# A Markovian Model-Driven Deep Learning Framework for Massive MIMO CSI Feedback

Zhenyu Liu, Mason del Rosario, and Zhi Ding

arXiv:2009.09468v1 [eess.SP] 20 Sep 2020

*Abstract*—Forward channel state information (CSI) often plays a vital role in scheduling and capacity-approaching transmission optimization for massive multiple-input multiple-output (MIMO) communication systems. In frequency division duplex (FDD) massive MIMO systems, forwardlink CSI reconstruction at the transmitter relies critically on CSI feedback from receiving nodes and must carefully weigh the tradeoff between reconstruction accuracy and feedback bandwidth. Recent studies on the use of recurrent neural networks (RNNs) have demonstrated strong promises, though the cost of computation and memory remains high, for massive MIMO deployment. In this work, we exploit channel coherence in time to substantially improve the feedback efficiency. Using a Markovian model, we develop a deep convolutional neural network (CNN)-based framework MarkovNet to differentially encode forward CSI in time to effectively improve reconstruction accuracy. Furthermore, we explore important physical insights, including spherical normalization of input data and convolutional layers for feedback compression. We demonstrate substantial performance improvement and complexity reduction over the RNN-based work by our proposed MarkovNet to recover forward CSI estimates accurately. We explore additional practical consideration in feedback quantization, and show that MarkovNet outperforms RNN-based CSI estimation networks at a fraction of the computational cost.

*Index Terms*—Massive MIMO, CSI compressed feedback, deep learning, FDD

## I. INTRODUCTION

Massive MIMO wireless interface has been identified as a critical radio technology at the physical layer capable of substantially improving the bandwidth efficiency and delivering Gigabits/s services to many heterogeneous subscribers simultaneously. The efficacy of such massive MIMO downlink depends on the availability of accurate forward (downlink) CSI estimates at the base station (BS) for transmission precoding. Given the large number of antennas in massive MIMO and potentially broad bandwidth, such downlink CSI estimation and acquisition require substantial amount of feedback from each subscriber user equipment (UE). To support high mobility UEs in modern mobile wireless, timely feedback for time varying (i.e., fading) CSI estimation [1], [2] pose critical challenges. Frequent reporting of CSI for massive MIMO coverage would consume too much network bandwidth and UE power. The need for efficient CSI feedback in massive MIMO networks strongly motivates many research efforts aimed at downlink CSI compression, feedback, and reconstruction.

The problem of CSI feedback and reconstruction in massive MIMO has been an active research area in recent years. Traditional vector quantization and codebook-based methods reduce feedback overhead by quantizing the CSI at the UE side [3]–[6]. However, the feedback overhead grows with the

number of antennas, often requiring large amount of uplink bandwidth or low accuracy for practical massive MIMO wireless transmission. Compressive sensing (CS)-based approaches exploit the sparsity channel property in some domain to lower the CSI feedback overhead [7]–[9]. However, CS-based approaches often hinge on strong channel sparsity conditions not strictly satisfied in some domains. Moreover, iterative CS reconstruction methods may need a large amount of computation time to accurately recover downlink CSI estimates.

There has been a surging wave of interest in applying artificial neural networks for forward CSI estimation [10]–[13]. The popularity and versatility of deep learning (DL) have motivated a number recent works that explored deep neural networks (DNN) for downlink channel compression and recovery, particularly for massive MIMO wireless interface. Typically, these DNNs have utilized two prevailing architectures that are successful in other applications. First, Convolutional Neural Networks (CNNs), which have demonstrated state-of-the-art performance in image processing tasks [14], have been integrated in an autoencoder for CSI compression and recovery of a single snapshot [15]. Second, Recurrent Neural Networks (RNNs) have been further investigated to exploit temporal CSI coherent for feedback compression in massive MIMO systems [16]–[20]. RNNs can leverage hidden states through architectures such as long short-term memory (LSTM) cells to exploit the effect of past inputs.

Existing works have demonstrated that DNNs can provide efficient CSI feedback and reconstruction for time-varying MIMO channels [16]–[20]. However, important issues remain unresolved in at least the following two aspects:

1) **Complexity and Storage**: The number of parameters in the RNN layers for CSI compression and reconstruction of massive MIMO systems can be staggeringly large. For example, the RNN module can add $10^8$ additional parameters [16], which raises storage and computation concerns. A fully connected layer-based autoencoder has been proposed for the CSI feedback in time varying channel to reduce the computational complexity and required memory [20] . However, the accuracy is less favorable in comparison to [16]. While other works have investigated RNNs of reduced size [19], [20], the least computationally expensive of these models still requires $10^7$ parameters per snapshot. Also, the networks in [19], [20] suffer from the significant feedback performance drop when the compression ratio is small, since the networks have to use the same compression ratio in successive time slots and can not get the accurate prior information in the initial time slot. Despite the reported

success of "stacked" LSTMs, the minimum necessary depth of recurrent layers for CSI recovery accuracy has not been adequately evaluated. Considering the large RNN parameter count, performance improvement should be substantial to justify the memory overhead.

2) **Physical Insight**: The success of RNNs in areas such as video processing [21] and natural language processing (NLP) [22], [23] has stimulated their applications in forward CSI feedback and reconstruction. However, despite the apparent similarities in terms of a time series prediction, the physical nature of underlying CSI in massive MIMO is considerably different from those in video contents and image contents. Leveraging domain knowledge and physical characteristics on mobile wireless channels can be beneficial. For example, LTE frames (subframes) occupy 10ms (1ms) of airtime and permit CSI feedback intervals that are integer multiples of either. DNN-based CSI feedback and recovery should consider the practical constraint of how often such feedback can be transmitted and how CSI of fading channels would vary due to the Doppler effect.

In order to reduce computational complexity and model size, we seek to systematically exploit the physical channel characteristics such as the temporal coherence of forward CSI. Instead of training an RNN as a black box to learn and acquire the underlying CSI characteristics for compression, feedback, and recovery, we directly leverage the known channel coherence temporally by developing a simple but effective Markovian model driven differential CSI feedback framework MarkovNet to improve CSI recovery accuracy and reduce computational complexity. Spherical CSI feedback framework and enhanced CSI feedback network structure are proposed to provide the accurate prior information in the initial time slot. CNN-based CSI feedback networks are trained to further compress and recover the differential CSI effectively. We show that this simple MarkovNet can directly take advantage of the channel fading property to deliver much more efficient CSI compression and recovery.

This paper is organized as follows. Section II describes the massive MIMO system model commonly adopted in this and similar works. Section III presents two approaches to exploit CSI temporal coherence: RNNs and differential encoding. Section IV describes our proposed differential encoding-based CSI feedback framework, MarkovNet, as well as data pre-processing techniques to further improve CSI feedback accuracy for individual channel snapshots such as power-based spherical normalization. Section V introduces the proposed CNN-based dimension compression and decompression module for model size and complexity reduction. Section VI presents our experimental results, including computational analysis and performance under feedback quantization, for MarkovNet in comparison with a benchmark RNN-based network. Section VII concludes this manuscript.

## II. SYSTEM MODEL

### A. Forwardlink Channnel Estimation and Reconstruction

In this paper, we consider a massive MIMO BS known in 5G as gNB equipped with $N_b \gg 1$ antennas to serve a number of single-antenna UEs within its cell. We apply orthogonal frequency division multiplexing (OFDM) in downlink transmission over $N_f$ subcarriers.

To model the received signal of a UE, consider the $m-$th subcarrier at time $t$. Let $\mathbf{h}_{t,m} \in \mathbb{C}^{N_b \times 1}$ denote the channel vector, $\mathbf{w}_{t,m} \in \mathbb{C}^{N_b \times 1}$ denote transmit precoding vector, $x_{t,m} \in \mathbb{C}$ be the transmitted data symbol, and $n_{t,m} \in \mathbb{C}$ be the additive noise. Then the received signal of the UE on the $m-$th subcarrier at time $t$ is given by

$$y_{t,m} = \mathbf{h}_{t,m}^H \mathbf{w}_{t,m} x_{t,m} + n_{t,m}, \tag{1}$$

where $(\cdot)^H$ represents the conjugate transpose. The downlink CSI matrix in the spatial frequency domain at time $t$ is denoted as $\tilde{\mathbf{H}}_t = \left[ \mathbf{h}_{t,1}, ..., \mathbf{h}_{t,N_f} \right]^H \in \mathbb{C}^{N_f \times N_b}$.

Based on the downlink channel matrix $\tilde{\mathbf{H}}_t$, the gNB can determine the transmit precoding vector for each subcarrier. However, since the size of the CSI matrix $\tilde{\mathbf{H}}_t$ is $N_f \times N_b$, the UE's CSI feedback payload is large and consumes a staggering amount of uplink bandwidth in massive MIMO systems.

To reduce the feedback overhead, we first exploit the sparsity of CSI in a different projection space, the delay domain. Multipath effects cause short delay spreads, resulting in sparse CSI matrices in the delay domain [24]. With the help of 2D discrete Fourier transform (DFT), CSI matrix $\mathbf{H}_f$ in spatial-frequency domain can be transformed to be $\mathbf{H}_d$ in angular-delay domain using

$$\mathbf{F}_d^H \mathbf{H}_f \mathbf{F}_a = \mathbf{H}_d, \tag{2}$$

where $\mathbf{F_d}$ and $\mathbf{F_a}$ denote the $N_f \times N_f$ and $N_b \times N_b$ unitary DFT matrices, respectively. After 2D DFT of $\mathbf{H}_f$, most elements in the $N_f \times N_b$ matrix $\mathbf{H}_d$ are negligible except for the first $R_d$ rows that dominate the channel response [15]. Therefore, we can approximate the channel by truncating CSI matrix to the first $R_d$ rows. $\mathbf{H}_t$ is utilized to denote the first $R_d$ rows of matrices after 2D DFT of $\tilde{\mathbf{H}}_t$. Using $\mathbf{H}_t$ as a supervised learning objective, a DL based encoder and decoder, which is often referred to as an autoencoder, can be jointly trained and optimized to achieve efficient CSI compression and reconstruction as shown in Fig. 1(a). Several recent works that adopted this autoencoder structure [15] [25] have reported notable successes.

To allow gNB to track the time-varying characteristics of wireless fading channels, UEs need to periodically estimate and feed back instantaneous CSI with high power and bandwidth efficiency. Considering a time duration with $T$ successive time slots, the sequence of time-varying channel matrix is defined as $\{\mathbf{H}_t\}_{t=1}^T = \{\mathbf{H}_1, \mathbf{H}_2, \cdots, \mathbf{H}_T\}$.

