Viewing Channel as Sequence Rather than Image: A 2-D Seq2Seq Approach for Efficient MIMO-OFDM CSI Feedback

Zirui Chen, Zhaoyang Zhang, Zhuoran Xiao, Zhaohui Yang, and Kai-Kit Wong

Abstract

In this paper, we aim to design an effective learning-based channel state information (CSI) feedback scheme for the multiple-input multiple-output (MIMO) orthogonal frequency division multiplexing (OFDM) systems from a physics-inspired perspective. We first argue that the CSI matrix of a MIMO-OFDM system is in general physically closer to a two-dimensional (2-D) sequence rather than an image, due to its apparent unsmoothness, non-scalablity and translational variance within both the spatial and frequency domains. On this basis, we introduce a 2-D long short-term memory (LSTM) neural network to represent the CSI and propose a 2-D sequence-to-sequence (Seq2Seq) model for CSI compression and reconstruction. Specifically, one two-layer 2-D LSTM is used for CSI feature extraction, and another two-layer 2-D LSTM is used for CSI representation and reconstruction. The proposed scheme can not only fully utilize the unique 2-D characteristics of CSI, but also well preserve the index information and unsmooth features of the CSI matrix compared with current convolutional neural network (CNN) based schemes. We show the computational complexity of the proposed scheme is linear in the number of transmit antennas and subcarriers. Its key performances like reconstruction accuracy, convergence speed, generalization ability after short-term training, as well as robustness to lossy feedback are comprehensively compared with existing popular convolutional networks. Experiment results show that our scheme can bring up to nearly 7 dB gain in reconstruction accuracy under the same overhead and reduce feedback overhead by up to 75% under the same accuracy compared with the conventional CNN-based approaches.

Part of this work was presented at IEEE WCNC 2022 [1]. This work was supported in part by National Key R&D Program of China under Grant 2020YFB1807101 and 2018YFB1801104, and National Natural Science Foundation of China under Grant U20A20158 and 61725104.

Z. Chen, Z. Zhang, Z. Xiao are with College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and with International Joint Innovation Center, Zhejiang University, Haining 314400, China, and also with Zhejiang Provincial Key Laboratory of Info. Proc., Commun. & Netw. (IPCAN), Hangzhou 310007, China. (e-mail: {ziruichen, ning_ming, zhuoranxiao}@zju.edu.cn)

Z. Yang and K. Wong are with the Department of Electronic and Electrical Engineering, University College London, WC1E 6BT London, UK. (email: {zhaohui.yang, kai-kit.wong}@ucl.ac.uk)

Index Terms

CSI feedback, deep learning, MIMO-OFDM, 2-D LSTM, Seq2Seq.

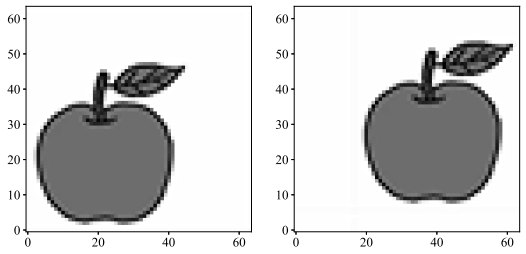
I. Introduction

Multiple-input multiple-output (MIMO) antenna array and orthogonal frequency division multiplexing (OFDM) are two of the key technologies in the fifth generation (5G) and beyond communication systems, thanks to their ability in achieving high spectrum efficiency under wideband transmission conditions [2], [3]. In a MIMO-OFDM system, the base station (BS) requires accurate downlink channel state information (CSI) for beamforming, subcarrier allocation and power control. Specifically, in the frequency division duplex (FDD) mode, downlink CSI needs to be estimated at the user equipment (UE) side, which then feedback the CSI to the BS. Due to the usually very large number of antennas and sub-carriers in a MIMO-OFDM system, the dimension of the CSI is extremely high, which results in a prohibitive CSI feedback cost.

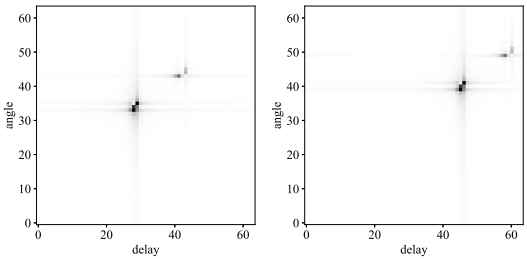
MIMO-OFDM CSI is mathematically presented as a matrix, of which the row/column index stands for antenna/subcarrier information. Due to the similarities in transmission paths between different antennas and subcarriers, there exist some implicit correlations among elements in CSI matrix. Thus, this makes the whole CSI matrix compressible. How to effectively compress CSI and reduce feedback cost has become an important topic in wireless communication. Various works have been done to solve this task, mainly based on signal processing and deep learning (DL) techniques.

Researchers initially tried pure signal processing-based methods.The works in [4], [5] proposed CSI reduction techniques based on vector quantization. Nevertheless, the overhead in the vector quantization scheme scales linearly with system dimensions, which limits the application in practical MIMO system. In addition, the authors in [6], [7] applied the compressed sensing (CS) algorithm. However, CS-based approaches rely on the sparsity of CSI data in a certain transform domain, which may not represent the channel structure accurately for many practical MIMO scenarios. Besides, CS-based approaches need to solve an optimization problem in the CSI reconstruction process, which results in high computational complexity.

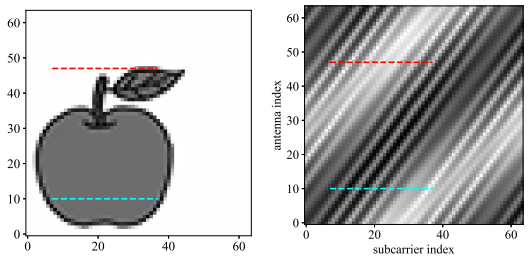
Recently, DL technology has become widely used in CSI compression task and achieved significant performance gains, demonstrating the great potential of DL techniques. The current DL-based CSI compression schemes can be divided into two categories: training a convolutional



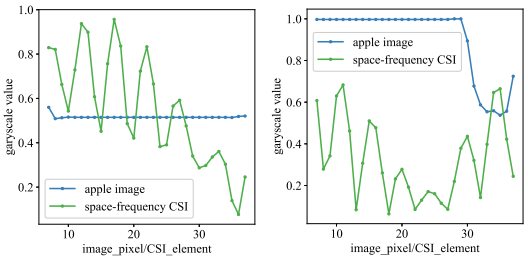
(a) Translational invariance of an image. Translation leads to similar image information.



(b) Translational variance of a angular-delay CSI matrix. Translation leads to totally different CSI characteristics.



(c) Smoothness of an image and unsmoothness of a spatial-frequency CSI matrix.



(d) Grayscale value changes versus the index of pixel and subcarrier. (Left figure is the cyan dot lines in (c), right figure is the red dot lines in (c))

Figure 1: Two essential differences between an image and a CSI matrix.

neural network (CNN) on angular-delay domain CSI and training a CNN on space-frequency domain CSI. In [8], the authors trained a CNN on angular-delay domain CSI named CsiNet for CSI compression and reconstruction. Further, a series of works [9]–[21] were derived to expand the CsiNet though adjusting the network structure of CNNs. Some works improved the accuracy [14]–[17] at the cost of high computational complexity, and some [19]–[21] reduced the complexity with slight accuracy loss. Unlike the above schemes based on angular-delay domain channel, a deep CNN based on the space-frequency domain CSI was proposed in [22]. In addition, the authors in [23] proposed a complex network with convolution and long short-term memory (LSTM) structure for the space-frequency domain channel. All above works [8]–[23] are all based on viewing CSI merely as an image and applying CNN-based image compression technology to solving the CSI feedback problem. However, there lacks theorectical physical support of directly applying CNNs to CSI feedback compression. Although CSI matrix and image do have similarities in some respects such as adjacent element/pixel correlation, they have two essential differences, translational variance/invariance and unsmothness/smoothness.

On one hand, the pixel coordinates of the image usually do not represent any exact physical meaning and the main information contained in the image does not strictly depend on the feature's

position. For example, as shown in Fig. 1(a), although the position of apple's pixels in these two images are very different, the main information contained in these two images is very similar. In brief, it is the relative relationship between adjacent pixels, not the coordinates of the pixels, that mainly determine the image's meaning. In contrast, as shown in Fig. 1(b), the element index in the CSI matrix has strict correspondence with the index of angles/time-delays or antennas/carriers, and any swap or translation will result in a totally different channel characteristic. Thus, the position coordinates of each element are critical and sensitive information in CSI, which is different from the piexl coordinates in images. Consequently, the translation invariance [24] of CNNs, helps to resist the interference of feature position in images, but brings information loss when CSI flows through CNN.

On the other hand, the adjacent pixels of images are usually numerically similar due to the high resolution of images, while the adjacent elements of the space-frequency domain CSI matrix are generally quite numerically different. Fig. 1(c) and Fig. 1(d) show the difference in smoothness between image and space-frequency domain CSI. Such difference makes the convolution operator's inherent smoothing effect [25], [26], quite suitable for images, but sometimes become a burden of CSI representation.

Considering the native diference between CSI matrix and image, in this paper, we first rethink the physical essence of the CSI data and then redesign the learning structure for CSI. Physically, each element in CSI matrix is a coupling of multipath channel responses. In a practical communication environment, the difference in channel responses between different antennas and subcarriers mainly comes from the propagation delay of electromagnetic waves, which brings about phase differences in each path. Because the adjacent antenna interval is equal and small, each propagation path's phase change between adjacent antennas is relatively uniform, which means that the channel responses have significant sequence characteristics in the spatial domain. Similarly, the sequence characteristics also exist in the frequency domain due to equal bandwidth between adjacent subcarriers. The two-dimensional (2-D) sequence characteristics drive us to reconsider the CSI matrix as a 2-D information sequence instead of an image.

For sequence data, the recurrent neural network (RNN) is commonly recognized as an effective structure because it can make full use of the dependencies between sequence elements [27]. Thus, we use the RNN with LSTM mechanism extended to the 2-D case named 2-D LSTM [28], [29] to complete CSI feature extraction and representation tasks, fully utilizing the 2-D sequence characteristics of CSI. Besides, since 2-D LSTM performs feature extraction and

representation through sequentially iterative computation, it can accurately capture and utilize the index information of CSI data and does not have smoothing effect, overcoming the limitations of CNN-based methods.

