passing of CSI matrix through the first part of the multi-resolution filtering architecture, the CSI matrix is reshaped from its original dimensions of $2 \times 32 \times 32$ to $64 \times 1 \times 32$. This reshaping is done through a set of selective compression filters. For a compression ratio of 8, the design selects a different set of filters, which can be switched based on the code. The compression ratio of 8 corresponds to a codeword dimension of $8 \times (1 \times 32)$. Similarly, for a compression ratio of 4, the codeword dimension becomes $16 \times (1 \times 32)$, and so on. As shown in Figure 5, the filtering process starts with 64 filters and progressively reduces the number of filters to 8, in two resolution passes. For a compression ratio of 4, the number of filters starts at 64 and is reduced to 16 through the multi-resolution filtering (Table II).

This multi-resolution filtering approach, along with the selective compression based on the target ratio, allows the model to efficiently encode the CSI matrix while preserving the necessary information.

3) Noise Resilience: : The model includes a mechanism to add Gaussian noise to the compressed CSI, simulating real-world noisy conditions. This addition helps in making the model robust against channel imperfections.

4) Denoising Capability: : The model incorporates a denoising network to remove the added noise. This network uses convolutional layers to clean the CSI feedback before it is further processed, ensuring high accuracy even under noisy conditions. In a symmetric manner, the convolution filters are also present in the decoder part of the architecture. Before the multi-resolution RefineNet, we reshape the CSI matrix to 2×32×32 (Figure 6).

5) Convolution Factorization: To prevent an increase in FLOPs while simultaneously enhancing accuracy, we utilize convolution factorization as introduced in CRNet[15]. Additionally, we employ different resolutions to make learning the random structure of CSI more effective. Large kernel sizes, as suggested in CsiNetPlus[16], are used to avoid futile coefficients and ensure coverage of the CSI that contains data. Conversely, CRNet[15] suggests smaller kernel sizes for learning granular features in the CSI matrix. We integrate both approaches into a comprehensive resolution block.

6) Training Settings: We trained the network for 500 epochs with a batch size of 200 and a learning rate of 0.001. The number of epochs should be carefully considered, as a high number of epochs can make the training process expensive and time-consuming for different datasets.

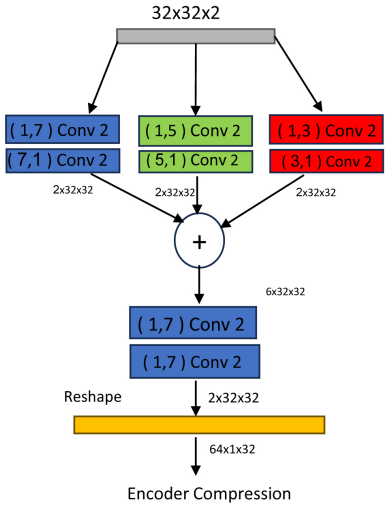

Fig. 4: Multiresolution Encoder

TABLE III: NMSE (dB) of Proposed Auto-Encoders Using Fully Connected Layers and Convolutional Latent Space for Indoor Scenario

CR	Architecture	With Conv-LatentSpace			With FC		
		FLOPs	Params	NMSE	FLOPs	Params	NMSE
4	CsiNet[6]	4.00M	17.9K	-9.72	5.41M	2103K	-17.36
	CRNet[15]	3.72M	17.62K	-7.35	5.12M	2103K	-26.99
	ACRNet[24]	3.16M	7.12K	-8.76	4.64M	2102K	-27.16
	CLLWCsiNet*	7.26M	70.26K	**-15.09**	/	/	/
8	CsiNet[6]	3.7M	10.71K	-3.56	4.37M	1054K	-12.70
	CRNet[15]	3.5M	10.43K	-4.34	4.07M	1054K	-16.01
	ACRNet[24]	2.93M	9.94K	-7.72	3.6M	1054K	-15.34
	CLLWCsiNet*	7.15M	66.7K	**-11.11**	/	/	/
16	CsiNet[6]	3.66M	7.12K	-2.19	3.84M	530K	-8.65
	CRNet[15]	3.38M	6.84K	-2.49	3.55M	530K	-11.35
	ACRNet[24]	2.81M	6.35K	-6.038	3.07M	529K	-10.36
	CLLWCsiNet*	6.63M	42.6K	**-8.62**	/	/	/

