DS-NLCsiNet: Exploiting Non-local Neural Networks for Massive MIMO CSI Feedback

Xiaotong Yu, Xiangyi Li, Huaming Wu, *Member, IEEE* and Yang Bai

Abstract—Channel state information (CSI) feedback plays an important part in frequency division duplex (FDD) massive multiple-input multiple-output (MIMO) systems. However, it is still facing many challenges, e.g., excessive feedback overhead, low feedback accuracy and a large number of training parameters. In this paper, to address these practical concerns, we propose a deep learning (DL)-based CSI feedback scheme, named DS-NLCsiNet. By taking advantage of non-local blocks, DS-NLCsiNet can capture long-range dependencies efficiently. In addition, dense connectivity is adopted to strengthen the feature refinement module. Simulation results demonstrate that DS-NLCsiNet achieves higher CSI feedback accuracy and better reconstruction quality for the same compression ratio, when compared to state-of-the-art compression schemes.

Index Terms—Massive MIMO, Frequency Division Duplex (FDD), CSI Feedback, Non-local Neural Networks, Densely Connected Convolutional Networks.

I. INTRODUCTION

RECENTLY, massive multiple-input multiple-output (MIMO) has emerged as one of the pivotal technologies for fifth-generation (5G) wireless communication systems [1]. Equipped with multiple transmitters and receivers, massive MIMO systems have achieved great efficiency in terms of system capacity and anti-interference ability. In frequency division duplex (FDD) systems, one of the key procedures is to exploit CSI at the base station (BS), which is essential for performance improvement. In conventional FDD MIMO systems, the downlink CSI is obtained at the user equipment (UE), and then fed back to the BS through feedback links without compression. However, this method is prohibited in massive MIMO systems and it is difficult to acquire a large amount of accurate CSI in practical FDD systems since the feedback overhead is extremely huge [2].

The challenge of CSI feedback in massive MIMO systems has stimulated numerous studies. Currently, compressive sensing (CS)-based methods [3]–[5] are exploited to compress the CSI, but these methods are not effective enough since they require strict sparsity, so that channel matrices can hardly fit. Moreover, although CS simplifies the compression process, it turns decompression into an optimization problem, which requires an iterative algorithm and thus increases huge computational and time costs.

Recently, deep learning (DL)-based methods have been introduced to CSI feedback tasks, and have shown great potential in CSI recovery [6]. Compared to CS-based methods, DL-based methods achieve a significant improvement in model performance and computational speed. Wen et al. [7] proposed an autoencoder (AE)-based network named CsiNet, which uses an encoder to compress the channel matrices into codewords, and a decoder to transform the codewords into recovered channel matrices. Exploiting convolutional operation and ResNet architecture [8] in the CSI feedback tasks, CsiNet outperforms existing CS-based algorithms at all compression ratios. Recurrent neural network (RNN) is widely utilized in new CSI feedback frameworks, such as CsiNet-LSTM [9], RecCsiNet [10] and ConvlstmCsiNet [11]. CsiNet-LSTM [9] focused on modifying the channel recovery module, while RecCsiNet [10] enhanced the feature compression and decompression module, while considering the temporal correlation of the channel matrices to further improve the recovery quality. ConvlstmCsiNet [11] further exploited depthwise separable convolutions to reduce the computational complexity and adopted convLSTM and Pseudo-3D to explore the spatial-temporal representation of channel information. In addition, CRNet [12] was based on an inception model to adapt to changes in granularity, and proposed an advanced training scheme to enhance the network performance. CsiNetPlus [13] has investigated the influence of convolutional kernels. CoCsiNet [14] utilized the correlation between nearby UEs to recover CSI cooperatively. CS-ReNet [15] can significantly reduce the feedback overhead and lower the complexity of implementing CS at the UE. Guo et al. [16] discussed the trend of computational complexity of neural networks and introduced compression and acceleration techniques for communication systems.

To significantly boost the correctness of CSI feedback and reduce the computational complexity of neural networks, we design a novel CSI feedback architecture based on non-local neural networks [17], where non-local blocks are applied to extract long-distance dependencies. In addition, we improve the RefineNet module with dense connectivity [18] to strengthen the feature propagation and enhance the information flow. The scheme brings improvements in terms of CSI feedback accuracy and reconstruction quality.