Similar with some machine translation tasks [30], [31], the compression and reconstruction of CSI is essentially a feature extraction and representation task from one 2-D sequence to one 2-D sequence. Therefore, we propose a 2-D Sequence to Sequence (Seq2Seq) neural network mainly composed of 2-D LSTMs for the CSI feedback task. The main contributions of this paper are summarized as follows:

- Through modeling and mathematical derivation of the downlink physical transmission process in MIMO-OFDM systems, we reveal the unique 2-D sequence characteristics of the MIMO-OFDM CSI and propose a new scheme of viewing CSI as a 2-D sequence to design learning structure, which is more physically reasonable than regarding CSI as an image.
- For effectively processing the CSI data, we introduce a 2-D LSTM neural network to perfectly match our research on the mathematical structure of the CSI. Moreover, we give a sub-block division method, which can be a referential data preprocessing approach for applying 2-D LSTM to CSI data.
- We analyze the framework of the CSI feedback task, which can be analogized to a machine translation task from one 2-D sequence to one 2-D sequence. Further, we propose a 2-D Seq2Seq model for CSI feedback and evaluate the computational complexity.
- Experiment results demonstrate that the proposed scheme outperforms the CNN-based reconstruction accuracy under the same computational complexity. Moreover, the proposed scheme has a fast convergence speed and strong robustness to meet demanding communication requirements.

The remainder of this paper is organized as follows. In Section II, we introduce the system model, including channel model, CSI feedback process, and the overall neural network structure. Then, our proposed 2-D Seq2Seq model and analysis are presented in Section III. Section IV shows the performance evaluation of our proposed model from various aspects. Section V draws the conclusion.

II. System Model

In this section, we first introduce the channel model of MIMO-OFDM system. Then, we describe the CSI feedback process. Finally, the overall structure for DL-based CSI feedback, an

autoencoder neural network, is presented.

A. Channel Model

We consider a MIMO-OFDM system, where a BS with $N_{\mathrm{t}} \gg 1$ antennas in the form of uniform linear array (ULA) serves a single-antenna UE adopting OFDM modulation with N_{c} subcarriers, as shown in Fig. 2. The channel between the BS and the user is assumed to be consist of P paths, which can be expressed as [32]

$$\mathbf{h}(f) = \sum_{p=1}^{P} \alpha_p e^{-j2\pi f \tau_p + j\phi_p} \mathbf{a}(\theta_p), \tag{1}$$

where f is the carrier frequency, α_p is the attenuation, τ_p is the propagation delay, ϕ_p is the initial phase shift and θ_p is the angle of departure (AoD) of the p-th path. Futhermore, $\mathbf{a}(\theta_p)$ is the array vector defined as

$$\mathbf{a}\left(\theta_p\right) = \left[1, e^{-j\chi\cos\theta_p}, \cdots, e^{-j\chi(N_{\mathrm{t}}-1)\cos\theta_p}\right]^T, \tag{2}$$

where $\chi = 2\pi df/c$, d is the antenna spacing, and c is the speed of light. Thus, the overall channel matrix $\mathbf{H} \in \mathbb{C}^{N_{\mathrm{t}} \times N_{\mathrm{c}}}$ between the BS and the user can be expressed as

$$\mathbf{H} = [\mathbf{h}(f_1), \mathbf{h}(f_2), \cdots, \mathbf{h}(f_{N_{\mathrm{c}}})], \tag{3}$$

where $f_i = f_0 + (i-1)\Delta f, (i = 1, 2, \cdots, N_{\mathrm{c}})$, f_i is the i-th subcarrier frequency, f_0 is the lowest subcarrier frequency and Δf is the bandwith of each subcarrier. This channel matrix $\mathbf{H}$ is referred to as the CSI in the literature.

B. CSI Feedback Process

For efficient transmission, the BS needs to acquire the downlink CSI for performing resource allocation, including beamforming designing and power allocation. In FDD mode, to this end, UE feeds back the estimated downlink CSI matrix $\mathbf{H}$ to the BS. However, considering the limited transmission resources in real scenarios, the excessive overhead of CSI feedback becomes prohibitive, especially in MIMO-OFDM systems.

As shown in channel model, the CSI of different subcarriers and antennas is correlated due to the similarity of propagation paths. Hence, we use this internal correlation to compress the

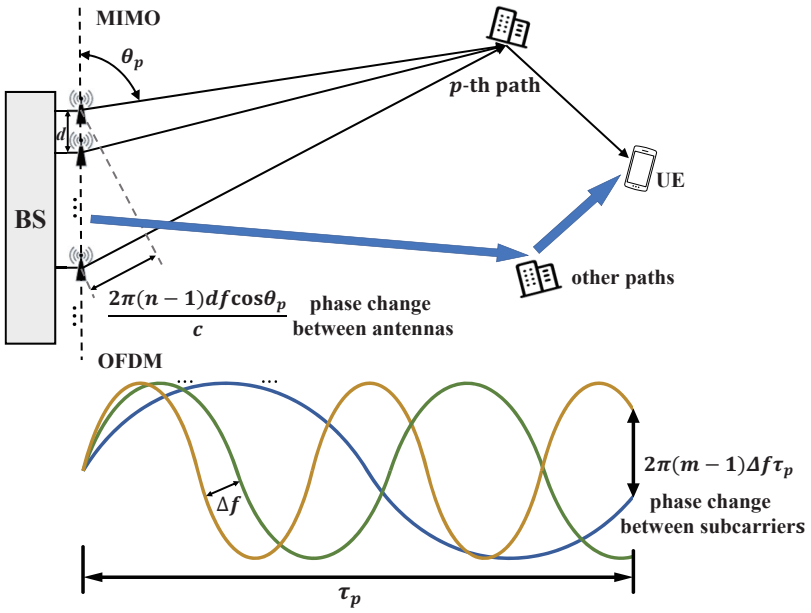


Figure 2: The MIMO-OFDM channel from BS to a user and the phase change of p-th path between different antennas and subcarriers.

CSI matrix. Specifically, we design a pair of matching functions f_{en} and f_{de} to complete the compression and reconstruction process of CSI, respectively, i.e.,

$$\mathbf{h}_{\text{com}} = f_{\text{en}}(\mathbf{H}), \tag{4}$$

$$\widehat{\mathbf{H}} = f_{\text{de}}(\mathbf{h}_{\text{com}}), \tag{5}$$

where $\mathbf{h}_{\text{com}}$ is the compressed CSI, a low-dimensional vector and $\widehat{\mathbf{H}}$ is the recovered CSI matrix [8]. With the help of f_{en} function, the UE transforms the CSI matrix $\mathbf{H}$ into a low-dimensional vector $\mathbf{h}_{\text{com}}$ and the BS generates the reconstructed CSI matrix $\widehat{\mathbf{H}}$ from received feedback information $\mathbf{h}_{\text{com}}$ based on f_{de} function. In this way, UE only needs to feedback a compressed low-dimensional vector $\mathbf{h}_{\text{com}}$ instead of the whole CSI matrix $\mathbf{H}$, which greatly reduces the feedback cost.

C. The Overall Structure for DL-based CSI Feedback

Since CSI is a complex coupling of multipath channel responses, the internal correlation between elements in the CSI matrix, brought by the single path similarity, is often implicit. Meanwhile, user location and scatter distribution are often unknown in the practical communication environment, which results in that the BS can not directly obtain specific single-path

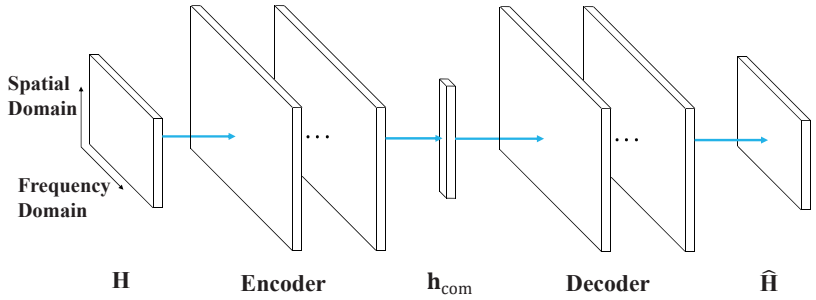


Figure 3: The overall autoencoder structure for CSI feedback.

information. Therefore, it is generally hard to directly extract the correlation through traditional signal processing algorithms. However, due to large-scale historical CSI data and the excellent ability in extracting implicit features of the deep neural network, we can use DL technology to explore the implicit correlation for efficient CSI compression.

As shown in Fig. 3, the overall deep learning structure for CSI feedback is an autoencoder neural network. The encoder part which compresses $\mathbf{H}$ into $\mathbf{h}_{\text{com}}$ implements the feature extraction and compression of CSI, completing the f_{en} function in equation (4), while the decoder part which reconstructs $\widehat{\mathbf{H}}$ from $\mathbf{h}_{\text{com}}$ performs channel representation and reconstruction, completing the f_{de} function in equation (5).

The neural network structure is a crucial parameter which largely determines the fitting and generalizing ability on the current task. If the physical characteristics of the CSI data can be fully integrated into the structure of the autoencoder, the neural network's feature extraction and representation capabilities on the CSI data will be greatly improved, hence parameter redundancy will be significantly reduced. Recalling Section II-A, MIMO-OFDM CSI has an elegant mathematical structure and contains unique physical properties. This elegant structure in CSI motivates us to redesign the learning structures in a similar structure. In the next section, we will introduce our mathematical derivation about the physical properties of CSI and the learning structures designed to match these properties.

III. Proposed 2-D Seq2Seq Model and Analysis

This section first analyzes the limitations of the CNN-based CSI feedback method, which is an important motivation for us to redesign the neural network. Then, we analyze the 2-D sequence characteristic of MIMO-OFDM CSI. Next, we introduce the 2-D LSTM neural network in detail. Finally, the proposed 2-D Seq2Seq model for CSI feedback is presented.

A. The Limitations of CNN-based CSI Feedback

Since both image and CSI can be mathematically represented by a matrix, and the adjacent matrix elements are correlated, the image and CSI have format similarities. Further, the CNNs widely used in image processing are adopted by current DL-based CSI feedback works thanks to these similarities. However, as shown in Fig. 1, there are several essential differences between CSI and images, which indicates CNN is not the best choice to deal with CSI feedback.

There are two main reasons that CNN works for image related tasks. On one hand, the convolution operator mainly extracts image features through template matching. Image features similar to the intensity distribution of the convolution kernel generate strong responses after being convolved. Then, these responses are aggregated after pooling or fully connected to judge the image's features. However, pooling or fully connected aggregation blurs the position information of features and finally makes the whole CNN insensitive to the position of image features. Therefore, CNN can effectively eliminate the interference of feature position [33], [34].