### B. High Efficiency CSI Feedback Encoding

To reduce feedback overhead, temporal coherence of the radio fading channels can be exploited. Since RF channels of mobile UEs are governed by physical scatters, multi-paths, bandwidth, and Doppler effect, the fading CSI exhibits physically coherent characteristics including coherence time, coherence bandwidth, and coherence space. For mobile users, coherence time measures temporal channel variations and describes the Doppler effect caused by UE mobility. For most
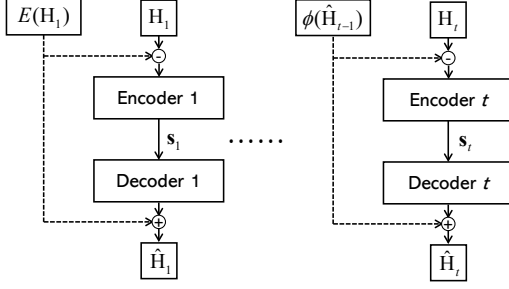
Fig. 1: Illustration of the temporal correlation based CSI feedback. ($t > 1$)



Fig. 2: (a) Illustration of a "stacked" LSTM network of depth 3 shown with recurrent connections. (b) Same network "unrolled" into $T$ timeslots. The network is trained with either perfect or quantized CSI, $\bar{\mathbf{H}}$, to generate CSI estimates, $\hat{\mathbf{H}}$.

application scenarios, the massive MIMO channels do not vary abruptly. By exploiting the channel coherence time, the UE and the gNB can rely on their previously stored CSI estimates to encode only the innovation components within the CSI. Specifically, the UE can encode and feed back CSI variations instead of the full CSI to substantially reduce feedback cost. Accordingly, gNB can combine the new feedback with its previously recovered CSI within coherence time to reconstruct subsequent CSI estimates.

We can adopt a general first order Markovian channel model

$$p(\mathbf{H}_t|\mathbf{H}_{t-1}, \cdots, \mathbf{H}_1) = p(\mathbf{H}_t|\mathbf{H}_{t-1}). \qquad (3)$$

Given knowledge of the CSI at the previous time slot, the minimum mean square estimation (MMSE) of $\mathbf{H}_t$ can be defined as

$$\phi(\mathbf{H}_{t-1}) = E\{\mathbf{H}_t|\mathbf{H}_{t-1}\}. \qquad (4)$$

We define the MMSE estimation error as

$$\mathbf{V}_t = \mathbf{H}_t - E\{\mathbf{H}_t|\mathbf{H}_{t-1}\} = \mathbf{H}_t - \phi(\mathbf{H}_{t-1}). \qquad (5)$$

Consider the scenario that, at time $t-1$, the UE and the gNB have successfully exchanged the CSI $\mathbf{H}_{t-1}$. Then it would be more efficient for the UE to compress and feed back the CSI estimation error $\mathbf{V}_t$ to the gNB instead of the raw $\mathbf{H}_t$.

Based on this CSI model, we can develop a novel DL encoder and decoder architecture by exploiting a trainable neural network to learn the unknown MMSE estimation function $\phi(\mathbf{H}_{t-1}) = E\{\mathbf{H}_t|\mathbf{H}_{t-1}\}$. This new DL encoder and decoder architecture is shown in Fig. 1.

As shown in Fig. 1, the feedback for the CSI matrix sequence can be divided into two phases: a) The feedback of CSI at the first (initial) time slot ($t = 1$) without prior information; b) The feedback of CSI in subsequent time slots ($t = 2, 3, ..., T$) given the prior CSI information. Denote $\hat{\mathbf{H}}_t$ as the reconstruction of CSI matrix $\mathbf{H}_t$ at time slot $t$. Define the encoding and decoding function as $f_e(\cdot)$ and $f_d(\cdot)$, respectively. For downlink CSI feedback architecture in the first time slot, the encoder network and decoder network can be denoted, respectively, by

$$\mathbf{s}_1 = f_{e,1}(\mathbf{H}_1 - E\{\mathbf{H}_1\}), \qquad (6)$$
$$\hat{\mathbf{H}}_1 = f_{d,1}(\mathbf{s}_1) + E\{\mathbf{H}_1\} \qquad (7)$$

This initial phase assumes that the CSI mean is known from training or past information. If such information is unavailable,
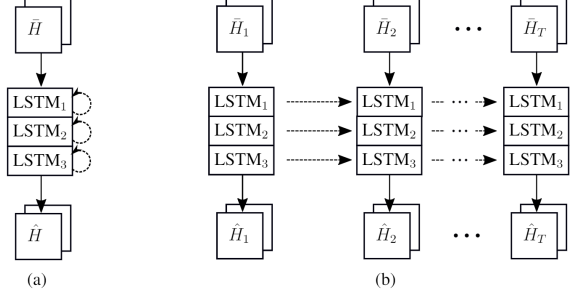
then $E\{\mathbf{H}_1\} = 0$ shall be applied. For downlink CSI feedback architecture of subsequent time slot $t$ ($t \geq 2$), the encoder network and decoder network can be executed, respectively, by

$$\mathbf{s}_t = f_{e,t}(\mathbf{H}_t - \phi(\hat{\mathbf{H}}_{t-1})), \qquad (8)$$
$$\hat{\mathbf{H}}_t = f_{d,t}(\mathbf{s}_t) + \phi(\hat{\mathbf{H}}_{t-1}) \qquad (9)$$

Since the optimum function $\phi(\hat{\mathbf{H}}_{t-1})$ is unknown, one direct solution is to approximate the function with deep neural network architecture trained by using a set of CSI samples.

## III. EXPLOITING CHANNEL TEMPORAL COHERENCE

We now discuss two avenues for exploiting the temporal coherence of CSIs at successive time-slots: a DNN architecture that utilizes long-short term memory (LSTM) layers and an information theoretic differential encoding approach.

### A. Recurrent Neural Networks

Recurrent neural networks (RNNs) include layers which encode memory of previous states. Through backpropagation training, recurrent layers learn whether to incorporate information stored in memory in the layer's output and whether that information should be kept in memory [26]. The memory incorporation enables RNNs to store, remember, and process information that resides in past signals for long time periods. RNNs can utilize past input sequence samples to predict future states. [27].

RNNs have found wide applications in areas such as natural language processing (NLP), including machine translation [22] and sentiment extraction [23]. For NLP tasks, empirical results have demonstrated the effectiveness of "deep" or "stacked" RNNs, networks which use the outputs of hidden recurrent layers as inputs to subsequent recurrent layers [28].

Prior works have investigated stacked RNNs for CSI estimation. Several proposals have favored the use of Long Short Term Memory (LSTM) cell [16]–[18], a recurrent unit that can tackle the vanishing gradient problem inherent in recurrent backpropagation [29]. Existing LSTM-based works in CSI estimation have assumed that stacked LSTMs are better than shallow LSTMs, presenting models which used LSTM cells of depth 3 [16]. Fig. 2 demonstrates the principle of this LSTM

network for CSI feedback and estimation. This bias towards deep RNNs is likely due to the aforementioned successes in NLP, where deep recurrent layers are theorized to learn hierarchical levels of semantic abstraction [23], [30].

This RNN approach has been recently proposed in [16]. In this work, we shall consider the proposed architecture of [16] as the benchmark method. However, deep LSTMs can be problematic, as the number of parameters per LSTM cell can be quite large. If a parsimonious model is desired due to memory constraints, then memory intensive RNNs can be very costly.

In this work, we explore ways to simplify the LSTM architectures without CSI performance loss, as deeper networks do not necessarily lead to better estimation accuracy. In fact, we shall show later (Fig. 11, Section VI-C) that a single LSTM layer (i.e., $D = 1$) could yield higher accuracy CSI estimates than deeper LSTMs (i.e., $D \in [2, 3]$) in some cases.

### B. CSI Entropy and Feedback Encoding

Despite the success of deep RNNs in CSI estimation and recovery, several important research questions remain.

- First, what simplifications can be made to reduce computational complexity while maintaining efficient CSI feedback and accurate CSI recovery?
- Second, how much CSI feedback bandwidth in terms of bitwidth per CSI coefficient is sufficient?
- Third, how frequently should a UE should provide CSI feedback for gNB to update its CSI estimate?

It is therefore important to tackle these open questions that hamper the practical application and efficacy of DL based CSI estimation and recovery in massive MIMO networks.

Consider random channel matrix $\mathbf{H}_t$ that consists of complex fading coefficients for the $t$-th timeslot. We denote its joint probability density function $p(\mathbf{H}_t)$ and define the corresponding entropy as

$$H(\mathbf{H}_t) = - \sum_{\mathbf{H}_t} p(\mathbf{H}_t) \log p(\mathbf{H}_t) \tag{10}$$

where (10) is the sum over all realizations of r.v. $\mathbf{H}_t$. The CSI entropy of (12) describes the required number of bits for the UE to feed back its CSI estimate to the gNB for reconstruction. Denote the $(i, j)$-th CSI element within $\mathbf{H}_t$ as $\mathbf{H}_{t,(i,j)}$ at time $t$. If all elements are independent, then we have a simple upper bound on the entropy of the full CSI matrix as

$$H(\mathbf{H}_t) \leq H_{\mathrm{UB}} = \sum_{i,j} H(\mathbf{H}_{t,(i,j)}) \tag{11}$$

This entropy bound $H_{\mathrm{UB}}$ describes the approximate number of total bits necessary for direct encoding of forwardlink CSI for UE feedback.

Fortunately, in mobile wireless networks, CSI within a coherence time exhibits strong correlation [31]. Therefore, instead of constructing CSI independently by relying on CSI feedbacks for individual time slots, the gNB can utilize this CSI dependency by leveraging both previously reconstructed CSIs and the current CSI feedback. In other words, the UE
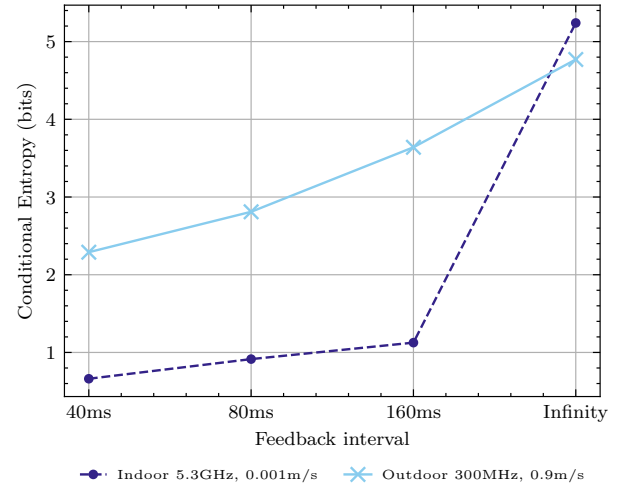


Fig. 3: Averaged conditional entropy (bits/element), under different feedback intervals ($\delta$).

feedback should focus on providing information that is not available at the gNB from CSIs of previous time slots.