The main contributions of this paper are listed as follows:
- We propose an innovative DL-based CSI feedback and recovery mechanism, referred to as DS-NLCsiNet, which has the potential for practical deployment on real FDD MIMO systems.
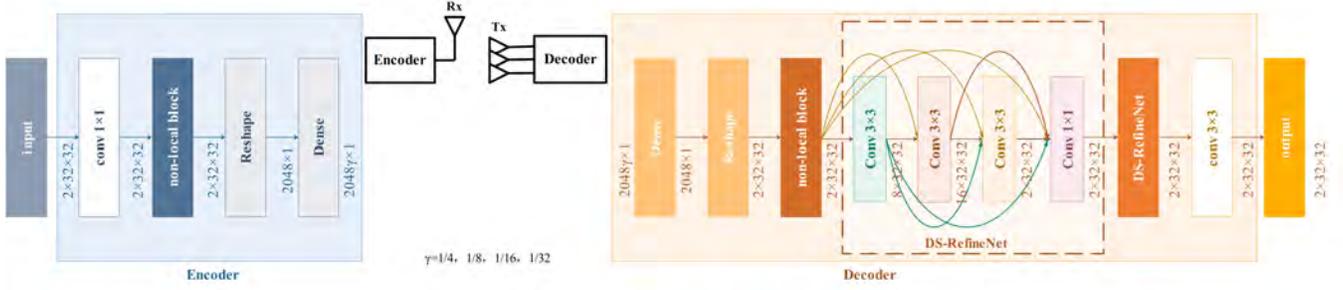- In DS-NLCsiNet, non-local blocks from non-local neural

Fig. 1: The architecture of proposed DS-NLCsiNet.

networks are applied to modify the feature extraction module and improve its efficiency in capturing long-range dependencies. Furthermore, dense connectivity is utilized to significantly enhance the recovery quality by encouraging feature reuse.

- Experimental results show DS-NLCsiNet can recover CSI more accurately and improve the quality of recovered CS significantly when compared with some existing methods.

II. SYSTEM MODEL

We consider a single-cell downlink FDD massive MIMO system constituted of N_t ($N_t \gg 1$) antennas at the BS as well as a single antenna at each user equipment (UE). The system is operated in orthogonal frequency-division multiplexing (OFDM) with $\widetilde{N}_c$ subcarriers.

The received signal at the nth subcarrier is given as:

$$y_n = \widetilde{\mathbf{h}}_n^H \mathbf{v}_n x_n + z_n, \tag{1}$$

where $\widetilde{\mathbf{h}}_n^H \in \mathbb{C}^{N_t \times 1}$, $\mathbf{v}_n \in \mathbb{C}^{N_t \times 1}$, $x_n \in \mathbb{C}$ and $z_n \in \mathbb{C}$ denote the channel vector in the frequency domain, precoding vector designed by the BS, modulated transmit data symbol, and additive Gaussian white noise at the nth subcarrier, respectively. Then the downlink CSI matrix $\widetilde{\mathbf{H}}$ is firstly obtained at the UE side via the downlink pilots, which can be modeled as:

$$\widetilde{\mathbf{H}} = [\widetilde{\mathbf{h}}_1, \widetilde{\mathbf{h}}_2, \cdots, \widetilde{\mathbf{h}}_{\widetilde{N}_c}] \in \mathbb{C}^{N_t \times \widetilde{N}_c}. \tag{2}$$

The total number of feedback elements is $N_t \times \widetilde{N}_c$, which will lead to high feedback overhead that beyond the system capacity in a massive MIMO system. In order not to concentrate on complicated details and challenges, we hypothesize that perfect CSI has been acquired by the UE, and the BS can process the precoding vector $\mathbf{v}_n$ as long as it receives the downlink CSI feedback $\widetilde{\mathbf{H}}$.

To reduce feedback overhead, $\widetilde{\mathbf{H}}$ can be further transformed into a sparsified matrix $\bar{\mathbf{H}}$ in the angular-delay domain via a two-dimensional discrete Fourier transform (2D-DFT) operation as follows:

$$\bar{\mathbf{H}} = \mathbf{F}_d \widetilde{\mathbf{H}} \mathbf{F}_a^H, \tag{3}$$

where $\mathbf{F}_d \in \mathbb{C}^{\widetilde{N}_c \times \widetilde{N}_c}$ and $\mathbf{F}_a \in \mathbb{C}^{N_t \times N_t}$ are both DFT matrices [9]. Moreover, since the time delay between multipath arrivals lies within a limited time period, only the first few columns of $\bar{\mathbf{H}}$ having distinct non-zero values [7]. Thus, we only retain the first N_c ($N_c < \widetilde{N}_c$) columns, and remove the rest columns. $\bar{\mathbf{H}}$ is then truncated to a $N_t \times N_c$ sized CSI matrix $\mathbf{H}$, which still requires huge overhead for the massive MIMO system.