On the other hand, the convolution operator is essentially a neighborhood weighting algorithm [35]. Neighborhood weighting brings an interpolation effect and makes the convolved result smooth, protecting the continuity between adjacent pixels of the image. Thus, CNN is suitable for image generation tasks such as image denoising [26] and image super-resolution completion [25].

Next we analyze the limitations of CNN on CSI feedback task based on the working principle of CNN. CSI feedback task includes feature extraction and representation of CSI. In CSI feature extraction and compression, the relationship between each element value and the corresponding position index in the CSI matrix tends to lose when CSI flows into a CNN because CNN is insensitive to feature's position. In CSI representation and reconstruction, due to the lack of a unique design for position information, it is difficult for the convolution operator to generate the feature to the corresponding position accurately. Meanwhile, the numerical change between adjacent elements of the CSI matrix is a complex multipath coupling offset with significant discontinuity. Thus smoothness of the convolution operator brings some difficulties to the CSI expression.

As a DL method, CNN undoubtedly has excellent data fitting ability. Therefore, the CNN-based CSI feedback method can also work well with sufficient data and network scale. The limitations noted above are to indicate that CNN's property does not precisely match the CSI feedback task,

which makes the CNN-based approach not very efficient. Thus, it is of importance to design proper learning network structure suitable for CSI feedback, which can potentially achieve good performance with low complexity.

B. 2-D Sequence Characteristic of MIMO-OFDM CSI

CNN has an excellent performance in image processing because it conforms to the physical characteristics of images, which drives us to turn to the physical properties of CSI for inspiration in designing more suitable neural networks. Next,we show the unique 2-D sequence characteristic of MIMO-OFDM CSI in the following lemma.

Lemma 1 *There exists a function* $\mathrm{g}_{\mathbf{H}}(\cdot)$*, making the CSI matrix* $\mathbf{H}$ *expressed as:*

$$\mathbf{H}=\begin{bmatrix} \mathrm{g}_{\mathbf{H}}(N_{\mathrm{t}},1) & \mathrm{g}_{\mathbf{H}}(N_{\mathrm{t}},2) & \cdots & \mathrm{g}_{\mathbf{H}}(N_{\mathrm{t}},N_{\mathrm{c}}) \\ \vdots & \vdots & \iddots & \vdots \\ \mathrm{g}_{\mathbf{H}}(2,1) & \mathrm{g}_{\mathbf{H}}(2,2) & \cdots & \mathrm{g}_{\mathbf{H}}(2,N_{\mathrm{c}}) \\ \mathrm{g}_{\mathbf{H}}(1,1) & \mathrm{g}_{\mathbf{H}}(1,2) & \cdots & \mathrm{g}_{\mathbf{H}}(1,N_{\mathrm{c}}) \end{bmatrix}, \tag{6}$$

where the complexity of function $\mathrm{g}_{\mathbf{H}}(\cdot)$ *is independent with* $N_{\mathbf{t}}$ *and* $N_{\mathbf{c}}$*.*

Proof: Based on the channel model in Section II-A, we can obtain the CSI of the n-th antenna and the m-th subcarrier, which is also the element of the n-th row and the m-th column of the CSI matrix $\mathbf{H}$, as follows:

$$\begin{aligned}\mathbf{H}_{n,m}&=\sum_{p=1}^{P}\alpha_p e^{-j2\pi(f_0+(m-1)\Delta f)\tau_p+j\varphi_p}e^{-j\chi(n-1)\cos\theta_p}\\&=\sum_{p=1}^{P}\alpha_p e^{-j2\pi f_0\tau_p+j\varphi_p}e^{-j[\chi(\cos\theta_p)(n-1)+2\pi\Delta f\tau_p(m-1)]}.\end{aligned} \tag{7}$$

Among the parameters in equation (7), the attenuation α_p, the propagation delay τ_p, the initial phase shift ϕ_p, the AoD θ_p of each p-th path, the lowest subcarrier frequency f_0, the bandwith of each subcarrier Δf, and the antenna spacing related parameter χ, are mainly determined by the setting of the transmission system such as the BS and the user's location, the scattering environment information. The transmission system determined parameters do not change with which antenna or subcarrier is used for transmission. Thus, the variables $\alpha_p, \tau_p, \phi_p, \theta_p$ and the coefficient $f_0, \Delta f, \chi$ on the right side of equation (7), are all independent of n and m.

Then, we can define a function $\mathrm{g}_{\mathbf{H}}(x,y)$ as follows,

$$\mathrm{g}_{\mathbf{H}}(x,y)=\sum_{p=1}^{P}\alpha_p e^{-j2\pi f_0\tau_p+j\varphi_p}e^{-j[\chi\cos\theta_p(x-1)+2\pi\Delta f\tau_p(y-1)]}, \tag{8}$$

where function $\mathrm{g}_{\mathbf{H}}(\cdot)$ is composed of the parameters α_p, τ_p, ϕ_p, θ_p $(p=1,2,\cdots,P)$ and f_0, Δf, χ. The complexity of the function $\mathrm{g}_{\mathbf{H}}(\cdot)$ is completely determined by the above path or system variables and independent with $N_{\mathbf{t}}$ and $N_{\mathbf{c}}$. Then, since $\mathbf{H}_{n,m}=\mathrm{g}_{\mathbf{H}}(n,m)$, we can obtain that

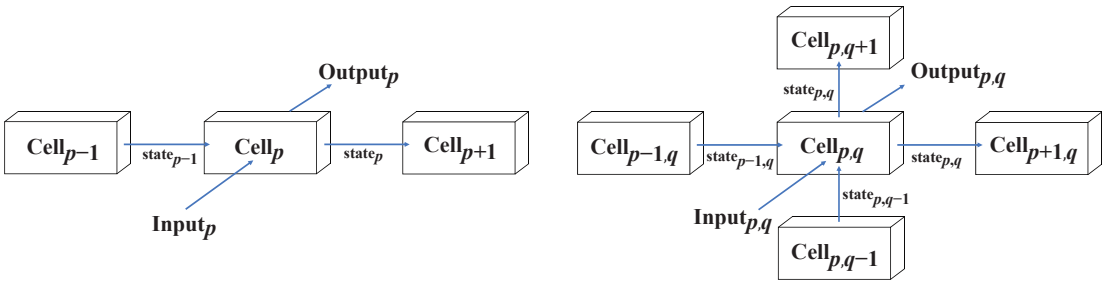
$$\mathbf{H}=\begin{bmatrix}\mathbf{H}_{N_{\mathrm{t}},1} & \mathbf{H}_{N_{\mathrm{t}},2} & \cdots & \mathbf{H}_{N_{\mathrm{t}},N_{\mathrm{c}}}\\ \vdots & \vdots & \iddots & \vdots\\ \mathbf{H}_{2,1} & \mathbf{H}_{2,2} & \cdots & \mathbf{H}_{2,N_{\mathrm{c}}}\\ \mathbf{H}_{1,1} & \mathbf{H}_{1,2} & \cdots & \mathbf{H}_{1,N_{\mathrm{c}}}\end{bmatrix}=\begin{bmatrix}\mathrm{g}_{\mathbf{H}}(N_{\mathrm{t}},1) & \mathrm{g}_{\mathbf{H}}(N_{\mathrm{t}},2) & \cdots & \mathrm{g}_{\mathbf{H}}(N_{\mathrm{t}},N_{\mathrm{c}})\\ \vdots & \vdots & \iddots & \vdots\\ \mathrm{g}_{\mathbf{H}}(2,1) & \mathrm{g}_{\mathbf{H}}(2,2) & \cdots & \mathrm{g}_{\mathbf{H}}(2,N_{\mathrm{c}})\\ \mathrm{g}_{\mathbf{H}}(1,1) & \mathrm{g}_{\mathbf{H}}(1,2) & \cdots & \mathrm{g}_{\mathbf{H}}(1,N_{\mathrm{c}})\end{bmatrix}, \tag{9}$$

which completes the proof. □

Remark 1 *Lemma 1 shows that the whole CSI matrix can be expressed as a 2-D sequence whose each element is only a function* $\mathrm{g}_{\mathbf{H}}(\cdot)$ *of its index. Besides, due to the path sparsity in high frequency communication, the complexity of function* $\mathrm{g}_{\mathbf{H}}(\cdot)$ *linear grows with the number of multiple paths, which is much smaller than the value of* $N_{\mathrm{t}}N_{\mathrm{c}}$.

According to Lemma 1, each element in the CSI matrix is a function of its row and column indexes, i.e., CSI has strong 2-D sequence characteristic. Next, we explain the essence of the 2-D sequence characteristic by analyzing the physical tranmission process in MIMO-OFDM system.

Consider a specific m-th column of the CSI matrix, i.e., CSI on the m-th subcarrier, $\mathbf{H}_{1,m}, \mathbf{H}_{2,m}, \cdots, \mathbf{H}_{N_{\mathrm{t}},m}$, which corresponds to the 1-st antenna to the N_{t}-th antenna. For any two adajcent antennas, the only difference in CSI is a fixed phase change $\chi\cos\theta_p$ for each p-th$(p=1,2,\ldots,P)$ path. Thus, when the antenna index n goes from 1 to $N_{\mathbf{t}}$, the response component of the p-th path forms a complex-valued proportional sequence. Further, the column $\mathbf{H}_{n,m}(n=1,2,...,N_{\mathrm{t}})$ is a sum of P complex-valued proportional sequences. It means that the CSI matrix has significant sequence characteristics in each column, which falls in the spatial domain. Similarly, consider a fixed n-th row of the CSI matrix, i.e., CSI on the n-th antenna, $\mathbf{H}_{n,1},\ \mathbf{H}_{n,2},\cdots,\mathbf{H}_{n,N_{\mathrm{c}}}$, which corresponds to the 1-st carrier to the N_{c}-th carrier. The only difference of the p-th path between any two adjacent subcarriers is also a fixed phase change $2\pi\Delta f\tau_p$. Thus, the sequence $\mathbf{H}_{n,m}(m=1,2,...,N_{\mathrm{c}})$ is also a sum of P complex-valued proportional sequences. It



(a) 1-D RNN (b) 2-D RNN

Figure 4: The graphical representation of 1-D abd 2-D RNNs.

means that the CSI matrix has significant sequence characteristics in each row, which belongs to the frequency domain. Fig. 2 shows the phase changes between different carriers and different antennas.