Taking advantage of the Markovian CSI model, we can investigate how much the gNB can benefit from the previous CSI. Given the Markovian channel model of (3), the conditional CSI entropy quantifies the amount of information needed to characterize the CSI matrix based on the available CSIs from earlier reconstruction:

$$H(\mathbf{H}_t | \mathbf{H}_{t-1}, \ldots, \mathbf{H}_1) = H(\mathbf{H}_t | \mathbf{H}_{t-1}) \tag{12}$$
$$= - \sum_{\mathbf{H}_{t-1}} \sum_{\mathbf{H}_t} p(\mathbf{H}_t) \log p(\mathbf{H}_t | \mathbf{H}_{t-1})$$

From the well known relationship of $H(\mathbf{H}_t | \mathbf{H}_{t-1}) \leq H(\mathbf{H}_t)$, it is clear that by utilizing the most recently reconstructed CSI, the gNB would require less feedback bandwidth and improve the UE feedback efficiency.

A stationary first order Markovian channel model is characterized by the conditional probability density function of $p(\mathbf{H}_t | \mathbf{H}_{t-1})$. In practice, such distribution information on CSI is difficult to acquire analytically. To gain valuable insights into the time-coherence between CSI at different feedback intervals, we shall provide a numerical evaluation of typical wireless channel models by comparing the entropy and the conditional entropy of the forwardlink CSI parameters. Note the following relationship between CSI entropies at $t$ and $t-\delta$ where $\delta$ is the feedback interval:

$$H(\mathbf{H}_{t,(i,j)} | \mathbf{H}_{t-\delta}) \leq H(\mathbf{H}_{t,(i,j)} | \mathbf{H}_{t-\delta,(i,j)}) \leq H(\mathbf{H}_{t,(i,j)}). \tag{13}$$

For practical reasons, we shall numerically evaluate the conditional entropy of $H(\mathbf{H}_{t,(i,j)} | \mathbf{H}_{t-\delta,(i,j)})$ averaged over the coefficients in $\mathbf{H}_t$. Such information can present important guidelines to the determining number of bits for CSI feedback and how often UE should provide such CSI feedback for the CSI estimation of massive MIMO systems by the gNB.

In this experiment, we consider the link with $N_b = 32$ transmit antennas and 1 receive antenna over $N_f = 1024$

subcarriers. After applying the 2D DFT, $R_d = 32$ rows of significant CSI elements in delay domain are retained in $\mathbf{H}_t$. For each element within $\mathbf{H}_t$, we apply a 14-bit uniform quantizer to encode raw CSI values, resulting in a normalized mean square quantization error of -40dB. Since the complex CSI matrices were always divided into real part and imaginary part as the real-valued input to the neural network [15]–[20], we consider the conditional entropy of the CSI's real part and imaginary part individually. Fig. 3 demonstrates the estimated conditional entropy averaged over the $2 \times N_b \times R_d$ CSI elements.

We generate 10,000 random indoor and outdoor channel responses using the channel models given in [32] and [16]. Following the examples in [16], the indoor channel is in the 5.3 GHz band, with little or no mobility at velocity of $10^{-3}$ m/s. The outdoor channel is in the 300 MHz band, at velocity of 0.9 m/s. The bandwidth for indoor and outdoor channels is 20 MHz. The conditional entropy is evaluated for different lengths of feedback interval $\delta = 40$ms, 80ms, 160ms, and $\infty$ (i.e., no feedback).

As the results of Fig. 3, the average conditional entropy varies between 1 to 5 bits/element for both the indoor and outdoor channel models tested. As the duration of the feedback interval grows, the conditional entropy increases because of the limited channel coherence time. In addition, it is intuitive that the outdoor channel exhibits higher conditional entropy since higher velocity corresponds to shorter coherence time [33]. For both channel models, the average entropy of the CSI elements without prior CSI can be seen for $\delta = \infty$ which attends its maximum value of approximately 5 bits. These numerical results strongly motivate a systematic selection of feedback interval and feedback bandwidth. For example, for a feedback interval of 80 ms, an average of approximately 3 bits per CSI coefficient can be used for the outdoor CSI feedback by UEs when prior CSI is utilized by the gNB. On the other hand, for the same feedback interval, an average bitwidth of 1 bit per CSI coefficient can be used for indoor CSI feedback.

The reduction of CSI entropy under conditions of known prior CSI knowledge motivates the idea of condition-based encoding such as differential encoding by the UE. Encoding the difference between successive feedback instants, $\mathbf{H}_t$ and $\hat{\mathbf{H}}_{t-\delta}$, can reduce the required number of UE feedback bits, allowing more compression without loss of information [34].

## IV. DIFFERENTIAL CSI ENCODING

### A. A Simplified Markovian Model

Although the general Markovian CSI model motivates the use of a DNN to approximate the conditional mean $\phi(\hat{\mathbf{H}}_{t-1})$ through training and learning, we can examine simpler CSI models in order to develop a low complexity encoder-decoder structure with consistently strong performance. Consider the simplified Markovian CSI model of [35]:

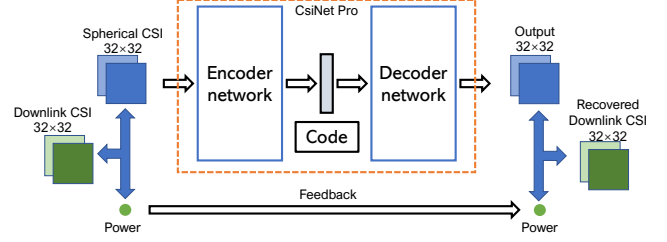$$\mathbf{H}_t = \gamma \mathbf{H}_{t-1} + \mathbf{V}_t \tag{14}$$



Fig. 4: Architecture of spherical CSI feedback in SphNet.

where $\gamma$ is a constant and $\mathbf{V}_t$ is a zero-mean i.i.d. random matrix. Given ensemble samples of $\mathbf{H}_{t-1}$ and $\mathbf{H}_t$, the unknown $\gamma$ can be estimated via

$$\hat{\gamma} = \frac{\text{Trace}(E\{\mathbf{H}_t \mathbf{H}_{t-1}^H\})}{E\|\mathbf{H}_{t-1}\|^2}. \tag{15}$$

Based on this simplified 1st order autoregressive (AR) model, we propose a low complexity encoder-decoder architecture for time slot $t$ ($t \geq 2$) as

$$\mathbf{s}_t = f_{e,t}(\mathbf{H}_t - \gamma \hat{\mathbf{H}}_{t-1}), \tag{16}$$
$$\hat{\mathbf{H}}_t = f_{d,t}(\mathbf{s}_t) + \gamma \hat{\mathbf{H}}_{t-1} \tag{17}$$

Based on this simplified model, we propose a differential encoding architecture named "MarkovNet" for efficient CSI feedback and reconstruction in the massive MIMO systems. The differential CSI feedback framework consists of two phases: a first network used for the initial CSI at $t_1$ (Spherical CSI feedback), and a second network in subsequent timeslots to compress and encode differential information as described by the encoder of (16).

To fully exploit the temporal CSI coherence, accurate CSI at the initial time slot $t_1$ is required to provide sufficient baseline information for the CSI feedback in subsequent time slots. To this end, our proposed framework shall apply CSI pre-processing and optimize the neural network structure. Specifically,

- We propose the spherical normalization for CSI pre-processing to the input data distribution to make the network's objective function more applicable to the commonly adopted accuracy metric, NMSE.
- We also optimize the CSI encoder-decoder to enhance CSI recovery accuracy.

### B. Transforming CSI Feedback in Spherical Coordinate

How to effectively apply DL techniques to exploit channel data properties and optimization objects remains an open research issue, as many existing DL based works mainly utilize the deep learning architectures and optimization functions successfully developed for other application areas. Direct adoption of DL architectures without customization for CSI data characteristics risks unsatisfactory performance. In particular, data processing methods and loss functions developed for computer vision may not be well suited for CSI encoding and reconstruction.

To begin, many existing DL-based CSI encoding-decoding schemes conveniently view the 2D MIMO channel matrix

$\mathbf{H}_t$ as akin to an image such that the normalized elements of the CSI matrix are utilized as image-like input to DL networks in both training and testing. However, the multipath fading MIMO channels exhibit unique special properties and probability distributions different from 2D image data.

Among other differences, images are represented as matrices of intensity pixel values. For color images, each color channel corresponds to a 2D matrix of pixel values that are unsigned integers, e.g., in the range between 0 and 255. By normalizing these pixels, there can be strong benefit in preparing the images as inputs of the DL model. However, unlike different images whose pixel values are mostly in the same order of magnitude, the dynamic range of CSI data can be much greater. For example, RF pathloss grows polynomially with distance between gNB and UE [36]. As a result, CSI of one UE can be different from CSI of another UE by several orders of magnitude, depending on their respective distances to gNB. A naive normalization can render CSIs of some UEs too small for the DL networks to respond to. Another different feature is that the baseband CSI parameters are complex values, consisting of both magnitude and phase, whereas image pixels are nonnegative real (with normalization).

In terms of learning objectives, several existing DL-based CSI feedback works adopted the loss function similar to image recovery for training the DL networks. Specifically, the objective is to minimize the mean square error (MSE):

$$\text{MSE} = \frac{1}{N} \sum_{k=1}^{N} \|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2, \tag{18}$$

where $k$ and $N$ are, respectively, the index and total number of samples in the data set, whereas $\|\cdot\|$ denotes the Frobenius norm. On the other hand, it is typical more meaningful in CSI estimation to apply the normalized MSE (NMSE)

$$\text{NMSE} = \frac{1}{N} \sum_{k=1}^{N} \|\mathbf{H}_k - \hat{\mathbf{H}}_k\|^2 / \|\mathbf{H}_k\|^2, \tag{19}$$

to assess the accuracy of CSI recovery at the gNB [37] and feedback efficiency [15], [16], [25]. By directly using MSE as the loss function, the DL networks would be biased toward the CSI accuracy of stronger MIMO channels.