After performing the 2D-DFT and truncation operation, we separate the channel matrix $\mathbf{H}$ into real and imaginary parts. Then we feed it into the autoencoder network as depicted in Fig. 1, which includes the encoder and decoder. The encoding and decoding procedures of CSI can be expressed as follows, respectively.

$$\boldsymbol{t} = f_{en}(\mathbf{H}), \tag{4}$$

$$\widehat{\mathbf{H}} = f_{de}(\boldsymbol{t}), \tag{5}$$

where the encoder compresses the CSI matrix of size q into a codeword $\boldsymbol{t}$ of length p, and then sends it back to the BS for CSI recovery. Then, the compression ratio (CR) is defined as $CR = p/q$. After the BS finishes decompressing the codeword $\boldsymbol{t}$ to the original channel, we can obtain the recovered channel matrix $\widetilde{\widehat{\mathbf{H}}}$ by performing zero filling and inverse DFT procedure.

III. DS-NLCsiNet

The architecture of the proposed DS-NLCsiNet is shown in Fig. 1, constituted of an encoder at the UE and a decoder at the BS.

In DS-NLCsiNet, it receives the truncated matrix $\mathbf{H}$ of size $N_t \times N_c \times 2$ as input and sends it to a 1×1 convolution for initial information interaction. Then a non-local block is applied to extract features, especially for capturing long-distance dependencies on the structure of channel matrix. The output of the non-local block remains the same shape with the input, and is then fed to the reshaping layer to stretch into a $2N_cN_t$-sized vector. The dense layer compresses the vector into the p-sized ($p < 2N_cN_t$) real-valued codeword $\boldsymbol{t}$, where p satisfies the compression ratio (CR) standard: $CR = p/2N_cN_t$. The codeword is then fed back to the BS.

After the decoder at the BS receives the codeword, it first decompresses the p-sized codeword and reshapes it into a $N_t \times N_c \times 2$ sized rough recovery of $\mathbf{H}$. Further refinements are divided into two parts: *Global Structure Refinement* and *Local Detailed Refinement*. A non-local block is first deployed to help reconstruct the global structure of CSI matrix, utilizing its high efficiency of transferring information between remote pixels. Then two DS-RefineNet blocks are used to supplement the local details, where the convolutions are local operations, which are more suitable for detailed reconstruction. Following the DS-RefineNets, a 3×3 convolutional layer is implemented to scale the values to the $[0, 1]$. The final reconstruction of $\mathbf{H}$ is generated. For each convolutional layer, we use leaky ReLU as the activation function and place a batch normalization layer.

A. Non-local Block

Existing DL-based CSI feedback architectures usually exploit convolutional or recurrent operations to extract the features of the channel matrix. However, these two methods can only deal with one local neighborhood at a time. To obtain a larger resolution view, we usually need to repeat these operations, which is computationally inefficient and will cause optimization difficulties. Therefore, we introduce non-local (NL) blocks from non-local neural networks [17] to the CSI Feedback architecture, in order to capture long-range dependencies well.

The NL-block is specially designed for sequence data (spatial, temporal or spatial-temporal) and can directly pass information between any two positions. The main idea comes from the NL-Mean algorithm for image denoising, that is, displaying mean operation on all image blocks, which is calculated by:

$$u(x_i) = \sum_{y \in \Omega} w(x_i, y) v(y). \tag{6}$$

To highlight commonalities and eliminate differences (usually noise), the normalized weight coefficient $w(x_i, y)$ is involved here, i.e., the more similar block y with the output $u(x_i)$, the higher weight is given.