Lemma 1 and the above physical analysis show the 2-D sequence properties of the MIMO-OFDM channel in both spatial and frequency domains, which inspire us to regard the CSI matrix $\mathbf{H}$ as a 2-D information sequence.

C. 2-D RNN and 2-D LSTM Mechanism for MIMO-OFDM CSI

To deal with sequence-dependent data, RNN shows superior advantage due to its recurrent calculation structure. For data with multi-dimensional (M-D) sequence-dependence, the authors in [28], [29] introduced M-D RNN and M-D LSTM as a generalization of standard RNN and standard LSTM, respectively. Through using a M-D recurrent calculation structure, M-D RNN can overcome the limitation of one-dimensional (1-D) RNN, which can only utilize 1-D sequence dependence. Thus, M-D RNN has achieved good performance on some complex tasks [36]–[40], such as handwriting recognition and speech signal processing. However, due to different application purposes, each M-D RNN in the above works [36]–[40] has unique designed structure and settings corresponding to solving its own task. In this paper, we use the 2-D structure for handling MIMO-OFDM CSI data. Meanwhile, by analyzing the requirements of the MIMO-OFDM system, we propose the unique structure and settings in 2-D LSTM, that can be suitable for CSI. The designed structure and detailed setting are introduced as follows.

1) From 1-D RNN to 2-D RNN: 1-D RNN is one of the most popular and famous neural networks, including traditional RNN without gate structure and various variants represented by LSTM. The structure of 1-D RNN is shown in Fig. 4(a). In 1-D RNN, the input of each cell

includes two parts, its own information and state information from previous cell. Based on the input, this cell computes the state information and output. Then, the state information of this cell will contribute to the input of the next cell. This design enables the network to make full use of the sequence dependence between different cells. The structure of 2-D RNN is designed in a similar way, as shown in Fig. 4(b). In 2-D RNN, the input of each cell includes three parts, its own information and state information from two adjacent cells in left and below directions. Based on the input, this cell computes the state information and output. Then, the computed state information will flow to the input of the next two adjacent cells in right and above directions. In this way, the neural network can utilize sequence dependence in both horizontal and vertical directions. What's more, similar to 1-D RNN, the parameters of all cells in a 2-D RNN are commonly shared, which leads to a lightweight level of a 2-D RNN.

2) Internal Structure of a 2-D LSTM Cell: The scale of antennas and subcarriers in the MIMO-OFDM system is usually large, making CSI a long 2-D sequence. Using traditional 2-D RNN to process CSI may degrade performance due to the long-range processing difficulty. Therefore, we use a 2-D RNN with 2-D LSTM mechanism to process CSI data instead of directly using traditional 2-D RNN structure. In the 1-D case, the LSTM mechanism adds input gate, forget gate, output gate, and cell memory to the traditional RNN cell for storing long-term memory. Introducing a lambda gate to the 1-D LSTM can form a 2-D LSTM for averaging weight from two adjacent cell memories in left and below directions. Fig. 5 shows the internal structure of a 2-D LSTM cell. In this figure, the black part stands for a 1-D LSTM cell, and the blue part is the supplementary part of extending a 1-D LSTM cell to a 2-D LSTM cell.

Moreover, in Fig. 5, the subscript p, q of each variable corresponds to the owning cell, s is the state information, x is the input information, i is the input gate, o is the output gate, f is the forget gate, λ is the lambda gate, c is the cell memory, $\tilde{c}$ is the work memory of current cell, $\sigma(\cdot)$ is the sigmoid function, and $\mathrm{g}(\cdot)$ is the activation function. In this work, we choose $\tanh(\cdot)$ as the activation function $\mathrm{g}(\cdot)$. The internal calculation process of a 2-D LSTM cell can

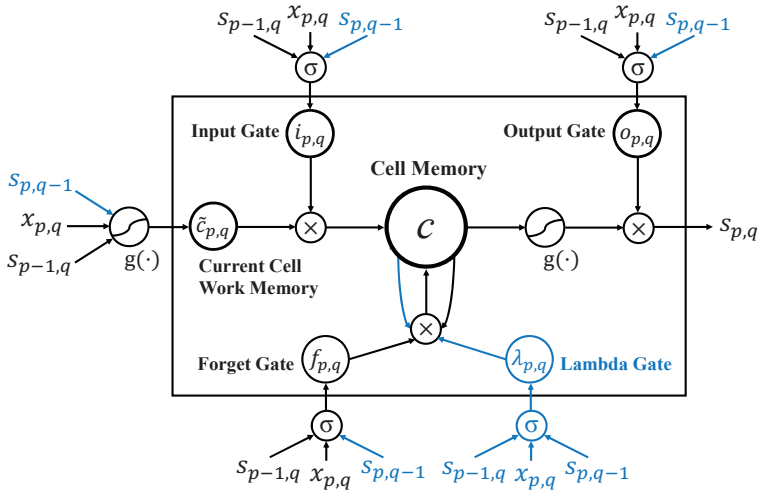


Figure 5: The internal structure of a 2-D LSTM cell.

be presented as follows:

$$\mathbf{i}_{p,q} = \sigma\left(\mathbf{W}_i\mathbf{x}_{p,q} + \mathbf{U}_i\mathbf{s}_{p-1,q} + \mathbf{V}_i\mathbf{s}_{p,q-1} + \mathbf{b}_i\right), \tag{10a}$$

$$\mathbf{f}_{p,q} = \sigma\left(\mathbf{W}_f\mathbf{x}_{p,q} + \mathbf{U}_f\mathbf{s}_{p-1,q} + \mathbf{V}_f\mathbf{s}_{p,q-1} + \mathbf{b}_f\right), \tag{10b}$$

$$\mathbf{o}_{p,q} = \sigma\left(\mathbf{W}_o\mathbf{x}_{p,q} + \mathbf{U}_o\mathbf{s}_{p-1,q} + \mathbf{V}_o\mathbf{s}_{p,q-1} + \mathbf{b}_o\right), \tag{10c}$$

$$\boldsymbol{\lambda}_{p,q} = \sigma\left(\mathbf{W}_\lambda\mathbf{x}_{p,q} + \mathbf{U}_\lambda\mathbf{s}_{p-1,q} + \mathbf{V}_\lambda\mathbf{s}_{p,q-1} + \mathbf{b}_\lambda\right), \tag{10d}$$

$$\tilde{\mathbf{c}}_{p,q} = \mathrm{g}\left(\mathbf{W}_{\tilde{c}}\mathbf{x}_{p,q} + \mathbf{U}_{\tilde{c}}\mathbf{s}_{p-1,q} + \mathbf{V}_{\tilde{c}}\mathbf{s}_{p,q-1} + \mathbf{b}_{\tilde{c}}\right), \tag{10e}$$

$$\mathbf{c}_{p,q} = \mathbf{f}_{p,q} \circ \left[\boldsymbol{\lambda}_{p,q} \circ \mathbf{c}_{p-1,q} + (\mathbf{1} - \boldsymbol{\lambda}_{p,q}) \circ \mathbf{c}_{p,q-1}\right] + \mathbf{i}_{p,q} \circ \tilde{\mathbf{c}}_{p,q}, \tag{10f}$$

$$\mathbf{s}_{p,q} = \mathbf{o}_{p,q} \circ \mathrm{g}\left(\mathbf{c}_{p,q}\right), \tag{10g}$$

where $\circ$ is the Hadamard product, $\mathbf{W}, \mathbf{U}, \mathbf{V}$ and $\mathbf{b}$ are parameters of the neural network and the subscript $i, f, o, \lambda, \tilde{c}$ of $\mathbf{W}, \mathbf{U}, \mathbf{V}, \mathbf{b}$ represents which gate or the cell memory computation this parameter participates, respectively.

3) Forward and Backward Propagation of a 2-D LSTM: The whole 2-D LSTM layer is composed of sequentially computed 2-D LSTM cell steps, whose unfloded view is shown in Fig. 6. The numbers of vertical and horizontal cell steps of the 2-D LSTM layer are respectively denoted by L_{ver} and L_{hor} and there is a total number of $L_{\mathrm{ver}}L_{\mathrm{hor}}$ cell steps to be calculated. Since the calculation of each cell step only depends on its previous cell steps in the horizontal and vertical directions, the cells distributed on the same diagonal can be calculated simultaneously. In wireless communication scenarios, the immediacy of CSI feedback is vital. Therefore, it

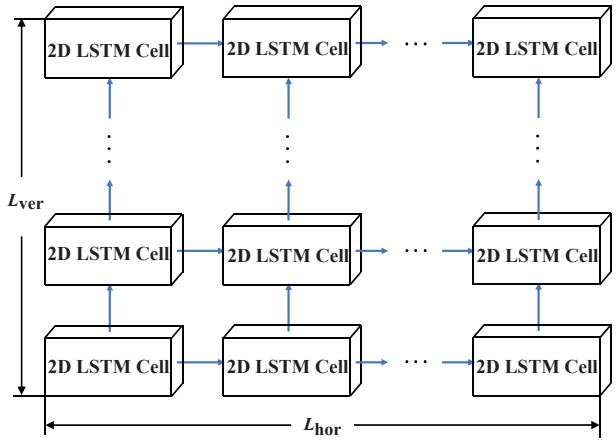


Figure 6: The unfloded view of a 2-D LSTM layer.

is necessary to use the parallel algorithm to reduce the computation time of the 2-D LSTM. The forward propagation of the whole 2-D LSTM is completed through calculating from the first diagonal cell to to the $(L_{\text{ver}} + L_{\text{hor}} - 1)$-st diagonal cell. This parallel computation can significantly boost the calculation speed and even reduce the calculation time from $\mathcal{O}(L_{\text{ver}}L_{\text{hor}})$ to $\mathcal{O}(L_{\text{ver}} + L_{\text{hor}})$.

To train the 2-D LSTM, we just need to use the backpropagation through time (BPTT) algorithm over two dimensions. Thus, 2-D LTSM neural networks can be directly deployed on all kinds of mainstream DL platforms. The backpropagation gradients pass in the reverse order of forwarding propagation, carring out iterative backtracking. Due to limited page, the detailed derivations of the backpropagation gradients can be found in [28].

Considering the 2-D sequence characteristic of CSI data and the excellent ability of 2-D LSTM in processing 2-D sequence data, it is very reasonable to apply the 2-D LSTM for CSI data feature extraction and representation. Furthermore, the CSI feedback task requires three processes, i.e., feature extraction, compression and representation reconstruction for CSI data. Thus, for CSI feedback task, we propose a 2-D Seq2Seq model, an autoencoder whose enoder and deocoder are both mainly composed of 2-D LSTMs.