In response to the domain-specific characteristics of data and objective in CSI estimation, we propose to use a spherical CSI data structure for feedback as shown in Fig. 4. The spherical CSI feedback architecture splits the downlink CSI matrix $\mathbf{H}_k$ into a magnitude value $p_k$ and a spherical downlink CSI matrix $\check{\mathbf{H}}_k$, where $p_k = \|\mathbf{H}_k\|$ is the size of the CSI matrix whereas the unit norm spherical CSI $\check{\mathbf{H}}_k = \mathbf{H}_k/\|\mathbf{H}_k\|$ represents remains on the surface of the unit hyper-sphere. The UE would encode and feedback both the size $p_k$ and the spherical CSI matrix $\check{\mathbf{H}}_k$ separately.

Spherical CSI feedback architecture presents numerical advantages. First, we can construct an encoder DL network that focuses on compressing and encoding the spherical CSI matrix $\check{\mathbf{H}}_k$. The size of the CSI would be directly sent back to the gNB separately since it contains little redundancy. Thus, even for CSI matrices of different magnitudes, they are equally important in training the encoder and decoder
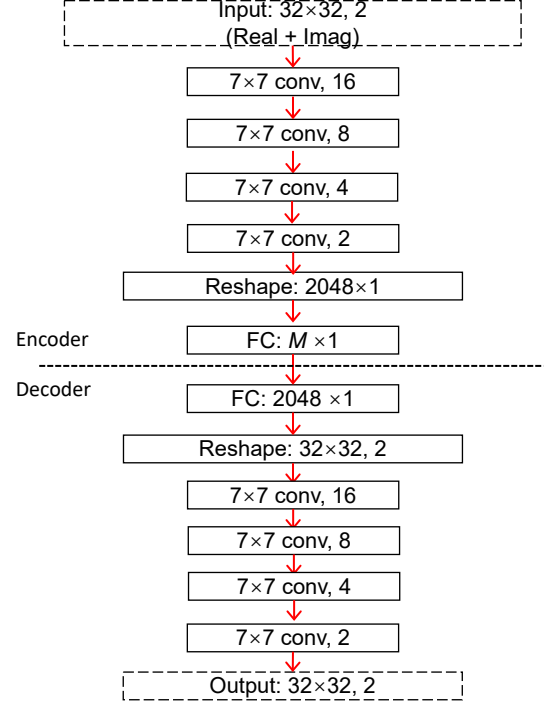


Fig. 5: Architecture of CsiNet Pro.

networks. During training the gradients for UEs that are far away from the gNB would no longer be negligible [38]. Moreover, the decoder will have a more limited domain for more accurate CSI recovery under spherical normalization [39].

As shown in Fig. 4, our joint CSI compression and reconstruction architecture still utilizes the effective autoencoder structure in which the encoder at the UE attempts to learn a low-dimensional CSI representation for a relatively high-dimensional dataset represented in the form of spherical CSI matrices. The decoder at the gNB reconstructs the CSI matrix based on feedback information extracted from the UE encoder and the direct feedback of CSI magnitude $p_k$.

### C. CsiNet Pro: A Novel CSI Encoder-Decoder Network

We propose an efficient neural network structure, named CsiNet Pro, for UE encoding and gNB decoding of CSI in massive MIMO networks. The structure of CsiNet Pro is illustrated in Fig. 5. In comparison with existing neural networks such as those from [15] [16], CsiNet Pro provides a deeper encoder that uses more convolutional layers to better extract features of CSI. There is a corresponding decoder at the gNB that also contains 4 convolution layers.

The design of encoder for dimension compression is crucial. However, the encoders in [15], [16], [25] all utilized one convolutional layer and one fully connected layer. As a major departure, the encoder of CsiNet Pro utilizes 4 convolutional layers for feature extraction and 1 fully connected layer for dimension compression. Specifically, the 4 convolutional layers apply $7 \times 7$ kernels to generate 16, 8, 4 and 2 feature maps, respectively (see Fig. 5).
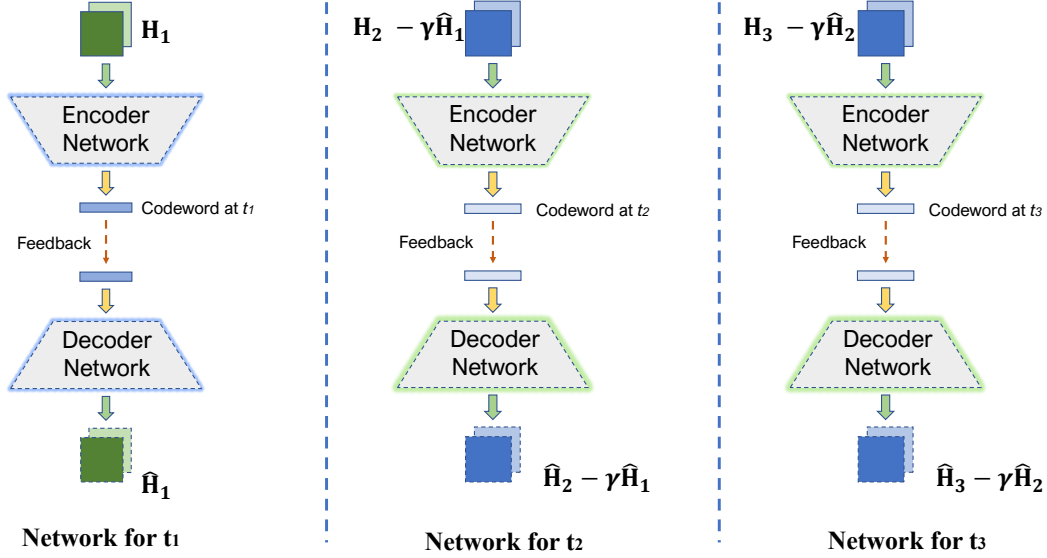
Fig. 6: Illustration of the multi-stage, differential CSI feedback framework MarkovNet.

Another major change in CsiNet Pro is the use of a different normalization range and output activation function. Recall that the decoder network utilizes 4 convolutional layers as shown in Fig. 5. Unlike the nonnegative pixel values in image reconstruction, CSI values contain both real and imaginary parts that can be either positive or negative. Thus, unlike previous works that normalize the CSI values to fall within $[0, 1]$ in order to use sigmoid or ReLU as the activation function of the last layer, our proposed CsiNet Pro normalizes the real and imaginary CSI values to the range [-1, 1] while using "tanh" as its activation function in the last layer.

We integrate the CsiNet Pro as part of the spherical CSI feedback framework shown in Fig. 4 to enhance the CSI recovery accuracy.

### D. Differential CSI Encoding

Motivated by the simplified first order AR model for CSI, we propose a differential CSI feedback framework MarkovNet to improve bandwidth efficiency. Different from the RNN based networks such as LSTM which relies on neural networks to learn the required information sharing and corresponding CSI compression simultaneously, MarkovNet proactively leverages the simplified AR model (14) for CSI and encode the CSI prediction error as shown in (16) between two successive time slots.

Recall that the difference based on first order estimation of the CSI in two adjacent time slots $\mathbf{H}_t - \hat{\gamma}\mathbf{H}_{t-1}$ is an approximation of the innovation $\mathbf{V}_t$. As shown in Fig. 6, for time-slots beyond the initial time-slot, the linear prediction difference $\mathbf{H}_t - \hat{\gamma}\mathbf{H}_{t-1}$ is sent to the encoder network to execute the encoding process of $\mathbf{s}_t = f_{e,t}(\mathbf{H}_t - \gamma\hat{\mathbf{H}}_{t-1})$ given in (16). At the gNB receiver, the decoder network can utilize the previously recovered CSI $\hat{\mathbf{H}}_{t-1}$ to reconstruct $\hat{\mathbf{H}}_t$ according to $\hat{\mathbf{H}}_t = f_{d,t}(\mathbf{s}_t) + \gamma\hat{\mathbf{H}}_{t-1}$ as described in (17).

MarkovNet from $t_2$ onward would employ the same network architecture CsiNet Pro as shown in Fig. 5. Compared with network for $t_1$ which uses a larger compression ratio to ensure the high recovery accuracy in the first timeslot, MarkovNet from $t_2$ can afford smaller compression ratio to achieve a higher bandwidth efficiency with the help of prior information.

MarkovNet exhibits several additional advantages in practical implementation. First, compared to RNN-based CSI feedback, MarkovNet can exploit pretrained model as initial neural network parameters for models used in later timeslots to improve training efficiency since the CSI at adjacent time slots share similar data features. Second, differential CSI matrix tend to be more sparse, hereby enabling MarkovNet to achieve a higher degree of compression during feedback. Third, for most wireless network applications, both gNB and UEs have limited power, computation, and storage resources. MarkovNet simplifies the learning tasks of neural networks and is more applicable in a wider variety of wireless deployment scenarios.

## V. MODEL REDUCTION

Practical implementation of deep neural networks for CSI feedback and recovery can be challenging to some mobile devices. Because DL network architectures often use large numbers of parameters, they require substantial computation and memory resources. Unrolled RNN models, such as the LSTM layers in Fig. 2 are particularly computationally expensive. For example, CsiNet-LSTM [16] at a compression ratio (CR) of $1/16$ contains $1.19 \times 10^8$ parameters per timeslot. One of the main advantages of MarkovNet (see Fig. 6) is its relatively low parameter count, as a comparable version of MarkovNet at a CR of $1/16$ has $5.43 \times 10^5$ parameters per timeslot, a difference of three orders of magnitude relative to CsiNet-LSTM.