Similar to the NL-Mean operation as shown in Eq. 6, the generic NL operation can be expressed as:

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j), \tag{7}$$

where x and y denote the input and output feature maps, respectively, i is the index of a position on feature maps, and j represents all possible positions on x. $C(x) = \sum_{\forall j} f(x_i, x_j)$ is the normalization factor. The function g computes the embedded feature representation of the input feature map at the position j. Here, we use a 3×3 convolution for g. The function f computes the correlation between index i and j, i.e. auto-correlation coefficient matrix. Several forms of f can be selected, e.g., Gaussian, embedded Gaussian and dot product [17], and we choose the embedded Gaussian form:

$$f(x_i, x_j) = e^{\theta(x_i)^T \emptyset(x_j)}, \tag{8}$$

where θ and $\emptyset$ represent the embedding spaces, and the potential of abstract feature representation can be explored in training to achieve better performance than the original Gaussian form. For more details, we use 3×3 convolution for both θ and $\emptyset$. Note that the Gaussian multiplication can be combined with $C(x)$ to exactly form into the expression of softmax activation.

The structure of NL-block in our model is as depicted in Fig. 2. The operation involved in a NL-block is given as:

$$z_i = NL(y_i) + x_i, \tag{9}$$

where $NL(\cdot)$ stands for NL-Block function and x_i denotes a residual connection. Considering that the matrix product process may take up a lot of memory, we add down-sampling operations (3×3 convolution with stride=2) to all θ, $\emptyset$ and g to downsize the feature maps, where the multiplied matrix can
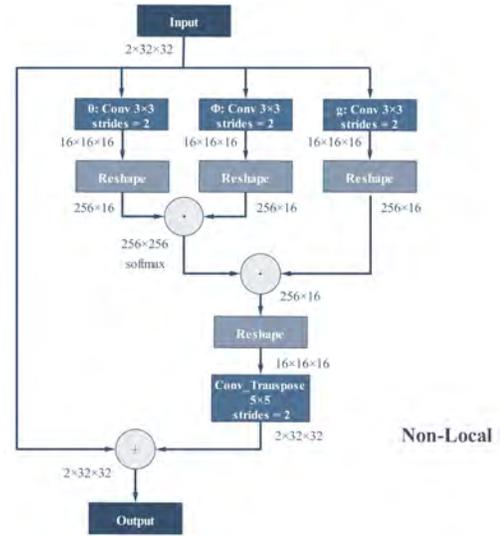

Fig. 2: The structure of a non-local block with embedded Gaussian version. "$\times$" denotes the channel matrix multiplication, and "+" denotes the element-wise sum. The Gaussian version can be implemented by removing θ and $\emptyset$. For convenience, the input size is set $N_c = N_t = 32$ in our model.

turn from the original shape 1024×1024 to the downsized shape 256×256. Meanwhile, the number of channels is increased to 16 in all θ, $\emptyset$ and g to compensate for the performance loss caused by down-sampling. The up-sampling operation (3×3 transposed convolution with stride=2) is used before the addition with the residual connection to recover the shape of feature maps. For regulation, batch normalization and ReLU activation layers are applied after all convolutions in the NL-block.

As shown in Eq. 7, the NL operation calculates the correlation $f(x_i, x_j)$ at all positions, so that it can directly pass information as well as extract correlation features between any two positions ($\forall j$) in one operation. It can also be regarded as a global convolution with its own self-correlation matrix as the kernel, which can cover the whole map, providing a global view for feature extraction. The correlation in long-distant positions can be efficiently captured, which makes NL-block more suitable for structural feature extraction. In this way, it only takes a few layers to achieve the best results without introducing too many parameters.

B. DS-RefineNet

To further improve the information flow between layers, we design a new densely connected convolutional network structure by utilizing dense connectivity [18], called DS-RefineNet.

The structure of the proposed DS-RefineNet is shown in Fig. 3. DS-RefineNet is based on the structure of RefineNet [7] with the same convolutions. Compared to skip connection implemented in RefineNet, we introduce direct connections from any layer to its all subsequent layers. This procedure can be described as:

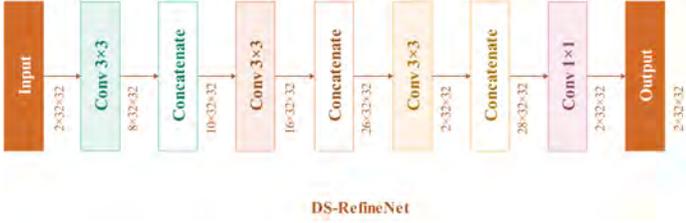$$x_l = H_l([x_0, x_1, \cdots, x_{l-1}]), \quad l = 1, 2, 3, \tag{10}$$

Fig. 3: The architecture of proposed DS-RefineNet

where H_l denotes the convolution operation, $[x_0, x_1, \cdots, x_{l-1}]$ represents the concatenation of the feature map in layers $0, \cdots, l-1$ and x_l denotes the output of lth convolution operation.