D. Proposed 2-D Seq2Seq Model

We first introduce the whole structure of our proposed 2-D Seq2Seq model, as shown in Fig. 8. In this model, the encoder contains a two-layer 2-D LSTM, a fully connected layer, and a $\tanh(\cdot)$ activation function, where the two-layer 2-D LSTM performs feature extraction and

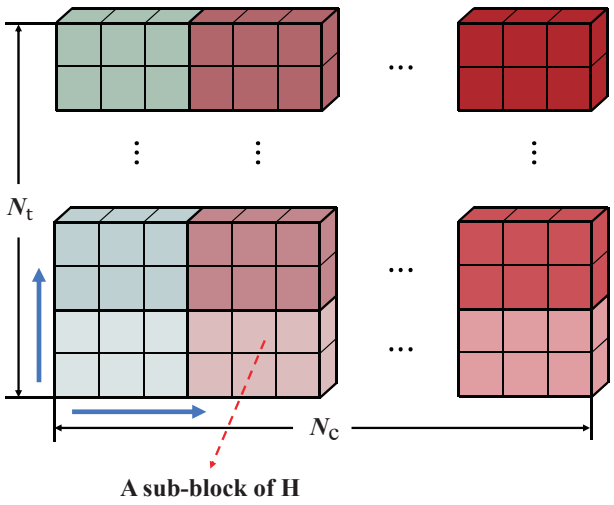


Figure 7: The division of a CSI matrix.

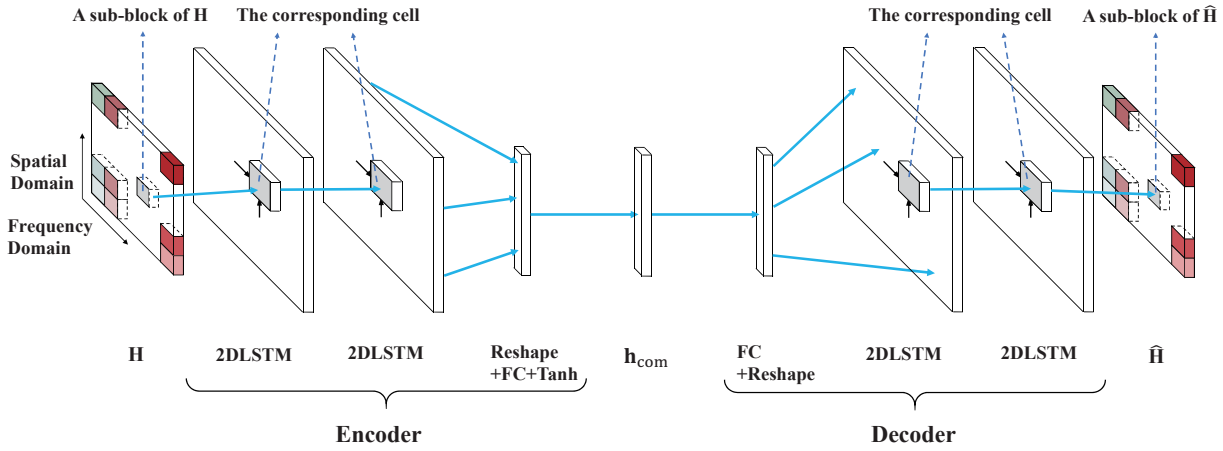


Figure 8: Overall structure of the proposed 2-D Seq2Seq model.

the fully connected layer realizes feature compression. The decoder is almost symmetrical with the encoder, consisting of a fully connected layer and a two-layer 2-D LSTM, where the fully connected layer reconstructs the compressed feature to the original CSI dimension size and the two-layer 2-D LSTM implements CSI representation. In addition, the two-layer 2-D LSTM is stacked two 2-D LSTM layers and the state information sizes of the two layers are equal. The size of the state information, also called the number of hidden-layer neurons, is a hyperparameter that can adjust the computational complexity of neural networks. Besides, to make 2-D LSTM extract the information behind CSI more effectively, based on assessing the approximate path information of CSI, we propose a sequence division preprocess for CSI data.

Specifically, we divide the CSI matrix into several sub-blocks, and all sub-blocks are arranged

sequentially to form a 2-D sequence, as shown in Fig. 7. When the whole CSI matrix is entered into the 2-D LSTM, each sub-block is fed into the corresponding cell of the 2-D LSTM. Similarly, when using the 2-D LSTM's output to reconstruct the CSI matrix, each cell obtains a reconstructed sub-block, and these sub-blocks are spliced in the order of the corresponding cell to form the reconstructed CSI matrix.

The reason for dividing CSI into some sub-blocks instead of some single CSI elements is that the path information contained in a single CSI element is often not obvious. Therefore, each sub-block requires several CSI elements to ensure that each cell's input information is relatively sufficient. Here we use a language sentence to make an analogy. A sub-block is similar to a word, while each CSI element is like a character. Only a complete word can contain enough information to participate in the semantics of the sentence. Instead, a character usually does not have sufficient meaning, and understanding sentences character by character is difficult.

Remark 2 *In CSI data preprocess, the sub-block size is a hyperparameter. On one hand, if the number of elements in a sub-block is too small, the information transmitted to each cell will be too poor. As a result, it is difficult for 2-D LSTM to obtain enough information from each sub-bock. On the other hand, if the number of elements in the sub-block is too large, the information sent to each cell will be too rich, making the information capacity of each cell relatively insufficient, and the correlation between different sub-blocks will also decrease, making the overall sequence characteristics weakened.*

According to the above analysis, we need to properly design the sub-block size in order to effectively apply 2-D LSTM to process CSI data. In particular, the influence of various sub-block size is shown in simulation results.

Next, we introduce the calculation process of the 2-D Seq2Seq model. In the encoder part, we first divide the original CSI matrix into several sub-blocks and each sub-block is entered into the corresponding two-layer 2-D LSTM cell. The output size of each cell in the second 2-D LSTM layer is set as the sub-block size. Following the two-layer 2-D LSTM, we splice the output of all cells into a $N_{\mathrm{t}} \times N_{\mathrm{c}}$-size matrix according to the corresponding cell order, and reshape the matrix into a $N_{\mathrm{t}}N_{\mathrm{c}}$-length vector. Then we forward the $N_{\mathrm{t}}N_{\mathrm{c}}$-length vector to the fully connected layer to generate a vector with the same length as $\mathbf{h}_{\mathrm{com}}$. Considering the quantization in practical applications, we use the $\tanh(\cdot)$ function to activate the output of the fully connected layer, obtaining the final feedback vector $\mathbf{h}_{\mathrm{com}}$, whose elements are all in

$(-1, 1)$. In the decoder part, we first input the feedback vector $\mathbf{h}_{\rm com}$ to the fully connected layer to generate a $N_{\rm t}N_{\rm c}$-length vector and reshape the vector to a $N_{\rm t} \times N_{\rm c}$-size matrix. Then, the matrix is divided into a 2-D sequence in the same way as the original CSI matrix, and each sub-block is input into the corresponding 2-D LSTM cell, respectively. The output size of each cell in the second 2-D LSTM layer is also set to the sub-block size. Finally, we splice the output of all cells into a $N_{\rm t} \times N_{\rm c}$-size matrix according to corresponding orders, which is the reconstructed CSI matrix $\widehat{\mathbf{H}}$.

To train the 2-D Seq2Seq model, we apply end-to-end learning to the whole autoencoder neural network. We denote the network parameter set by Θ= $\{\Theta_{\rm en}$,$\Theta_{\rm de}\}$, then $\widehat{\mathbf{H}} = f(\mathbf{H};\Theta) = f_{\rm de}(f_{\rm en}(\mathbf{H};\Theta_{\rm en});\Theta_{\rm de})$. Mean square error (MSE) is used as the loss function of the neural network, which can be written as

$$\mathrm{Loss}(\Theta) = \frac{1}{N}\sum_{n=1}^{N}\left\|\widehat{\mathbf{H}}_n - \mathbf{H}_n\right\|_2^2, \tag{11}$$

where N is the number of channel samples in the training set and $\|\cdot\|_2$ is the Euclidean norm. In addition, the parameter set Θ can be updated though the existing gradient descent based optimizers such as the adaptive momentum estimation (Adam) optimizer [41].

E. Parameter Scale and Computational Complexity

We analyze our proposed model's parameter scale and computational complexity in this subsection. We set the number of hidden-layer neurons in each 2-D LSTM cell as S, and the feedback length is L. Then, the number of total parameters in the whole 2-D Seq2Seq model is

$$\begin{aligned}&\mathcal{O}\left(2\left(\frac{5N_{\rm t}N_{\rm c}S}{L_{\rm ver}L_{\rm hor}} + 25S^2 + \frac{N_{\rm t}N_{\rm c}S}{L_{\rm ver}L_{\rm hor}} + N_{\rm t}N_{\rm c}L\right)\right)\\ =&\mathcal{O}\left(\frac{12N_{\rm t}N_{\rm c}S}{L_{\rm ver}L_{\rm hor}} + 50S^2 + 2N_{\rm t}N_{\rm c}L\right),\end{aligned} \tag{12}$$

where $\frac{N_{\rm t}N_{\rm c}}{L_{\rm ver}L_{\rm hor}}$ is the size of each sub-block in the CSI marix, $\frac{5N_{\rm t}N_{\rm c}S}{L_{\rm ver}L_{\rm hor}}$ is the number of connection parameters for inputting $\mathbf{H}$ into the two-layer 2-D LSTM of the encoder, the term $25S^2$ is the internal parameter of the two-layer 2-D LSTM in the encoder, the $\frac{N_{\rm t}N_{\rm c}S}{L_{\rm ver}L_{\rm hor}}$ is the number of connection parameters that the 2-D LSTM cell in the encoder generates an output with the sub-block size, the $N_{\rm t}N_{\rm c}L$ is the number of parameters of the fully connected layer in the enocoder and multiplying 2 is due to the symmetry between the encoder and the decoder.