Our proposed MarkovNet can clearly reduce the model size by eliminating the repeated structure used to learn the from the sequence data in RNN-style architecture. It is important to note, however, that the fully connected (FC) layers for dimension
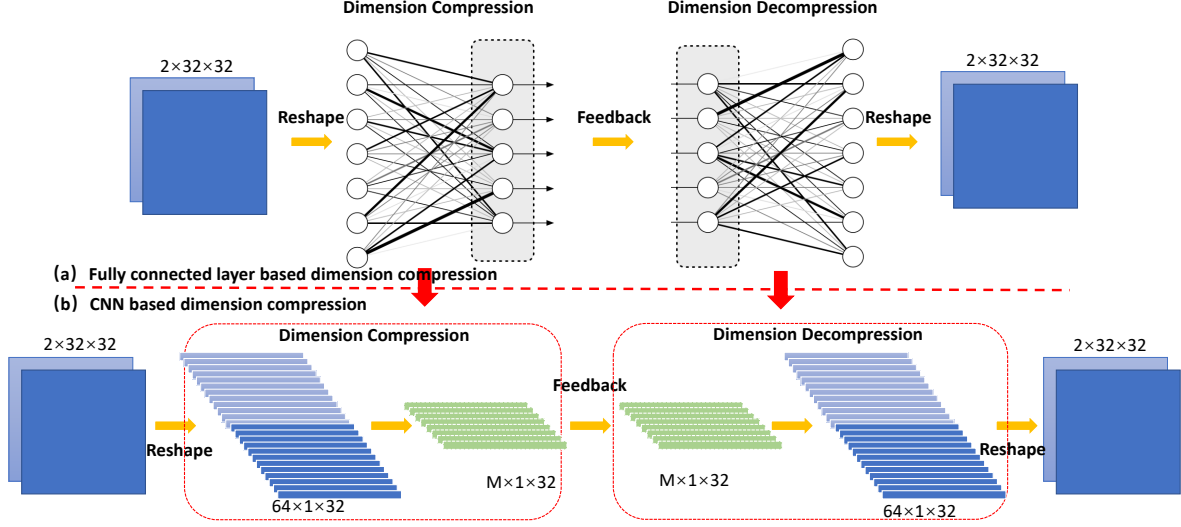
Fig. 7: Proposed CNN-based dimension compression module.

compression and decompression in the current MarkovNet still contains a large number of parameters. For example, there are more than $10^6$ parameters for the FC layers at CR = $1/8$.

The FC layers for dimension compression and decompression, as shown in Fig. 7(a), has often been adopted in deep learning based CSI feedback [15]–[18], [25]. However, elements of the CSI matrix only exhibit strong correlation with its neighbors in angular-delay domain. Thus, we recognize that the FC layers, though effective and popular, still contains a large fraction of non-essential connections with very weak weight parameters. This realization presents another opportunity for model reduction. To further reduce model size, we propose a CNN-based latent structure to replace the FC layers for dimension compression. As shown in Fig. 7(b), we slice the two square feature maps into $64$ feature maps of dimension $1 \times 32$. We then design $M$ CNN kernels of length $1 \times 7$ to compress the codewords dimension. The integer $M$ is adaptive in accordance with the encoder compression ratio denoted by $\frac{M}{64}$. Through this feature processing, connections between CSI elements that are far apart in the angular-delay domain are removed. Strongly correlated features of CSI matrix across the angular-delay domain can effectively be captured by the small CNN kernels.

TABLE I: Number of parameters and FLOPs comparison for FC-based and proposed CNN-based dimension compression and decompression module. M: million, K: thousand.

| | Number of parameters | | FLOPs | |
|---|---|---|---|---|
| | **FC-based** | **Proposed** | **FC-based** | **Proposed** |
| **CR=**$1/4$ | 2.1 M | 14.4 K | 4.2 M | 0.9 M |
| **CR=**$1/8$ | 1.1 M | 7.2 K | 2.1 M | 0.5 M |
| **CR=**$1/16$ | 0.5 M | 3.7 K | 1.0 M | 0.2 M |

To illustrate the effect of the proposed model size reduction, we summarize the number of parameters and the floating point operations (FLOPs) in Table I. This information provides a comparison of the storage size and computational complexity between the use of FC-layer and proposed CNN-layer in CSI compression module and the corresponding decompression module. As shown in Table I, the proposed CNN-based dimension compression and decompression module reduces the number of parameters by over 100 times and the number of FLOPs by at least 4 times. The comparison results demonstrate that our new CNN design for CSI compression and decompression represents an important step in broadening the range of practical applications for effectively deploying deep learning based CSI encoding, feedback, and reconstruction in massive MIMO wireless systems.

## VI. PERFORMANCE EVALUATION

We assess the performance of both RNN-based CsiNet-LSTM [16] and MarkovNet for two different massive MIMO scenarios generated from the well known COST 2100 model [32].

1) **Indoor** channels using a 5.3GHz downlink at 0.001 m/s UE velocity, served by a gNB at center of a 20m×20m coverage area.
2) **Outdoor** channels using a 300MHz downlink at 0.9 m/s UE velocity served by a gNB at center of a 400m×400m coverage area.

We give $N_b = 32$ antennas to the gNB to serve single antenna UEs randomly distributed within the coverage area. We use $N_f = 1024$ subcarriers and truncate the delay-domain CSI matrix to include the first $R_d = 32$ rows.

The gNB uses antennas arranged in a uniform linear array (ULA) with half-wavelength spacing. UEs are randomly positioned within the coverage area such that their CSIs are random. For each indoor/outdoor environment, we generate a dataset of $10^5$ sample channels and divide them into $7.5 \cdot 10^4$ and $2.5 \cdot 10^4$ for training and testing sets, respectively. The batch size for the training of MarkovNet is 200. MarkovNet at t1 was trained for 1000 epochs using MSE as the loss
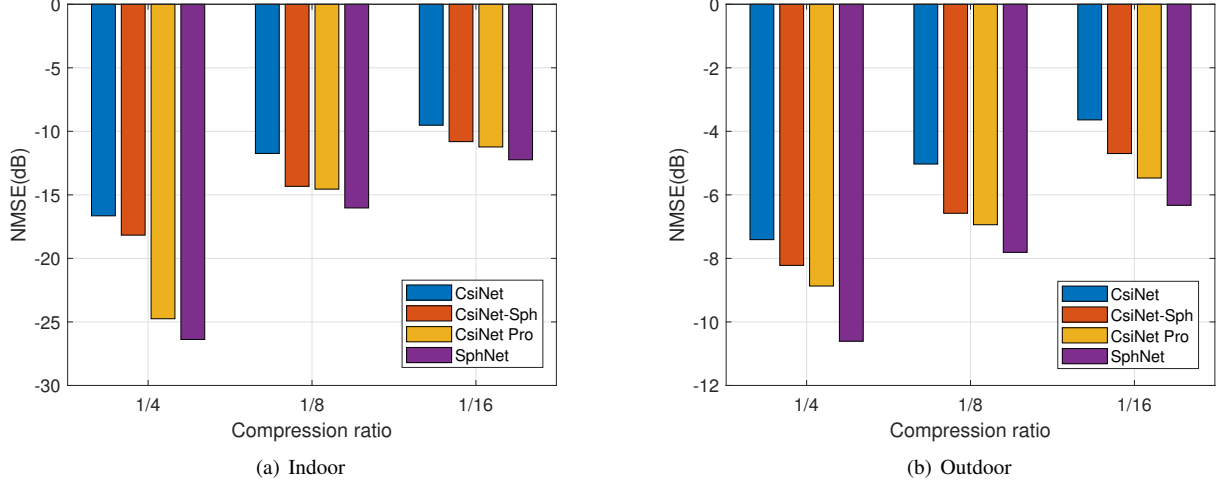
(a) Indoor

(b) Outdoor

Fig. 8: NMSE of different networks in the first time slot of MarkovNet over varying compression ratios (CR).



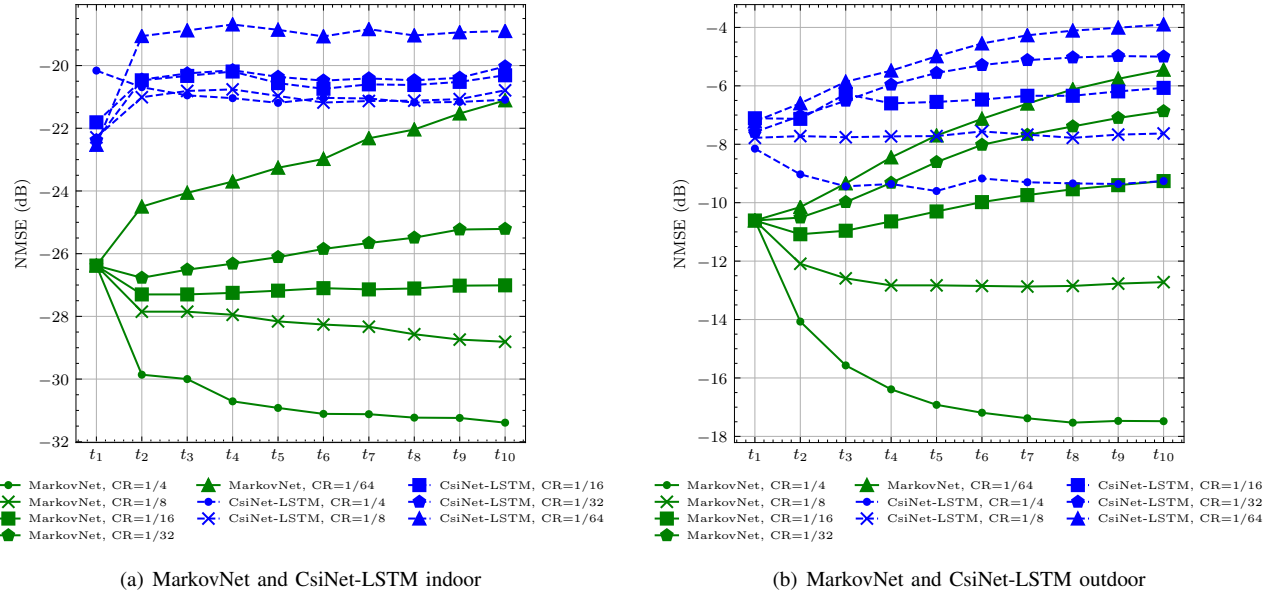(a) MarkovNet and CsiNet-LSTM indoor

(b) MarkovNet and CsiNet-LSTM outdoor

Fig. 9: NMSE comparison between MarkovNet and CsiNet-LSTM over varying compression ratios (CR).

function. For the MarkovNet after $t_2$, only 150 epochs are used with the help of initialization using the pretrained model of the previous time slot to reduce training expenses. We utilize the Adam optimizer with default learning rate $10^{-3}$, and hyperparameters (i.e., batch size, epochs) for each test will be clarified in each relevant subsection. NMSE is used to compare the CSI recovery accuracy of different networks.

### A. MarkovNet

In this part, we evaluate the performance of MarkovNet considering the performance at the first timeslot ($t_1$), the overall performance of MarkovNet, and the performance of MarkovNet-CNN.

*1) Performance evaluation at $t_1$:* To enable efficient differential CSI feedback, high accuracy CSI feedback is required at $t_1$ to provide a good starting CSI condition for subsequent

timeslots. Here, we demonstrate that our proposed spherical CSI feedback framework improves the CSI recovery accuracy for a single time slot compared to different CSI feedback frameworks.