* / Indicates NMSE is not reported

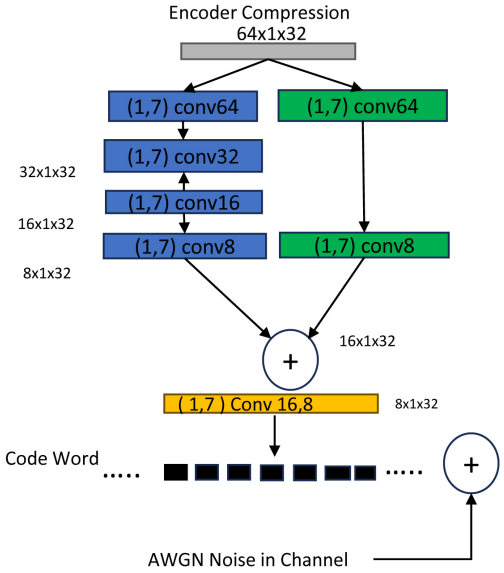

Fig. 5: Convolutional Laten-Space

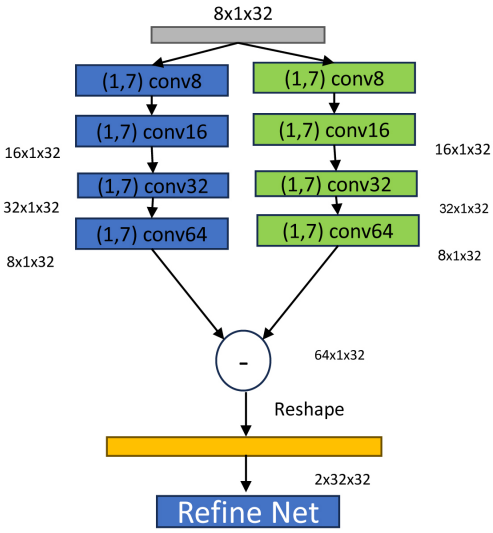

Fig. 6: Denoising Module and Multiresolution RefineNet

IV. Imperfect Wireless Channel

In addition to multipath fading and limited coherence time and frequency, noise is a crucial aspect of wireless channels that must be considered in channel state information feedback. Many proposed models assume a perfect channel in the reconstruction process, which does not reflect real-world conditions and may lead to reported results that cannot be achieved in practice.

Sun et al. [28] acknowledge the presence of imperfect channels. However, there exists a discrepancy between the imperfect channel and their suggested model. They do not account for the imperfect channel during the feedback transmission process. Instead, they attempt to mitigate noise from the channel state information matrix by employing a denoising module at the frontend before generating the codeword and prior to entering the decoder. This approach may not fully capture the complexities of real-world channel imperfections and could potentially lead to suboptimal performance in practical scenarios.

Ye and colleagues [23] address the challenge of imperfect channels by proposing the DNNet, drawing inspiration from DNCNN [29]. Their denoising module is designed to seamlessly integrate with various auto-encoders such as CsiNet [6], CRNet [15], and ACRNet [25]. The training process for the denoising module consists of two stages: pretraining and joint training.

During the pretraining stage, a new dataset is generated alongside COST2100. This dataset includes compressed codewords with and without additive Gaussian noise. Subsequently, in the joint training stage, the trained CsiNet and DNNet from the pretraining phase are connected. This combined CsiDNet+DNNet model outperforms both the standalone pretraining model and CsiNet in the presence of additive Gaussian noise.