Besides, the computational complexity is

$$\begin{aligned}&\mathcal{O}\left(2\left(L_{\mathrm{ver}}L_{\mathrm{hor}}\left(\frac{5N_{\mathrm{t}}N_{\mathrm{c}}S}{L_{\mathrm{ver}}L_{\mathrm{hor}}}+25S^2+\frac{N_{\mathrm{t}}N_{\mathrm{c}}S}{L_{\mathrm{ver}}L_{\mathrm{hor}}}\right)+N_{\mathrm{t}}N_{\mathrm{c}}L\right)\right)\\=&\mathcal{O}\left(12N_{\mathrm{t}}N_{\mathrm{c}}S+50L_{\mathrm{ver}}L_{\mathrm{hor}}S^2+2N_{\mathrm{t}}N_{\mathrm{c}}L\right),\end{aligned}\tag{13}$$

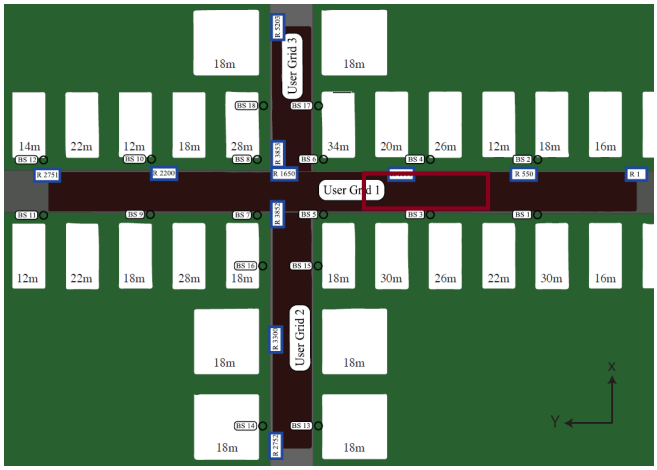
where multiplying 2 is also due to the symmetry between the encoder and decoder, and the term $L_{\mathrm{ver}}L_{\mathrm{hor}}\left(\frac{5N_{\mathrm{t}}N_{\mathrm{c}}S}{L_{\mathrm{ver}}L_{\mathrm{hor}}}+25S^2+\frac{N_{\mathrm{t}}N_{\mathrm{c}}S}{L_{\mathrm{ver}}L_{\mathrm{hor}}}\right)$ is the total computation of $L_{\mathrm{ver}}L_{\mathrm{hor}}$ two-layer cells in the encoder and $N_{\mathrm{t}}N_{\mathrm{c}}L$ is the computation of the fully connected layer in the enocoder.

IV. Performance Evaluation

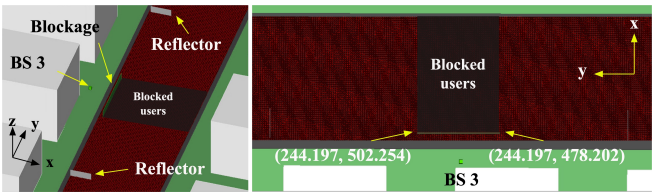
In this section, we evaluate the performance of the proposed scheme. We first introduce the datasets and the performance indices. Then, we compare our scheme with two typical DL-based CSI feedback schemes regarding channel reconstruction accuracy, convergence, and robustness. Finally, we analyze the impact of sub-block size, a hyperparameter mentioned in Section III-D, on the performance of the 2-D Seq2Seq model.

A. Datasets and Performance Indices

In this work, considering that the massive MIMO systems are mainly deployed in high-frequency outdoor communication scenarios, we use DeepMIMO datasets [42], [43] at 3.5GHz for training and testing. Moreover, we use two different outdoor scenarios 'O1' and 'O1_B' as communication environments, to solid the evaluation. Fig. 9(a) shows scenario 'O1', a common urban outdoor environment with two streets and one intersection. We set the BS 3 equipped with a ULA to serve the users in the red box area of Fig. 9(a), where we sample the channel data. Adding 'O1' scenario with a block wall and two reflective surfaces generates scenario 'O1_B', which is a more complicated communication scenario. Some users in 'O1_B' scenario only rely on the reflective path to communicate with the BS without the line of sight (LOS) path, as shown in Fig. 9(b). Compared to 'O1' scenario, the scattering environment information in 'O1_B' scenario is more complex, which brings more challenges to all schemes. The user areas and BSs in 'O1' and 'O1_B' are the same. Table I shows the detailed simulation settings for DeepMIMO datasets. In both 'O1' and 'O1_B' scenarios, we both randomly sample 60,000 channel data and divide them into the training set, validation set, and testing set with the ratio of 4:1:1.



(a) 'O1' scenario



(b) 'O1_B' scenario

Figure 9: Experimental scenarios in DeepMIMO dataset [42].

We use normalized MSE (NMSE) and cosine correlation ρ [8] as the performance indices, which are defined as follows:

$$\mathrm{NMSE} = \mathbb{E}\left\{\frac{\left\|\mathbf{H}-\widehat{\mathbf{H}}\right\|_2^2}{\|\mathbf{H}\|_2^2}\right\}, \tag{14}$$

and

$$\rho = \mathbb{E}\left\{\frac{1}{N_c}\sum_{m=1}^{N_c}\frac{|\widehat{\mathbf{h}}_m^H\mathbf{h}_m|}{||\widehat{\mathbf{h}}_m||_2||\mathbf{h}_m||_2}\right\}, \tag{15}$$

where the $\mathbf{h_m}/\widehat{\mathbf{h}}_m$ is the CSI of m-th subcarrier, i.e. the m-th columon of the CSI matrix $\mathbf{H}/\widehat{\mathbf{H}}$. Moreover, Floating-point operations (FLOPs) is used to represent the computational complexity of the neural networks.

B. Performance Comparisons

In this subsection, we compare our scheme with current DL-based CSI feedback schemes. As we introduced in Section I, current schemes can be divided into two categories: training a CNN on angular-delay domain CSI or space-frequency domain CSI. In each category, we

Table I: Parameters setting for DeepMIMO datasets

Parameters	**Value**
Frequncy band	3.5GHz
Bandwidth	100MHz
Base station	BS 3
Antenna	ULA
Number of antennas (N_{t})	32
Number of subcarriers (N_{c})	256
Number of paths (P)	25
User area	R701 - R1400

Table II: Parameters setting for all neural networks

Parameters	**Value**
Batch size	256
Learning rate	1×10^{-3} (multiply by 0.2 every 100 epochs after the 400th epoch)
Training epochs	800
Loss function	MSE
Optimizer	Adam [41]
FLOPs	$\approx$ {60million, 120million, 240million}
Length of feedback vector	128 by default
Number of training samples	40000
Number of testing samples	10000

choose a typical method as the benchmark. In particular, one benchmark is the CsiNet [8] which uses a residual CNN to compress angular-delay domain CSI, a typical method of the first category. Another benchmark is a scheme that uses an autoencoder network named AE_CNN whose encoder and decoder are two symmetric CNNs to compress space-frequency domain CSI, representing the second category. All schemes are trained individually on the 'O1' and 'O1_B' training sets. Meanwhile, for all simulations about 2-D Seq2Seq in this subsection, the sub-block size is set as 1 antenna $\times$ 32 subcarriers. Before providing simulation results, we make the following explanations.

- To ensure fair comparison, we select the same training strategy for all neural networks, as shown in Table II and compare performance under the same computational complexity.
- To comprehensively evaluate the performance, we train CsiNet, AE_CNN and 2-D Seq2Seq under various feedback lengths and various computational complexity by adjusting three hyperparameters: the width of the bottleneck layer, the number of convolutional kernels in CsiNet or AE_CNN, and the number of hidden-layer neurons in 2-D Seq2Seq.

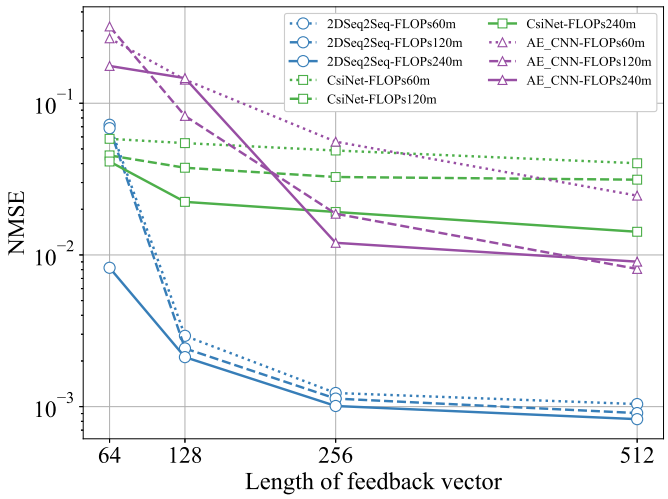
Table III: The cosine correlation ρ under various feedback lengths.

	O1								
	AE_CNN			CsiNet			2-D Seq2Seq		
Feedback length	FLOPs 60m	FLOPs 120m	FLOPs 240m	FLOPs 60m	FLOPs 120m	FLOPs 240m	FLOPs 60m	FLOPs 120m	FLOPs 240m
64	0.8506	0.8282	0.9147	0.9702	0.9609	0.9762	0.9592	0.9609	**0.9963**
128	0.9241	0.9605	0.9238	0.9778	0.9880	0.9911	0.9986	**0.9989**	**0.9989**
256	0.9735	0.9915	0.9885	0.9814	0.9880	0.9921	0.9994	**0.9995**	**0.9995**
512	0.9894	0.9964	0.9962	0.9856	0.9865	0.9942	0.9995	**0.9996**	**0.9996**
	O1_B								
	AE_CNN			CsiNet			2-D Seq2Seq		
Feedback length	FLOPs 60m	FLOPs 120m	FLOPs 240m	FLOPs 60m	FLOPs 120m	FLOPs 240m	FLOPs 60m	FLOPs 120m	FLOPs 240m
64	0.7117	0.4471	0.7262	0.9482	0.9296	0.9135	0.9233	0.9420	**0.9598**
128	0.8123	0.6471	0.7962	0.9582	0.9495	0.9434	0.9810	0.9864	**0.9892**
256	0.8920	0.9118	0.9159	0.9565	0.9738	0.9733	0.9904	0.9938	**0.9950**
512	0.9305	0.9336	0.9451	0.9549	0.9761	0.9768	0.9946	0.9959	**0.9961**

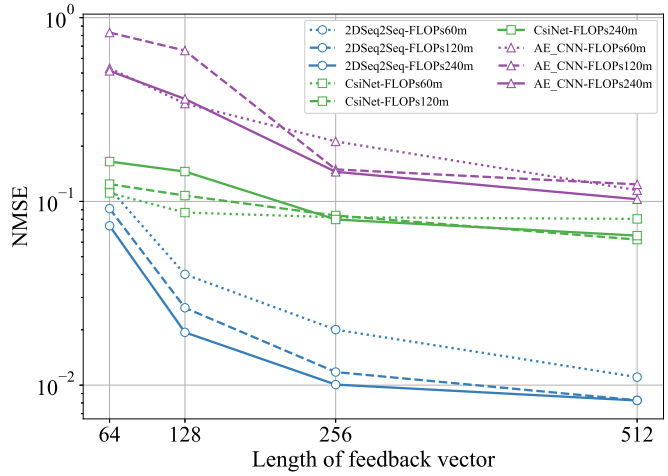
- The legends in the figures and table represent which scheme and the computational complexity. For example, '2DSeq2Seq-FLOPs60m' means the 2DSeq2Seq scheme with about 60 million FLOPs.