Fig. 8 compares the performance of channel reconstruction from the use of CsiNet, CsiNet-Sph (CsiNet with the help of spherical feedback), CsiNet Pro, and SphNet (CsiNet Pro with the help of the spherical feedback framework). As shown in Fig. 8, SphNet achieves the best performance in single shot feedback for CSI recovery without relying on prior CSI knowledge, which means that SphNet can improve the accuracy of prior information for the MarkovNet. On the one hand, CsiNet Pro outperforms the CsiNet in different CR and scenarios, which means the enhanced network structure is effective. On the other hand, we can observe that spherical feedback can provide the most noticeable performance gain
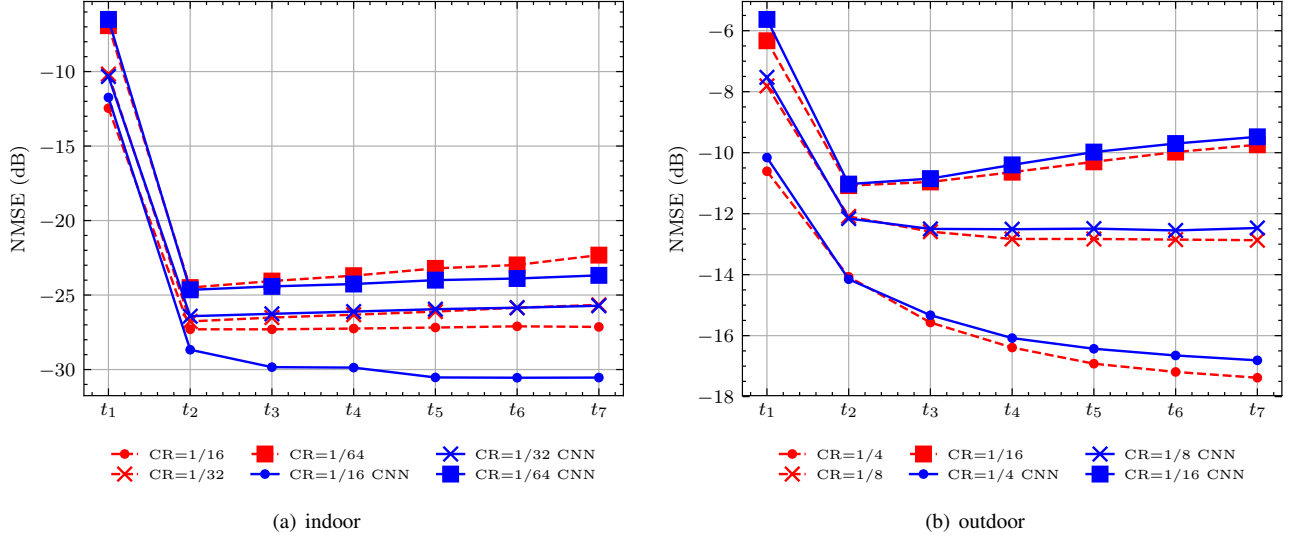
(a) indoor

(b) outdoor

Fig. 10: NMSE comparison between MarkovNet and MarkovNet-CNN over varying CR.

to both CsiNet and CsiNet Pro. This establishes the strength of spherical normalization to efficiently capture the CSI data feature.

*2) Overall performance evaluation of MarkovNet:* Every instance of MarkovNet contains two different compression ratios. For the first time slot, we initialize MarkovNet with CR=$1/4$ at $t_1$ to provide an accurate CSI baseline for subsequent time slots. For the rest timeslots, MarkovNet maintains the same CR for all subsequent timeslots ($t_2$ to $t_{10}$). To evaluate MarkovNet's performance under different amounts of compression, we vary the second CR used in the later timeslots from $1/4$ to $1/64$ and train each network. For example, in the Fig. 9 that follows, "MarkovNet, CR=$1/16$" uses CR=$1/16$ at timeslots $t_2$ through $t_{10}$ and CR=$1/4$ at timeslot $t_1$.

Fig. 9 compares the performance between MarkovNet and CsiNet-LSTM. The benefit of differential CSI encoding is demonstrated by the CSI recovery accuracy of MarkovNet using different compression ratios beyond $t_2$ in comparison with CsiNet-LSTM. MarkovNet consistently achieves higher CSI accuracy than CsiNet-LSTM at every CR level. With the help of differential CSI encoding, MarkovNet is an effective encoding framework given limited UE power and bandwidth for CSI encoding. For the indoor channels, MarkovNet can deliver reliable CSI accuracy of -30dB even for the compression ratio of $1/64$, which is a 10dB improvement over CsiNet-LSTM. Although the outdoor scenario continues to be more challenging, our results show that 1/4 or 1/8 compression ratio can achieve NMSE of $-17$dB and $-12$dB, respectively. On the other hand, CsiNet-LSTM is shown to provide NMSE only at $-9$ and $-7.5$dB, respectively.

*3) Performance and Complexity Trade-off of MarkovNet-CNN:* Fig. 10 shows the performance comparison between the MarkovNet and MarkovNet-CNN at different meaningful compression ratios. Since the trend of CSI accuracy is similar over time, we focus on the performance from $t_1$ to $t_7$. For the first time slot, we initialize MarkovNet and MarkovNet-CNN with CR=$1/4$ at $t_1$ to provide an accurate CSI baseline for sub-

sequent time slots. Note that, to show the influence of CNN-based dimension compression module at $t_1$, the results we shown in Fig. 10 at $t_1$ are from the labeled compression ratios. Both MarkovNet and MarkovNet-CNN achieve comparable CSI accuracy at $t_1$, indicating that CNN layer for compression and decompression is not only more efficient in memory and computation, but also delivers similar CSI accuracy. For the rest timeslots, MarkovNet maintains the same CR for all subsequent timeslots ($t_2$ to $t_7$). Beyond $t_2$, MarkovNet-CNN achieves modestly higher accuracy for indoor channels as shown in Fig. 10(a). The benefit of MarkovNet-CNN likely arises from the reduction of many redundant weights from the FC layer such that there are fewer opportunities for local minimum convergence. For outdoor channels, MarkovNet-CNN achieves comparable CSI accuracy as MarkovNet for compression ratio of $1/8$ and $1/16$ while exhibiting a modest loss of accuracy at CR=$1/4$. One possible reason is that outdoor channels can benefit more from higher number of connectivity in layers for compression and feature extraction because of their much more complex characteristics.
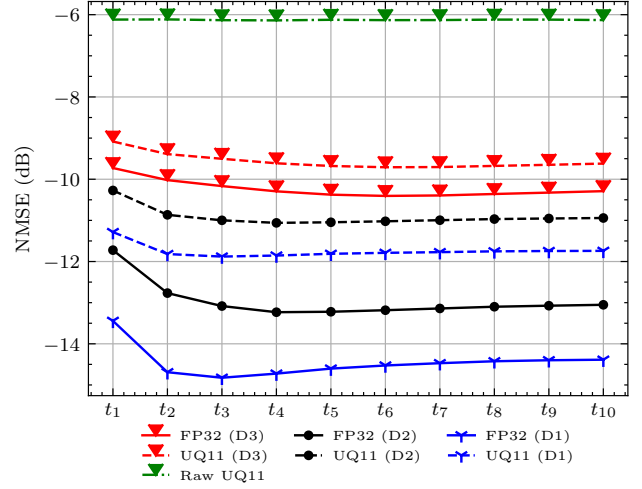
### B. Model size and Computational Complexity

We demonstrate that latent convolutional layers require significantly fewer parameters than FC-layers without loss of performance. Table II compare the model size and computational complexity (respectively) of CsiNet-LSTM, MarkovNet, and MarkovNet-CNN associated with a single timeslot. Across all compression ratios, MarkovNet uses 60 times fewer parameters than CsiNet-LSTM. More importantly, MarkovNet-CNN can use 1/3000 the number of parameters needed by CsiNet-LSTM while achieving better CSI recovery accuracy.

Table II also shows the average number of floating point operations (FLOPs) associated with a single timeslot for each network [40], [41]. MarkovNet and MarkovNet-CNN can save computation load by more than $\frac{8}{9}$ and $\frac{9}{10}$ FLOPs, respectively, when compared with the CsiNet-LSTM in each compression ratio. For CsiNet-LSTM, the amount of computation does
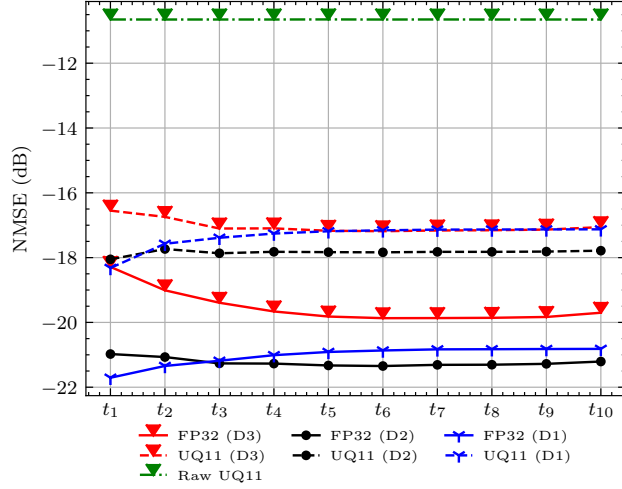
(a) Sweep depth for quantized/perfect CSI (Indoor)



(b) Sweep depth for quantized/perfect CSI (Outdoor)

Fig. 11: Stacked LSTMs for 3 depths trained on perfect and quantized CSI trained on Indoor 5.3GHz network (a) and on Outdoor 300MHz network (b).

.

TABLE II: Model size and computational complexity of tested networks. M: million, K: thousand.

| | Parameters | | | FLOPs | | |
|---|---|---|---|---|---|---|
| | CsiNet-LSTM | MarkovNet | MarkovNet-CNN | CsiNet-LSTM | MarkovNet | MarkovNet-CNN |
| CR=$\frac{1}{4}$ | 132.7 M | 2.1 M | 34.9 K | 412.9 M | 44.5 M | 41.2 M |
| CR=$\frac{1}{8}$ | 123.2 M | 1.1 M | 27.8 K | 410.8 M | 42.4 M | 40.7 M |
| CR=$\frac{1}{16}$ | 118.5 M | 542.9 K | 24.2 K | 409.8 M | 41.3 M | 40.5 M |
| CR=$\frac{1}{32}$ | 116.1 M | 280.7 K | 22.4 K | 409.2 M | 40.8 M | 40.4 M |
| CR=$\frac{1}{64}$ | 115.0 M | 149.6 K | 21.5 K | 409.0 M | 40.5 M | 40.3 M |

not change significantly even at low compression ratios. For example, a 16-fold drop in compression ratio (from 1/4 to 1/64) only results in a 1% saving of FLOPs. In contrast, MarkovNet and MarkovNet-CNN require much lower computational complexity in proportion at lower compression ratios. In MarkovNet and MarkovNet-CNN networks, for example, a 16-fold CR reduction (from 1/4 to 1/64) reduces the number of FLOPs by 9% and 2%, respectively.