The structure of dense connectivity has a better effect on improving NN's flexibility than skip connection in RefineNet. In RefineNet, skip connection only allows the origin input to access the final output, while in DS-RefineNet, every two layers have directed connections and each layer can access the origin input as well as the output at any front node. When facing the NN's degradation problem, connections are densely everywhere in the structure of DS-RefineNet, providing nearly all the selections for the data to choose how to flow the indispensable layers and jump across those unnecessary ones. While in RefineNet, there are only two ways, i.e., data flows through all layers or jumps across them by the skip connection. All layers are regarded as a whole that the data cannot deal with any single layer, which greatly reduces the network efficiency.

The flexibility of neural networks in DS-RefineNet brings many benefits. On one hand, by creating a short path from early layers to later layers, it can largely alleviate the problem of vanishing gradient; on the other hand, this connection mode in DS-RefineNet makes the transmission of features and gradients more efficient, and the network is easier to train. Each layer can directly access the gradient from the loss function and the original input signal, resulting in implicit deep supervision. In the structure of decoder, the received compressed signal is first input in the non-local block to get a rough reconstruction of CSI. Then the rough reconstruction is fed into two DS-RefineNet blocks for more detailed refinements.

In every DS-RefineNet block, there are three 3×3 convolutional layers. The batch normalization layer followed with leaky ReLU activation is implemented before each convolutional layer. For the lth layer, we concatenate multiple inputs of $H_l(\cdot)$ as a tensor. The first, second and third convolutional layer generates 8, 16 and 2 feature maps, respectively. The final output of DS-RefineNet is the concatenation of each layer's output in the channel axis.

Due to the concatenation operation, the neural network in the second DS-RefineNet may become too wide. To further reduce the number of feature maps, a transition layer should be implemented between two DS-RefineNet blocks. Different from [18], we use 1×1 convolution as a substitute for the transition layer. Since the architecture of CSI feedback schemes is simpler than that of computer vision networks [12], the bottleneck before every 3×3 convolution will not be utilized

here to reduce the input feature maps. The implementation of DS-RefineNet has a significant impact on strengthening feature propagation and encouraging feature reuse. After these two DS-RefineNet blocks, the feature maps will pass through a 3×3 convolution and a sigmoid activation to output the final reconstructed CSI.

IV. SIMULATION RESULTS AND ANALYSIS

In this section, to verify the effectiveness of the proposed CSI compression feedback algorithm, experimental simulations are developed for indoor and outdoor scenarios, respectively, in an FDD massive MIMO system. Comparative analysis of our proposed scheme with several other methods of CSI feedback compression networks is also performed.

A. Parameter Setting

To train DS-NLCsiNet, the end-to-end learning for the encoder and decoder is applied. The output to DS-NLCsiNet is $\hat{\mathbf{H}}_i$, which can be expressed as:

$$\hat{\mathbf{H}}_i = f(\mathbf{H}_i; \Theta)$$
$$= f_{de}(f_{en}(\mathbf{H}_i; \Theta_{en}); \Theta_{de}), \quad (11)$$

where $\mathbf{H}_i$, $\Theta = \{\Theta_{en}; \Theta_{de}\}$ and $f = f_{de}(f_{en}(\cdot))$ denote the input channel matrices, parameter set and autoencoder network, respectively.

We use Adam optimizer with default setting to train our frameworks,and we choose the mean squared error (MSE) as loss function, which is given as:

$$L(\Theta) = \frac{1}{N} \sum_{i=1}^{N} \| f(\mathbf{s}_i; \Theta) - \mathbf{H}_i \|_2^2, \quad (12)$$

where $\|\cdot\|_2$ is the Euclidean norm, and N is the number of samples in the training data.