1) The Accuracy of Reconstructed CSI: The accuracy of the reconstructed CSI is a crucial performance evaluation indicator. Fig. 10(a), Fig. 10(b) and Table III show the comparisons among different methods in terms of reconstruction accuracy under various settings of feedback lengths. To be specific, Fig. 10(a) and Fig. 10(b) show the NMSE between the reconstructed CSI and the original CSI. Table III shows the cosine correlation ρ between reconstructed CSI and original CSI. The highest value of ρ in each feedback length is marked in bold font in Table III. Besides, we present the grayscale visualization of an original CSI and corresponding reconstructed CSI based on all schemes in Fig. 11, when the feedback length is 128 and the computational complexity of all schemes is about 240 million FLOPs.

According to the Fig. 10(a), Fig. 10(b) and Table III, our proposed method always provides lower NMSE and higher cosine correlation ρ compared to the benchmarks. Moreover, especially when the feedback information is relatively sufficient, the CSI reconstruction of 2-D Seq2Seq can be of very high quality, demonstrating the proposed scheme's superiority. Meanwhile, the performance of all schemes in 'O1_B' scenario degrades compared to 'O1' scenario, where AE_CNN degrades most significantly. What's more, when the feedback length is 64 or 128, the generalize performances of the AE_CNN under different FLOPs have a relatively signif-



(a) 'O1' scenario



(b) 'O1_B' scenario

Figure 10: The NMSE under various feedback lengths.

icant difference in the 'O1_B' scenario, which rarely appears in other schemes or scenarios. The significant performance degradation and unstable generalization capability indicate that the AE_CNN scheme is inadequate for the CSI compression task in the 'O1_B' scenario. The reason is due to the fact that the CSI in 'O1_B' scenario without LOS has complex unsmooth features and CNN cannot capture the highly unsmooth characteristics of spatial-frequency domain CSI.

From the visualization results shown in Fig. 11, we can intuitively observe that the proposed 2-D Seq2Seq method maintains the 2-D sequence characteristics of CSI very well and achieves extremely high-quality CSI reconstruction. Though comparing the CSI reconstructed by CsiNet

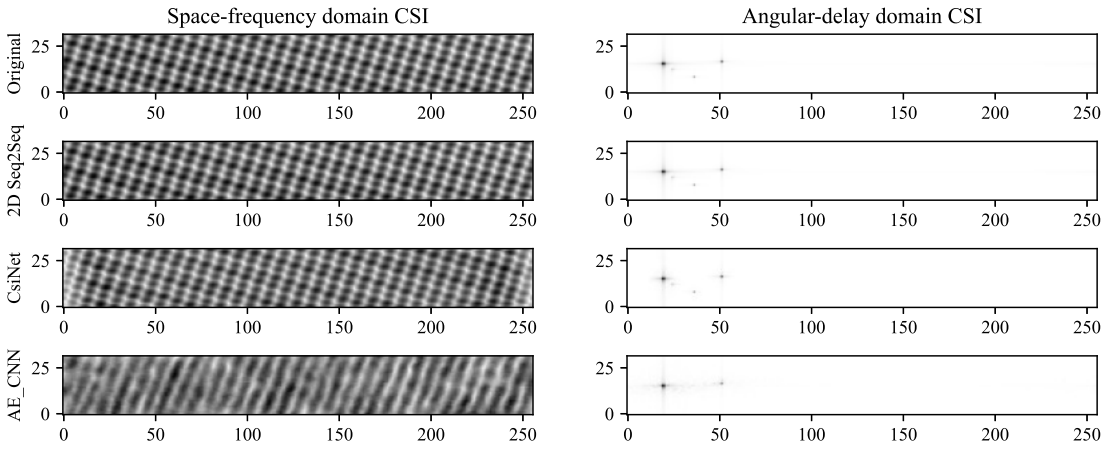


Figure 11: Grayscale visualization of the original CSI and reconstructed CSI.

and the original CSI in the angular-delay domain, we can find that the CSI reconstructed by CsiNet loses some channel power extension components in the angular-delay domain, which causes some distortions when the CSI is transferred to the space-frequency domain. This phenomenon reflects that the CNN-based CSI feedback method may lead to feature loss at the edge area of channel power extension components in angular-delay domain, which can verify the analysis in Section III-A. By comparing the CSI reconstructed by AE_CNN and the original CSI in the space-frequency domain, we can see that the CSI reconstructed by AE_CNN is over-smooth, which indicates that the CNN-based CSI feedback method can break some unsmooth features in CSI which we have also pointed in Section III-A.

2) Convergence and Generalization after Short-term Training: Fast and stable convergence is also very important for neural networks. Since the initial value of the training loss curve is significantly affected by network parameter initialization, it lacks clear significance to compare the loss curve under different schemes or settings. Therefore, here we only show the convergence curve of 2-D Seq2Seq with about 60 million FLOPs in Fig. 12, where the length of feedback vector is 128. The convergence speed of our proposed model is rapid. Furthermore, although the mini-batch gradient descent method is used, which brings some randomness to the gradient descent process, the overall convergence process is generally stable.

We pay more attention to whether the neural network can achieve excellent generalization after short-term training, directly determining the training time and cost. In the actual deployment, we consider broadcasting the network after short-term training to the users for carrying out CSI compression task and further optimizing it through online learning, in order to deploy the

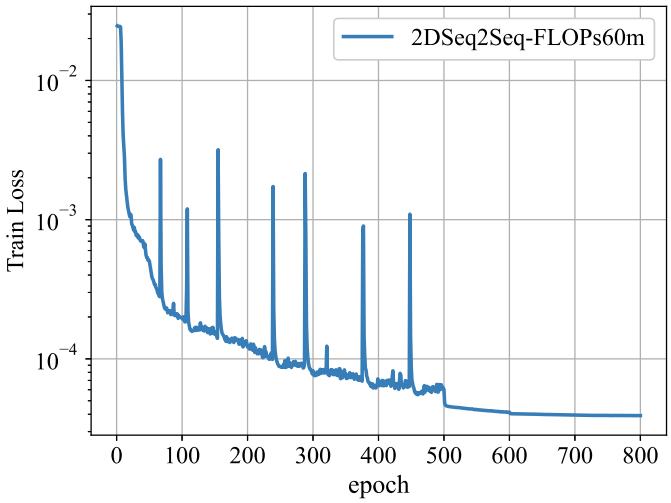
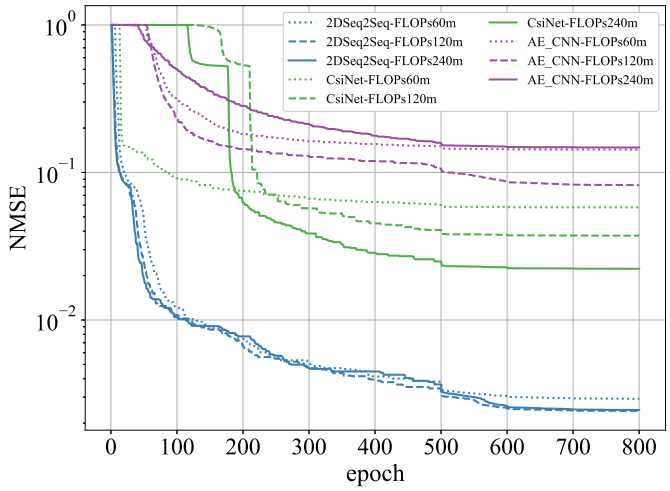


Figure 12: The loss function value of training 2-D Seq2Seq.

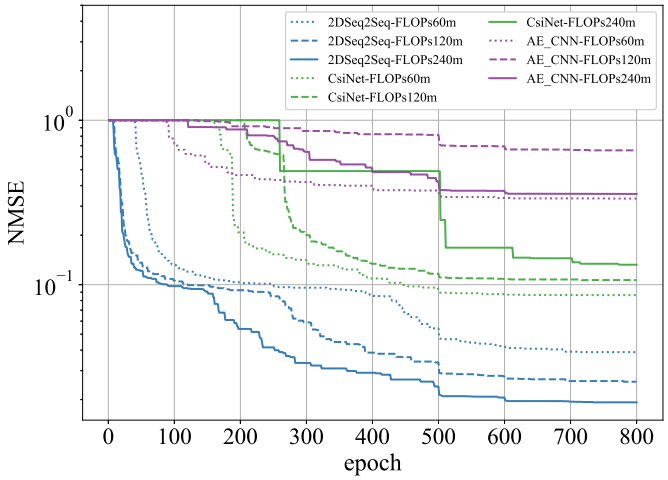
neural network faster. Here we compare the generalization ability of the networks on the test set after various training epochs. We set the feedback vector length as 128 in all schemes, and the simulation results are shown in Fig. 13(a) and Fig. 13(b). It should be noted that the network has application value only when NMSE is less than 1. Meanwhile, after the training of each epoch, we do not directly adopt the model after this epoch's training, but the best model in all previous epochs including this epoch, to evaluate the generalization ability, i.e., the best model's NMSE during training shown in Fig. 13. This choice prevents the training process randomness from interfering with the actual deployment performance.

As shown in Fig. 13, under both datasets and all computational complexity, the proposed 2-D Seq2Seq can obtain the best generalization ability after a short-term train, significantly outperforms than benchmarks, indicating that the proposed 2-D Seq2Seq can best meet the requirements of actual deployment. Furthermore, fast convergence and excellent generalization ability reflect that the proposed network structure fits the CSI feedback task well, again indicating the effectiveness of integrating CSI physical properties into neural network structure.