We note that when deploying MarkovNet and MarkovNet-CNN as a cooperative learning mechanism at both UE and gNB, 50% additional parameters and FLOPs are required in comparison with the training phase. This is because the trained decoder must be duplicated at the UE side to generate the decoded CSI for the previous time slot used by the encoder. Despite this additional cost, both MarkovNet and MarkovNet-CNN still can reduce the number of parameters by orders of magnitude, and save over $\frac{5}{6}$ FLOPs in comparison with CsiNet-LSTM.

### C. LSTMs for CSI Estimation

In this section, we explore the effects of varying LSTM depth on network performance. For all experiments, we use the Adam optimizer with learning rate of $10^{-3}$ and a batch size of 100. For the LSTM-only networks in the ablation

TABLE III: Average NMSE across ten timeslots ($T = 10$) for stacked LSTMs of increasing depth trained on quantized CSI under 11-bit uniform quantization ('UQ11' in Fig. 11). NMSE of quantized CSI under uniform quantization ('Raw UQ11') is shown for comparison.

| Environment | Raw UQ11 | Depth 1 | Depth 2 | Depth 3 |
|---|---|---|---|---|
| Indoor | -10.66 dB | -21.03 dB | -21.26 dB | -19.54 dB |
| Outdoor | -6.41 dB | -14.64 dB | -13.11 dB | -10.40 dB |

study, we train for 500 epochs. For CsiNet-LSTM, we pretrain CsiNet at each compression ratio for 600 epochs, and we then initialize the CsiNet at each timeslot of CsiNet-LSTM with the pretrained weights before training for 500 epochs.

*1) Ablation study on LSTM depth:* We seek to know whether shallow RNNs perform comparably well to deep networks. To investigate the effect of network depth, we train stacked LSTMs of increasing depth on $\{\bar{\mathbf{H}}_t\}_{t=1}^{10}$, which are CSI matrices quantized with two different schemes: 1) Single-precision floating point (FP32) and 2) 11-bit Uniform Quantization (UQ11). For a visual depiction of this network, see Fig. 2(b). We train each network with Adam using the default
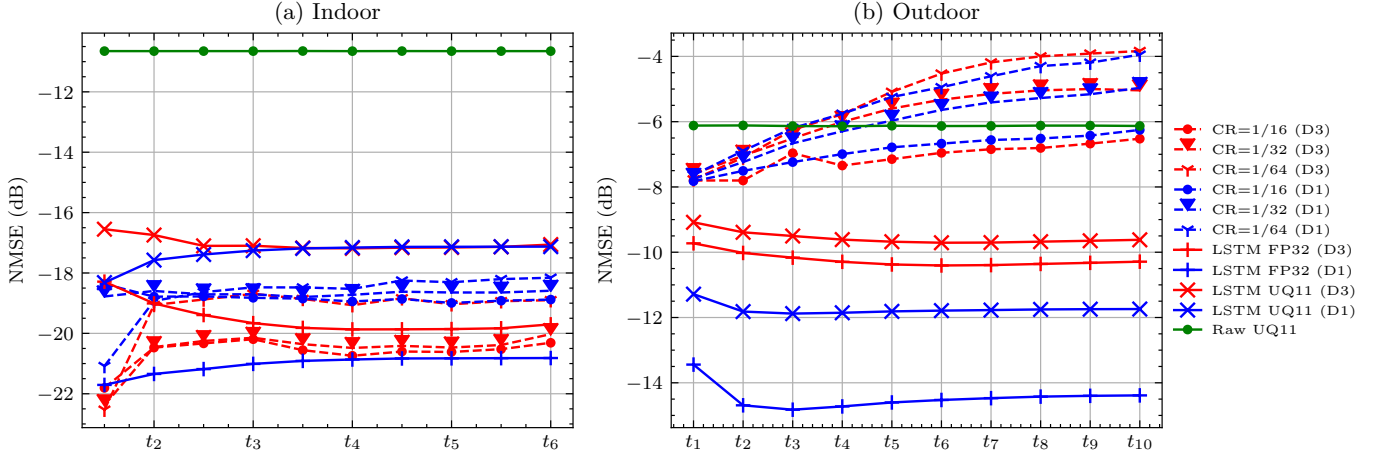
Fig. 12: CsiNet-LSTM over varying compression ratios (CR) compared to LSTMs trained on perfect CSI and quantized CSI. D$N$ indicates a stacked LSTM depth $N$.

Figure 11 shows the NMSE per timeslot for each of these RNNs, and the average performance across all timeslots is shown in Table III. At all depths, LSTMs are able to improve the test NMSE relative to quantized CSI. However, there is not a clear linear relationship between network depth and NMSE performance. For the outdoor network (Fig. 11(b)), the network performance and LSTM depth appear negatively correlated – increasing depth results in decreasing performance. For the indoor network, the best network has a depth of $D = 2$, indicating the the best choice of depth is channel-dependent.

*2) LSTM Depth in CsiNet-LSTM:* While LSTMs can perform admirably when using noisy CSI samples, these samples were not subject to compression. Compression is imperative for channel feedback, as transmitting uncompressed CSI will consume an undue amount bandwidth.

In this experiment, we use a known CNN/RNN for CSI estimation, CsiNet-LSTM [16]. Fig. 12 illustrates the NMSE for different depths of CsiNet-LSTM in each of the 10 time slots. The original network utilizes LSTMs of depth 3 (D3 in Figures 12(a), 12(b)), and we also train CsiNet-LSTM with one LSTM layer (D1) for comparison. We show the performance of the D3 and D1 networks to LSTMs trained directly on FP32 and UQ11 CSI samples (i.e., the same networks in Fig. 11. We train each network end-to-end based on the original paper's hyperparameters and dataset splits.

Figure 12(a) shows the NMSE in channel reconstruction by CsiNet-LSTM for the indoor dataset. While the shallower D1 network with fewer parameters in fact outperformed the deeper D3 network in the FP32 and UQ11 scenarios, the D3 variant of CsiNet-LSTM performs better than the D1 version for the all three compression ratios.

This performance trend relative to LSTM depth does not hold for the outdoor network. Figure 12(b) shows the NMSE in channel reconstruction for the outdoor dataset. Clearly, the shallower D1 network performs similarly to the D3 network at the tested compression ratios.

These results indicate that a simple, shallower LSTM can perform similarly to complex, deeper networks for both indoor and outdoor datasets.

TABLE IV: MarkovNet and CsiNet-LSTM mean NMSE degradation (increase) under different feedback quantization bits. The mean is taken across all tested compression ratios, and the degradation in NMSE is relative to floating point 32 bit precision.

| Network | Environment | 6 bits | 4 bits |
|---|---|---|---|
| MarkovNet | Indoor | 0.70 dB | 5.49 dB |
| | Outdoor | 0.03 dB | 0.58 dB |
| CsiNet-LSTM (D3) | Indoor | 2.30 dB | 11.96 dB |
| | Outdoor | 0.07 dB | 1.25 dB |
| CsiNet-LSTM (D1) | Indoor | 2.44 dB | 11.55 dB |
| | Outdoor | 0.30 dB | 1.21 dB |

*D. Network Performance Under Feedback Quantization*

To understand the effect of feedback quantization, we apply $\mu$-law companding to the encoded layer of both tested networks. $\mu$-Law companding uses a logarithmic transformation that emphasizes lower magnitude samples. For signal value $x$, the compression portion of the $\mu$-law scheme is written as

$$f(x) = \frac{\text{sgn}(x) \ln(1 + \mu|x|)}{\ln(1 + \mu)}, \ 0 \le |x| \le 1. \quad (20)$$

Uniform quantization is applied to the compressed signal. For signal value $x$, the quantization/dequantization operation produces a value $\hat{x}$, which can be written

$$\hat{f}(x) = \Delta \left\lfloor \frac{f(x)}{\Delta} \right\rceil \quad (21)$$

for fixed step size $\Delta$. After the quantized feedback is received, then we expand the result using the inverse of (20),

$$F(\hat{x}) = \frac{\text{sgn}(\hat{f}(x))((1 + \mu)^{|\hat{f}(x)|} - 1)}{\mu}, \ -1 \le y \le 1. \quad (22)$$