The proposed algorithm of DNNet can be outlined as follows: DNNet comprises a Noise Extraction Unit (NEU),

designed to extract noise from the codeword. This noise is then subtracted from the noisy codeword using one input layer, one output layer, and $L-2$ hidden layers. Both the input and output layers maintain the same dimensions as the compressed codewords. Let suppose the generated codeword by encoder part of CsiNet is :

$$S = [s_1, s_2, s_3, ..., s_{cw}] \tag{1}$$

After adding additive Gaussian noise to the codeword the noisy codeword is as follow:

$$\tilde{S} = S + \mathbf{n} \tag{2}$$

$$\tilde{S} = [\tilde{s}_1, \tilde{s}_2, \tilde{s}_3, ..., \tilde{s}_{cw}] \tag{3}$$

And then this codeword is input layer of DNNet. The output of NEU is the noise that is extracted from codeword:

$$\tilde{n} = g_L(...g_1(\tilde{S})) \tag{4}$$

The structure of proposed DNNet is as follows for each *lth* layer:

$$\begin{cases} z & \text{if } l = 1 \\ \zeta(W_l z + b_l) & \text{if } 2 \leq l \leq L-1 \\ W_l z + b_l & \text{if } l = L \end{cases} \quad \text{(from [23])} \tag{5}$$

where $\zeta(x) = (1 + exp(-x))^{-1}$ is sigmoid activation function. However, it's worth noting that the proposed model significantly increases the parameter count of the networks, often reaching millions, which might not be suitable for all applications. Additionally, the training process is described as complex in two stages.The symmetrical structure of DNNet significantly increases the number of (FLOPs) and parameters by orders of millions due to its repetitive pattern. For instance, when targeting a compression ratio of 16, the pattern begins with a fully connected layer comprising 128 dimensions, followed by another fully connected layer with 1024 dimensions as a hidden layer. The proliferation of fully connected layers stands out as the primary factor driving up both FLOPs and parameters within the model. Consequently, the complexity of the training process and the sheer number of parameters render DNNet impractical for deployment in real-world wireless communication systems. As illustrated in Figure 7, we evaluated different proposed models under various ranges of noise power, from -5 dB to 40 dB. Our proposed model outperforms previous models when the noise power is nearly greater than the signal power. Beyond 20 dB, where the signal power exceeds the noise power, CRNet [15] and ACRNet1x [24] exhibit better performance relative to our proposed model. To evaluate the previous models, we employed federated learning techniques and augmented them with additional layers to apply noise. For federated learning, we utilized the pretrained models as well. The convolutional latent space demonstrates relative correlations across various noise power levels compared to other models. Even with approximately 70K parameters, our proposed model maintains excellent performance. The proposed CLLWCsiNet model introduces innovations enhancing

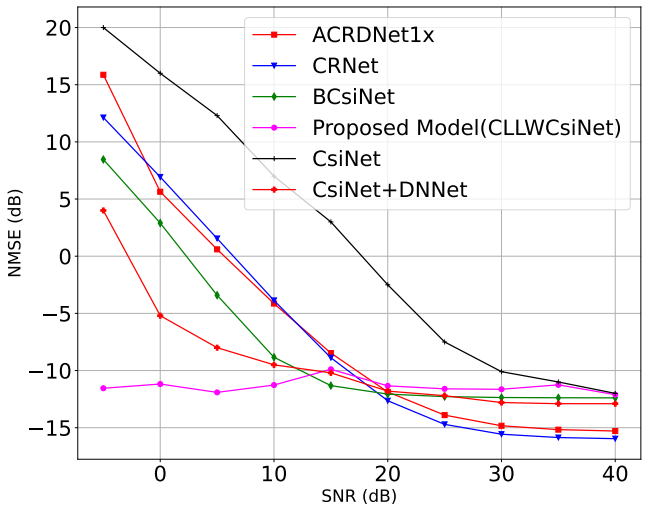

Fig. 7: Performance of Models under the various noise power and compression ratio of 8

noise resilience and predictive accuracy, notably achieving higher correlations between successive NMSE values, making it a standout for applications requiring reliable performance in dynamic wireless environments.

V. Model Generalization

Most research on CSI compression through deep learning employs the COST2100 channel model for both indoor and outdoor scenarios, specifically at 5.3 GHz for indoor and 300 MHz for outdoor scenarios. However, an important question arises: will these proposed models be successful on other channel models? [15] state that overfitting is rare in the construction of CSI feedback matrices, presenting two primary reasons. First, the number of FLOPs is relatively low compared to the large models used in image processing. Second, the structure of the CSI feedback matrix is unique. [24] discuss the random structure of CSI feedback, which complicates the learning process due to the lack of common patterns in the feedback.