3) Robustness to Lossy Feedback: As introduced in Section II-B, users need to feed the compressed feedback vector $\mathbf{h}_{\rm com}$ back to the BS through the uplink channel. In the practical feedback process, $\mathbf{h}_{\rm com}$ may be polluted due to the uplink channel noise, which challenges the robustness of the neural networks. Here we evaluate the robustness to lossy feedback of the proposed scheme. Different from the above simulation that supposes ideal feedback vector $\mathbf{h}_{\rm com}$, we add disturbance with different intensity to the $\mathbf{h}_{\rm com}$ to evaluate the robustness of the



(a) 'O1' scenario



(b) 'O1_B' scenario

Figure 13: The best model's NMSE during training.

proposed scheme and compare it with that of benchmarks. Specifically, we add noise to $\mathbf{h}_{\text{com}}$ by multiplying a Gaussian random variable of $N(1, \sigma^2)$ to the $\mathbf{h}_{\text{com}}$, where varying σ indicates various pollution levels. Here, as an example, we set the length of the feedback vector as 128 and select the 'O1' scenario. Fig. 14 depicts the NMSE versus the intensity level of noise disturbance. From this figure, at low disturbance intensity ($\sigma \leq 0.2$), the proposed 2-D Seq2Seq shows the best NMSE performance and strong robustness. At high disturbance intensity ($\sigma > 0.2$), the CsiNet shows slightly lower NMSE than the proposed scheme. This phenomenon reflects that

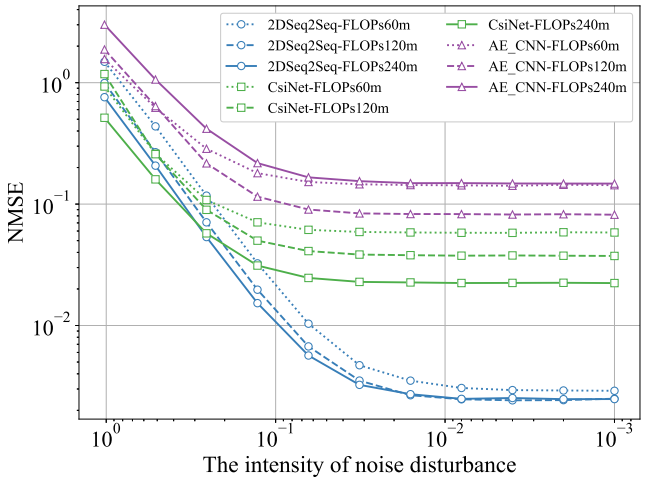


Figure 14: The NMSE under different disturbance σ in 'O1' scenario.

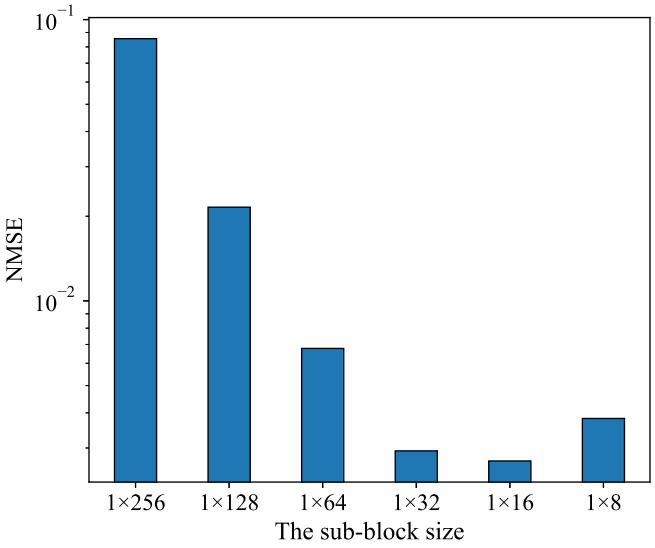


Figure 15: The NMSE under various sub-block sizes.

our proposed scheme inevitably loses some robustness while achieving high accuracy.

C. Tradeoff on the Sub-block Size

In Section III-D, we indicate that there is a tradeoff on the sub-block size. Different from the fixed sub-block size used in Section IV-B, here we change the number of the subcarriers in a sub-block and fix the antenna number as 1 to compare the CSI reconstruction accuracy under different sub-block sizes. We fix the feedback length as 128 and the number of hidden neurons as 60. Fig. 15 shows the NMSE under different sub-block sizes.

As the number of subcarriers in a sub-block decreases, NMSE of the 2-D Seq2Seq model first decreases and then increases. Moderate subcarrier number setting yields the best performance. This is consistent to our analysis finding in Remark 2, which shows too rich or too poor information in a sub-block both results in performance degradation. Therefore, there is a tradeoff on the sub-block size. Meanwhile, the network is not very sensitive to the size of the sub-blocks due to its good adaptive ability. As shown in Fig. 15, the network can both achieve excellent performance when the number of subcarriers is 16 and 32. Therefore, when applying a 2-D LSTM to the CSI data, we only need to avoid adopting too extreme sub-block size to obtain high-quality CSI reconstruction without a harsh CSI division scheme.

V. Conclusion

In this paper, we proposed a novel learning structure for CSI feedback. To address the limitations of current CNN-based CSI compression methods, we analyzed the physical essence of CSI, and proposed a new method for CSI feature extraction and representation using 2-D LSTM neural networks. Then, we designed a 2-D Seq2Seq model mainly composed of 2-D LSTMs for CSI compression and reconstruction to reduce CSI feedback cost. Simulation results show that the proposed 2-D Seq2Seq scheme outperforms other DL-based CSI compression schemes under various communication scenarios and various feedback requirements. Besides, in hyperparameter setting of 2-D Seq2Seq, it is revealed that the sub-block size should be chosen as a moderate number.

In our work, rather than directly designing the neural network structure for the CSI feedback task, we first rethought the physical essence and unique properties of CSI data, and proposed a reasonable perspective to view the MIMO-OFDM channel and a appropriate neural network structure to process CSI data on the above basis. Since this new perspective is closer to the physical essence of CSI, it may bring performance gains for more CSI data-related problems including CSI feedback task. We hope that our work can provide some inspirations for further exploring the applications of DL technology in MIMO-OFDM systems.

References

[1] Z. Chen, Z. Zhang, and Z. Xiao, "Viewing the MIMO channel as sequence rather than image: A Seq2Seq approach for efficient CSI feedback," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Austin, TX, USA, Apr. 2022 (accepted to appear).

[2] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, thirdquarter 2016.

[3] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-wave massive MIMO communication for future wireless systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 836–869, Secondquarter 2018.

[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," *IEEE Journal on selected areas in Communications*, vol. 26, no. 8, pp. 1341–1365, Oct. 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] P.-H. Kuo, H. T. Kung, and P.-A. Ting, "Compressive sensing based channel feedback protocols for spatially-correlated massive antenna arrays," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2012, pp. 492–497.

[7] X. Rao and V. K. 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, Jun. 2014.

[8] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 748–751, Oct. 2018.

[9] Q. Cai, C. Dong, and K. Niu, "Attention model for massive MIMO CSI compression feedback and recovery," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2019, pp. 1–5.

[10] 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, May 2020.

[11] 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.

[12] C. Lu, W. Xu, S. Jin, and K. Wang, "Bit-level optimized neural network for multi-antenna channel quantization," *IEEE Wireless Communications Letters*, vol. 9, no. 1, pp. 87–90, Jan. 2020.

[13] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–6.

[14] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2827–2840, Apr. 2020.

[15] Z. Cao, W.-T. Shih, J. Guo, C.-K. Wen, and S. Jin, "Lightweight convolutional neural networks for CSI feedback in massive MIMO," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2624–2628, Aug. 2021.

[16] X. Song, J. Wang, J. Wang, G. Gui, T. Ohtsuki, H. Gacanin, and H. Sari, "SALDR: Joint self-attention learning and dense refine for massive MIMO CSI feedback with multiple compression ratio," *IEEE Wireless Communications Letters*, vol. 10, no. 9, pp. 1899–1903, Sep. 2021.

[17] Z. Liu, L. Zhang, and Z. Ding, "An efficient deep learning framework for low rate massive MIMO CSI reporting," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4761–4772, Aug. 2020.

[18] Z. Lu, J. Wang, and J. Song, "Binary neural network aided CSI feedback in massive MIMO system," *IEEE Wireless Communications Letters*, vol. 10, no. 6, pp. 1305–1308, 2021.

[19] Y. Sun, W. Xu, L. Liang, N. Wang, G. Y. Li, and X. You, "A lightweight deep network for efficient CSI feedback in massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1840–1844, Jun. 2021.

[20] S. Ji and M. Li, "CLNet: Complex input lightweight neural network designed for massive MIMO CSI feedback," *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2318–2322, Oct. 2021.

[21] S. Jo and J. So, "Adaptive lightweight CNN-based CSI feedback for massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2776–2780, Dec. 2021.

[22] Q. Yang, M. B. Mashhadi, and D. Gündüz, "Deep convolutional compression for massive MIMO CSI feedback," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, Oct. 2019, pp. 1–6.

[23] Y. Liao, H. Yao, Y. Hua, and C. Li, "CSI feedback based on deep learning for massive MIMO systems," *IEEE Access*, vol. 7, pp. 86810–86820, 2019.

[24] W. Zhang et al., "Shift-invariant pattern recognition neural network and its optical architecture," in *Proceedings of annual conference of the Japan Society of Applied Physics*, Aug. 1988.

[25] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, Sep. 2014, pp. 184–199.

[26] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems*, vol. 29, pp. 2802–2810, Dec. 2016.

[27] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[28] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Ph. D. thesis*, 2008.

[29] G. Leifert, T. Strauß, T. Grüning, and R. Labahn, "Cells in multidimensional recurrent neural networks," *arXiv preprint arXiv:1412.2620*, 2014.

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

[31] I. Sutskever, O. Vinyals, and Quoc V Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, Dec. 2014, pp. 3104–3112.

[32] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1994–1998, Nov. 2019.

[33] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, Sep. 2014, pp. 818–833.

[34] D. Gupta, "Architecture of convolutional neural networks (CNNs) demystified," *Analytics Vidhya*, 2017.

[35] R. C. Gonzalez, R. E. Woods, et al., *Digital image processing*, Prentice hall Upper Saddle River, NJ, 2002.

[36] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," *Advances in neural information processing systems*, vol. 21, pp. 545–552, Dec. 2008.

[37] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, Oct. 2016, pp. 228–233.

[38] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "Exploring multidimensional LSTMs for large vocabulary asr," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Mar. 2016, pp. 4940–4944.

[39] T. N. Sainath and B. Li, "Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks," in *Interspeech, 17th Annual Conference of the International Speech Communication Association*, Sep. 2016, pp. 813–817.

[40] P. Bahar, C. Brix, and H. Ney, "Towards two-dimensional sequence to sequence model in neural machine translation," *arXiv preprint arXiv:1810.03975*, 2018.

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

[42] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," *arXiv preprint arXiv:1902.06435*, 2019.

[43] Remcom, "Wireless InSite," http://www.remcom.com/wireless-insite.