Fig. 13 and Fig. 14 show the performance of MarkovNet and CsiNet-LSTM with $\mu$-law companding and fixed quantization step size at two different quantization levels, 6 bits and 4
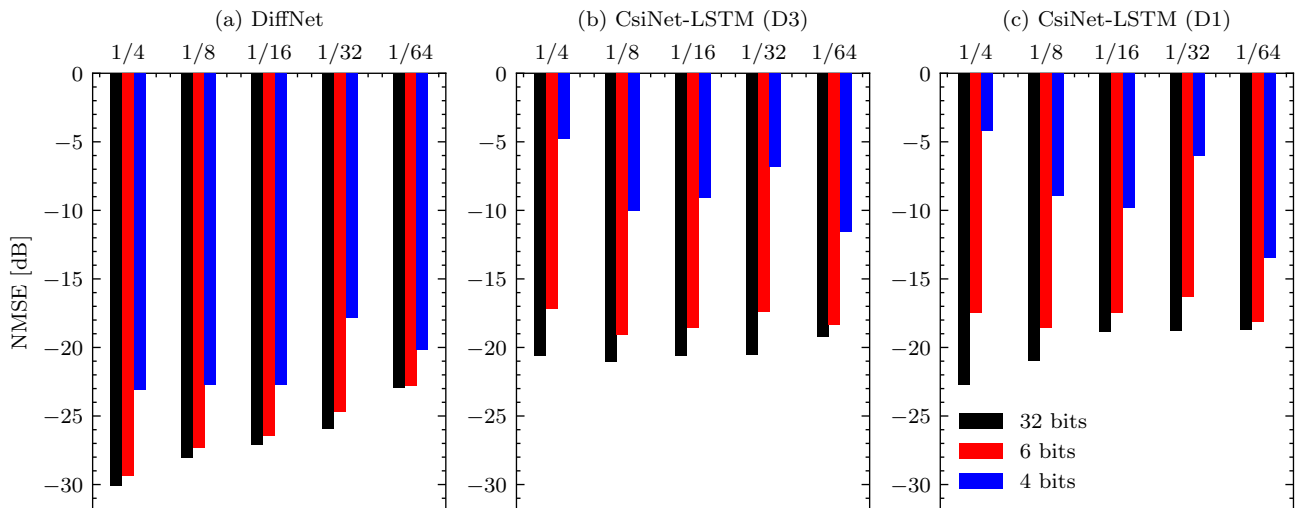
Fig. 13: MarkovNet and CsiNet-LSTM NMSE performance (dB) for indoor network with feedback subject to mu-law quantization using fixed step size, $\Delta = 2^{b-1}$, for $b$ bits.
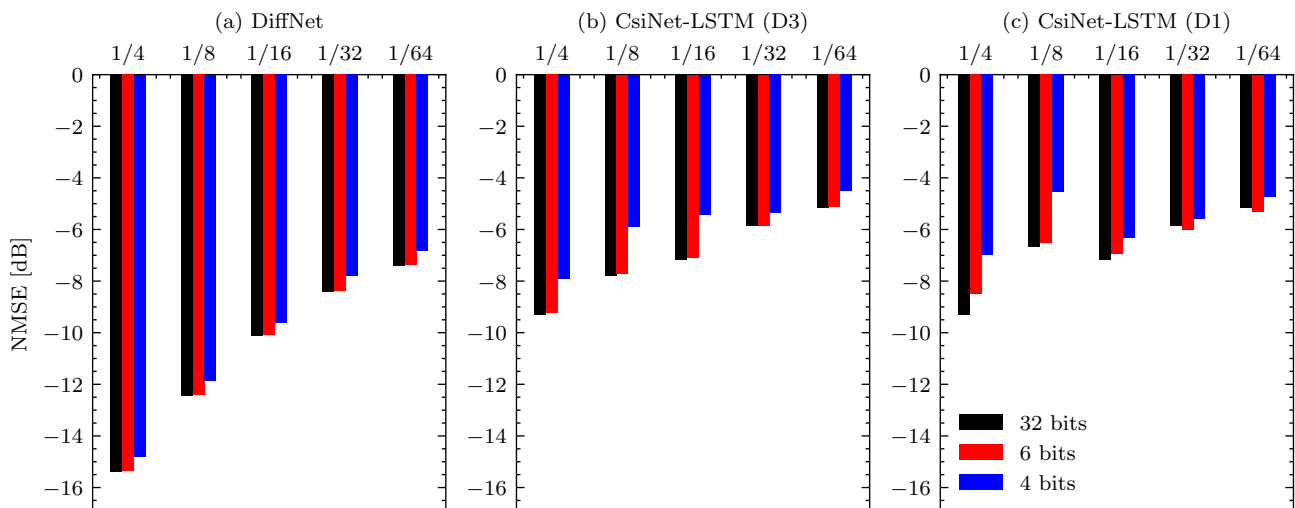


Fig. 14: MarkovNet and CsiNet-LSTM NMSE performance (dB) for Outdoor network with feedback subject to mu-law quantization using fixed step size, $\Delta = 2^{b-1}$, for $b$ bits.

bits, in comparison to the non-quantized network (i.e., 32 bit floating point). The networks with quantized feedback use 8 bit quantization at the first timeslot to establish good intial CSI estimates. Note that the networks are not re-trained or fine-tuned after applying quantization.

MarkovNet is more robust to feedback quantization noise than CsiNet-LSTM at either LSTM depth, as the former maintains NMSE better than -10 dB in both environments while the latter only exceeds -10dB at low compression ratios in the Indoor environment. Table IV summarizes the mean decrease in NMSE for each network, and for 6 and 4 bit $\mu$-law quantization, MarkovNet has a smaller mean degradation in NMSE performance compared to CsiNet-LSTM.

## VII. CONCLUSION

To better exploit temporal correlation, we provide an information theoretic basis for utilizing differential encoding with CNNs rather than applying overly parameterized LSTMs. We propose MarkovNet, a CNN with differential encoding, which achieves superior estimation accuracy and lowers computational complexity relative to an LSTM-based CSI estimation network. MarkovNet maintains accurate CSI estimates even under feedback quantization. We expand on a prior LSTM-based estimation technique and show that more parsimonious models can yield comparable or better performance.

## REFERENCES

[1] E. P. Simon, L. Ros, H. Hijazi, J. Fang, D. P. Gaillot, and M. Berbineau, "Joint carrier frequency offset and fast time-varying channel estimation for mimo-ofdm systems," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 955–965, 2011.

[2] W.-G. Song and J.-T. Lim, "Channel estimation and signal detection for mimo-ofdm with time varying channels," *IEEE Communications Letters*, vol. 10, no. 7, pp. 540–542, 2006.

[3] B. Makki and T. Eriksson, "On hybrid arq and quantized csi feedback schemes in quasi-static fading channels," *IEEE Transactions on Communications*, vol. 60, no. 4, pp. 986–997, April 2012.

[4] D. J. Love, R. W. Heath, V. K. N. Lau, D. Gesbert, B. D. Rao, and M. Andrews, "An overview of limited feedback in wireless communication systems," vol. 26, no. 8, pp. 1341–1365, 2008.

[5] H. Shirani-Mehr and G. Caire, "Channel state feedback schemes for multiuser mimo-ofdm downlink," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2713–2723, Sep. 2009.

[6] A. Hindy, U. Mittal, and T. Brown, "CSI feedback overhead reduction for 5g massive mimo systems," in *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 0116–0120.

[7] X. Rao and V. K. N. Lau, "Distributed compressive csit estimation and feedback for fdd multi-user massive mimo systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3261–3271, June 2014.

[8] M. E. Eltayeb, T. Y. Al-Naffouri, and H. R. Bahrami, "Compressive sensing for feedback reduction in mimo broadcast channels," *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3209–3222, Sep. 2014.

[9] Z. Gao, L. Dai, S. Han, C. I, Z. Wang, and L. Hanzo, "Compressive sensing techniques for next-generation wireless communications," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 144–153, 2018.

[10] E. Chen, R. Tao, and X. Zhao, "Channel equalization for ofdm system based on the bp neural network," in *2006 8th international Conference on Signal Processing*, vol. 3. IEEE, 2006.

[11] N. Taşpinar and M. N. Seyman, "Back propagation neural network approach for channel estimation in ofdm system," in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*. IEEE, 2010, pp. 265–268.

[12] C.-H. Cheng, Y.-P. Cheng *et al.*, "Using back propagation neural network for channel estimation and compensation in ofdm systems," in *2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems*. IEEE, 2013, pp. 340–345.

[13] K. Hiray and K. V. Babu, "A neural network based channel estimation scheme for ofdm system," in *2016 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, 2016, pp. 0438–0441.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.90

[15] C. Wen, W. Shih, and S. Jin, "Deep learning for massive mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, Oct 2018.

[16] T. Wang, C. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, April 2019.

[17] C. Lu, W. Xu, H. Shen, J. Zhu, and K. Wang, "Mimo channel information feedback using deep recurrent network," *IEEE Communications Letters*, vol. 23, no. 1, pp. 188–191, Jan 2019.

[18] Y. Liao, H. Yao, Y. Hua, and C. Li, "Csi feedback based on deep learning for massive mimo systems," *IEEE Access*, vol. 7, pp. 86 810–86 820, 2019.

[19] X. Li and H. Wu, "Spatio-temporal representation with deep neural recurrent network in mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 653–657, 2020.

[20] Y. Jang, G. Kong, M. Jung, S. Choi, and I. Kim, "Deep autoencoder based csi feedback with feedback errors and feedback delay in fdd massive mimo systems," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 833–836, 2019.

[21] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," 2017. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14184/14304

[22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, vol. 4, no. January, pp. 3104–3112, 2014.

[23] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 720–728, 2014.

[24] R. H. Jr and A. Lozano, *Foundations of MIMO communication*. Cambridge University Press, 2018.

[25] Z. Liu, L. Zhang, and Z. Ding, "Exploiting Bi-Directional Channel Reciprocity in Deep Learning for Low Rate Massive MIMO CSI Feedback," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 889–892, 2019.

[26] M. Hermans and B. Schrauwen, "Training and analysing deep recurrent neural networks," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 190–198. [Online]. Available: http://papers.nips.cc/paper/5166-training-and-analysing-deep-recurrent-neural-networks.pdf

[27] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, no. March 2014, 2014.

[28] Yoav Goldberg, "A Primer on Neural Network Models for Natural Language Processing," *Journal of Artificial Intelligence Research*, vol. 57, pp. 345–420, 2016. [Online]. Available: http://www.jair.org/papers/paper4992.html

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[30] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–27, 2009.

[31] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[32] L. Liu, C. Oestges, J. Poutanen, K. Haneda, P. Vainikainen, F. Quitin, F. Tufvesson, and P. D. Doncker, "The cost 2100 mimo channel model," *IEEE Wireless Communications*, vol. 19, no. 6, pp. 92–99, December 2012.

[33] T. S. Rappaport, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[34] S. Dhanani and M. Parker, "11 - entropy, predictive coding and quantization," in *Digital Video Processing for Engineers*, S. Dhanani and M. Parker, Eds. Oxford: Newnes, 2013, pp. 69 – 81. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780124157606000118

[35] K. Huber and S. Haykin, "Improved bayesian mimo channel tracking for wireless communications: incorporating a dynamical model," *IEEE transactions on wireless communications*, vol. 5, no. 9, pp. 2458–2466, 2006.

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

[37] S. Gao, P. Dong, Z. Pan, and G. Y. Li, "Deep learning based channel estimation for massive mimo with mixed-resolution adcs," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1989–1993, Nov 2019.

[38] Y. LeCun, L. Bottou, G. Orr, and K. Mller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[39] Z. Liu, M. del Rosario, X. Liang, L. Zhang, and Z. Ding, "Spherical normalization for learned compressive feedback in massive mimo csi acquisition," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[40] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

[41] A. Nisar, J. A. Sue, and J. Teich, "Performance comparison between machine leamins based lte downlink grant predictors," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*. The Steering Committee of The World Congress in Computer Science, Computer , 2019, pp. 226–232.