To the best of our knowledge, this is the first time the proposed models have been tested on the Quadriga dataset for an outdoor scenario at 300 MHz and 28 GHz mmWaves. As shown in Table IV, state-of-the-art models fail to reconstruct the CSI feedback accurately, indicating potential overfitting of the proposed models on the COST2100 dataset. To address this, we employ the 3GPP 38.901 UMa NLOS channel model using the Quadriga software. We generate 20,000 samples to test the trained model through federated learning. To the best of our knowledge, this is the first research to test the proposed models on a different dataset to generalize the models. We separate the real and imaginary parts of the CSI matrices generated in the angular-delay domain.

We employed the 3GPP 38.901 UMa NLOS channel model at two different frequencies. We evaluated some state-of-the-art models on the mmWave 2.8 GHz frequency band using

TABLE IV: Performance Comparison for 3GPP 38.901 UMa NLOS Channel Model at CR=4 for Outdoor Scenario

Frequency	Model	NMSE (dB) Quadriga	NMSE (dB) COST2100
28 GHz	ACRNet1x[24]	-0.09	-
	CRNet[15]	1.22	-
	BCsiNet[17]	1.33	-
	CLLWCsiNet*	0.08	-
300 MHz	ACRNet1x[24]	0.55	-10.71
	CRNet[15]	1.31	-12.70
	BCsiNet[17]	1.61	-8.35
	CLLWCsiNet*	0.008	-7.89

* – Indicates NMSE is not reported
* ∗ Proposed Model

13,000 samples for the outdoor scenario. The second setting is at 5.3 GHz, with UE mobility of 0.9 meters per second, a setting similar to the COST2100 channel model for outdoor scenarios. This allows us to determine whether the proposed models are overfitting to the COST2100 channel model, an important consideration that has been largely neglected in most previous research. We use the saved checkpoints reported by some studies to generate their results.

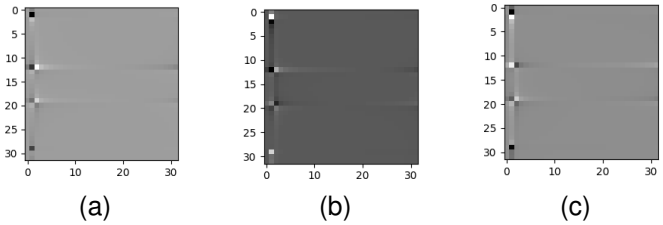

Fig. 8: CSI Feedback of Quadriga channel model for Outdoor scenario 28GHz.

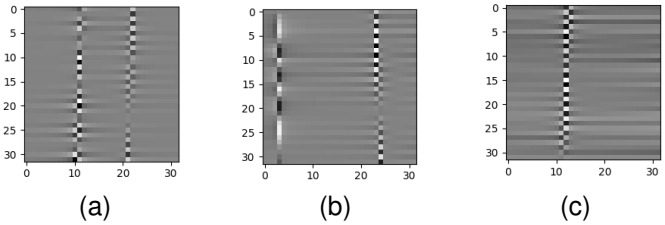

Fig. 9: CSI Feedback of Quadriga channel model for Outdoor scenario 300MHz.

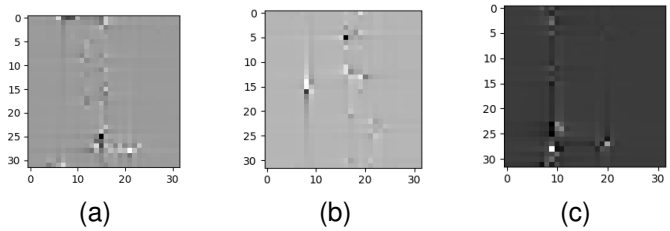

Fig. 10: CSI Feedback of COST2100 channel model for Outdoor scenario 300MHz.

The table IV illustrates a notable decline in performance among the models when evaluating the channel state information generated by Quadriga models. This observation suggests a clear instance of overfitting.

TABLE IV evaluates four models (ACRNet, CRNet, BCsiNet, CLLWCsiNet) for their suitability in outdoor communication systems operating at 28 GHz and 300 MHz. Based on NMSE for data sets (Quadriga, COST2100), CLLWCsiNet appears to outperform the other models at both frequencies, followed by ACRNet and CRNet while other models evaluate in perfect channel, CLLWCsiNet has been consider under 40db SNR.

VI. Model Quantization

Quantization is a crucial technique for optimizing deep neural networks (DNNs) to be deployed on resource-constrained devices, such as mobile phones and IoT devices. Each quantization approach offers distinct advantages and trade-offs. Neural network binarization is an extreme form of quantization where weights and activations are limited to binary values, typically -1 and 1, resulting in significant model size reduction and computational efficiency.

$$\omega_b = \text{sign}(\omega) \tag{6}$$

where ω are the original weights and ω_b are the binarized weights. This drastic reduction in precision requires specialized training methods to maintain accuracy. [27] Lu et al. [17] were pioneers in applying binary neural networks, and they later introduced vector quantization in the ACRNet[24] architecture. Their approach combined the binarization of fully connected layers with subsequent vector quantization to generate bit streams of 0s and 1s between the encoder and decoder. Convolutional quantization focuses on reducing the precision of weights and activations specifically in convolutional layers, which are prevalent in CNNs used for image processing.

$$\chi_q = \text{round}(\frac{\chi}{S}) + Z \tag{7}$$

where S is the scale factor and Z is the zero-point [31]. Static quantization, or post-training quantization, is performed on a pre-trained model using calibration data to determine scaling factors for weights and activations. This method reduces the precision of these elements, typically to 8-bit integers, and is suitable for deployment on fixed-resource devices with consistent input data[32]. Dynamic quantization

TABLE V: NMSE (dB) 8-bit Dynamic Quantization of Proposed Auto-Encoders for Indoor and Outdoor Scenario

CR	Architecture	Indoor		Outdoor	
		Params	NMSE	Params	NMSE
4	BACRNet1x[24]	71K	-14.20	71K	-7.03
	QACRNet1x	2632	-16.19	2632	-9.44
	BCsiNet[17]	1088K	-17.25	1088K	-8.35
	BACRNet10x[24]	91K	**-17.27**	91K	-8.78
	QACRNet10x	23K	-13.94	23K	**-12.23**
8	BACRNet1x[24]	38K	-11.52	38K	-5.52
	QACRNet1x	2632	-11.33	2632	-7.20
	BCsiNet[17]	547K	-12.39	547K	-6.26
	BACRNet10x[24]	58K	-14.96	58K	-6.63
	QACRNet10x	23K	**-16.66**	23K	**-8.57**
16	BACRNet1x[24]	7.12K	-2.19	21K	-2.92
	QACRNet1x	2632	-9.07	2632	-4.67
	BCsiNet[17]	276K	-8.99	276K	-4.11
	BACRNet10x[24]	42K	-11.7	42K	-4.63
	QACRNet10x	23K	**-12.19**	23K	**-5.75**

* Q Indicates dynamic quantization based architecture

involves quantizing weights during training and performing quantization of activations dynamically at inference time, allowing the model to adapt to varying input data distributions. This quantization process ensures that the model maintains flexibility and efficiency, adapting to different data patterns while reducing computational overhead. [33]

In communication systems, particularly for CSI feedback from user equipment (UE) to the base station (BS), the choice of quantization method is critical. Static quantization is generally preferred for CSI feedback due to its efficiency and consistency. It provides a reliable way to map floating-point CSI values to a lower bit-width representation, typically 8-bit integers, reducing the amount of data to be transmitted without significantly affecting accuracy. This method is particularly suitable for scenarios with consistent channel conditions, allowing for effective pre-calibration. Dynamic quantization, on the other hand, could be considered in environments with highly dynamic channel conditions where adaptability is crucial. This method allows for real-time adjustment of the quantization parameters based on the current channel state, potentially improving accuracy in dynamic environments. However, the additional computational complexity at the UE and potential latency issues need to be carefully managed. It should be noted that all previous works have merely converted the coded sequence produced by the encoder into binary form (zeros and ones) from the perspective of a telecommunications system designer. However, for the first time, we have utilized quantization to optimize neural networks by reducing their complexity, thereby improving evaluation time and enabling implementation on telecommunications systems with limited capacity. The NMSE reported in Table III reflects the practical accuracy of the proposed models. We extend quantization to the entire network, including both neural and convolutional networks, achieving a higher NMSE compared to ACRNet, even with a significantly lower number of parameters.

Quantization-aware training (QAT) and post-training quantization (PTQ) are two primary approaches to implement quantization. QAT integrates the quantization process into the training phase, allowing the model to learn and compensate for the quantization errors, which results in higher accuracy for the quantized model. This involves simulating quantization during both forward and backward passes of training. PTQ, however, applies quantization to a pre-trained model, often followed by techniques such as weight clustering and fine-tuning to mitigate the loss in accuracy.

Dynamic quantization is a valuable technique for applications requiring efficient use of hardware resources and fast real-time inference, such as in wireless communications. This paper introduces dynamic quantization for the first time in this context.

VII. Conclusion

In this work, we introduced CLLWCsiNet, a novel deep learning model tailored for compressing CSI feedback in MIMO systems under 5G and 6G network constraints. CLLWCsiNet utilizes convolutional latent-space compression to significantly reduce parameters while maintaining high predictive accuracy, making it suitable for resource-limited user equipment. By integrating noise resilience and a denoising network, our model demonstrates robust performance in noisy channel conditions, crucial for reliable CSI reconstruction. Evaluation across diverse channel models, including Quadriga datasets at various frequencies, highlighted CLLWCsiNet's superior performance over existing models like ACRNet and CRNet. This underscores its potential to generalize beyond the COST2100 dataset, mitigating overfitting concerns observed in previous studies. Additionally, we explored static and dynamic quantization techniques tailored for efficient deployment in wireless communication systems. Dynamic quantization emerged as a promising approach, adapting to dynamic channel conditions while maintaining computational efficiency. In conclusion, CLLWCsiNet represents a significant advancement in CSI compression for MIMO systems, offering a balanced solution of accuracy and efficiency in challenging wireless environments. Future research will focus on optimizing architectures for heterogeneous networks and refining quantization strategies for broader deployment scenarios.

REFERENCES

[1] Lu, L., et al. "An Overview of Massive MIMO: Benefits and Challenges," in IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 5, pp. 742-758, 2014.

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

[3] David James Love, undefined., et al. "An overview of limited feedback in wireless communication systems," in IEEE Journal on Selected Areas in Communications, vol. 26, 2008.

[4] Zhu, Z., et al, "Asymmetric Non-Local Neural Networks for Semantic Segmentation," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 593-602.

[5] P. Kuo, H. Kung, P. Ting, "Compressive sensing based channel feedback protocols for spatially-correlated massive antenna arrays," in 2012 IEEE Wireless Communications and Networking Conference (WCNC), 2012, pp. 492-497.

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

[7] Liu, L., et al. "The COST 2100 MIMO channel model," in IEEE Wireless Communications, vol. 19, pp. 92-99, 2012.

[8] He, K., et al, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[9] Tianqi Wang, et al. "Deep Learning-based CSI Feedback Approach for Time-varying Massive MIMO Channels," in CoRR, vol. abs/1807.11673, 2018.

[10] Lu, C., et al. "MIMO Channel Information Feedback Using Deep Recurrent Network," in IEEE Communications Letters, vol. 23, no. 1, pp. 188-191, 2019.

[11] X. Li, H. Wu. "Spatio-Temporal Representation With Deep Neural Recurrent Network in MIMO CSI Feedback," in IEEE Wireless Communications Letters, vol. 9, no. 5, pp. 653-657, 2020.

[12] Z. Liu, M. Rosario, Z. Ding. "A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback," in IEEE Transactions on Wireless Communications, vol. 21, no. 2, pp. 1214-1228, 2022

[13] M. Boloursaz Mashhadi, D. Gündüz. "Deep Learning for Massive MIMO Channel State Acquisition and Feedback," in Journal of the Indian Institute of Science, vol. 100, 2020.

[14] G. Jiajia, C. Wen, S. Jin. "Deep Learning-Based CSI Feedback for Beamforming in Single-and Multi-cell Massive MIMO Systems," in IEEE Journal on Selected Areas in Communications, 2021.

[15] Zhilin Lu, Jintao Wang, Jian Song. "Multi-resolution CSI Feedback with Deep Learning in Massive MIMO System," in ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1-6, 2019.

[16] Jiajia Guo, et al. "Convolutional Neural Network-Based Multiple-Rate Compressive Sensing for Massive MIMO CSI Feedback: Design, Simulation, and Analysis," in IEEE Transactions on Wireless Communications, vol. 19, pp. 2827-2840, 2019.

[17] Z. Lu, J. Wang, J. Song. "Binary Neural Network Aided CSI Feedback in Massive MIMO System," in IEEE Wireless Communications Letters,, 2020.

[18] Yu, X., et al. "DS-NLCsiNet: Exploiting Non-Local Neural Networks for Massive MIMO CSI Feedback," in IEEE Communications Letters, vol. 24, no. 12, pp. 2790-2794, 2020.

[19] Zhu, Z., et al, "Asymmetric Non-Local Neural Networks for Semantic Segmentation," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 593-602.

[20] Q. Cai, C. Dong, K. Niu, "Attention Model for Massive MIMO CSI Compression Feedback and Recovery," in 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019, pp. 1–5.

[21] J. Hu, L. Shen, G. Sun, "Squeeze-and-excitation networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[22] Burkhardt, F., et al, "QuaDRiGa: A MIMO channel model for land mobile satellite," in The 8th European Conference on Antennas and Propagation (EuCAP 2014), 2014, pp. 1274-1278.

[23] Hongyuan Ye, undefined., et al. "Deep Learning-Based Denoise Network for CSI Feedback in FDD Massive MIMO Systems," in IEEE Communications Letters, vol. 24, pp. 1742-1746, 2020.

[24] Lu, Z., et al. "Binarized Aggregated Network With Quantization: Flexible Deep Learning Deployment for CSI Feedback in Massive MIMO Systems," in IEEE Transactions on Wireless Communications, vol. 21, no. 7, pp. 5514-5525, 2022.

[25] Chakma, A., et al. "Deep Decoder CsiNet for FDD Massive MIMO System," in IEEE Wireless Communications Letters, vol. 12, no. 12, pp. 2073-2077, 2023

[26] Y. Cui, A. Guo, C. Song. "TransNet: Full Attention Network for CSI Feedback in FDD Massive MIMO System," in IEEE Wireless Communications Letters, vol. 11, no. 5, pp. 903-907, 2022.

[27] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, Tijmen Blankevoort (2021). A White Paper on Neural Network Quantization. ArXiv, abs/2106.08295.

[28] Markus Nagel, et al. "A White Paper on Neural Network Quantization," in ArXiv, vol. abs/2106.08295, 2021.

[29] Zhang, K., et al. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," in IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142-3155, 2017.

[30] Markus Nagel, undefined., et al. "A White Paper on Neural Network Quantization," in ArXiv, vol. abs/2106.08295, 2021.

[31] Y. Chenna. "Quantization of Convolutional Neural Networks: A Practical Approach," in International Journal of Recent Technology and Engineering (IJRTE), vol. 8, 2023.

[32] S. Han, H. Mao, and W. J. Dally, "Quantizing deep convolutional networks for efficient inference: A whitepaper," arXiv preprint arXiv:1512.02572, 2015.

[33] Y. Zhou, S. Liu, J. Wang, Y. Eiss, X. Huang, and E. Elsen, "Efficient 8-bit quantization of Transformer neural machine language translation model," arXiv preprint arXiv:2006.16669, 2020.