We use the COST2100 channel model [19] to generate the values of $\mathbf{H}$, considering two different scenarios: the indoor picocellular scenario at the 5.3GHz band and the outdoor rural scenario at the 300MHz band. The BS uses $\widetilde{N}_t = 32$ antennas and $N_c = 1024$ subcarriers. We reserve the first $N_c = 32$ columns of the channel matrix $\mathbf{H}$ since only they have non-zero values. Then $\mathbf{H}$ is truncated into the shape of 32×32. We use $100,000$ samples for training, $30,000$ for validation and $20,000$ for testing, respectively. The batch size, epochs and learning rate are set as 200, 1000 and 0.001, respectively.

B. Complexity Analysis

Compared with other state-of-the-art CSI feedback methods, the complexity analysis of the proposed DS-NLCsiNet is depicted in Table I, where the number of parameters and MACCs[1] stand for space and time complexity, respectively.

As shown in Table I, our frameworks do not introduce too many parameters when compared with CsiNet, while greatly enhancing the recovery quality of CSI. The increase in MACCs mainly comes from convolution layers. When CR is relatively

[1]MACC: multiply-accumulate operations. A multiplication operation and an additive operation count for one MACC operation.

TABLE I: The number of parameters and MACCS

	CR	1/4	1/8	1/16	1/32
Params	CsiNet [7]	2,103,904	1,055,072	530,656	268,448
	RecCsiNet [10]	28,331,104	22,300,512	19,481,824	18,121,632
	ConvlstmCsiNet [11]	28,326,904	22,296,312	19,477,624	18,117,432
	NLCsiNet	2,107,684	1,058,852	534,436	272,228
	DS-NLCsiNet	2,108,992	1,060,160	535,744	273,536
MACCs	CsiNet [7]	4,366,336	3,842,048	3,579,904	3,448,832
	RecCsiNet [10]	153,059,328	128,942,080	117,669,888	112,230,400
	ConvlstmCsiNet [11]	121,708,544	97,591,296	86,319,104	80,879,616
	NLCsiNet	10,424,448	9,375,872	8,851,548	8,589,440
	DS-NLCsiNet	11,497,120	10,448,544	9,924,256	9,662,112

small, the amount of computation of convolutional layers is more than that of dense layers. The model parameters and MACCs of DS-NLCsiNet are much lower than those of RecC-siNet and ConvlstmCsiNet, which improve the reconstruction accuracy at the cost of huge space and time complexity, since the dense layers in LSTM cell substantially increase the amount of computation.

C. Comparative Analysis

To gain insight into the proposed DS-NLCsiNet, the following CSI feedback methods are implemented in the same environment for comparison:

- **CsiNet [7]**: A well-known CSI sensing and recovery mechanism that applies RefineNet.
- **NLCsiNet**: Instead of using DS-RefineNet, this scheme only employs NL-blocks with RefineNet.
- **DS-CsiNet**: Instead of using NL-blocks, this scheme only employs DS-RefineNet.
- **DS-NLCsiNet**: This is the proposed DL-based CSI feedback scheme that combines NL-blocks and DS-RefineNet.

Two metrics can be used to evaluate the performance of different CSI feedback architectures as follows:

- *Normalized Mean Square Error (NMSE)*: It quantifies the difference between the original channel matrices and the recovered matrices, which can be defined as:

$$NMSE = \mathbb{E}\left\{ \frac{\left\| \mathbf{H} - \widehat{\mathbf{H}} \right\|_2^2}{\|\mathbf{H}\|_2^2} \right\}. \tag{13}$$

- *Cosine similarity*: It evaluates the similarity between the input and the output matrices by calculating cosine similarity of the channel response at each subcarrier, which can be defined as:

$$\rho = \mathbb{E}\left\{ \frac{1}{\widetilde{N}_c} \sum_{n=1}^{\widetilde{N}_c} \frac{\left| \widehat{\tilde{\mathbf{h}}}_n^H \tilde{\mathbf{h}}_n \right|}{\left\| \widehat{\tilde{\mathbf{h}}}_n \right\|_2 \left\| \tilde{\mathbf{h}}_n \right\|_2} \right\}. \tag{14}$$

We compare DS-CsiNet, NLCsiNet and DS-NLCsiNet with CsiNet. The corresponding NMSE and ρ of each network are summarized in Table II, where the best results are marked in bold font. Simulation results demonstrate that our proposed

TABLE II: NMSE in dB and cosine similarity ρ

CR	Methods	Indoor		Outdoor	
		NMSE	ρ	NMSE	ρ
$\frac{1}{4}$	CsiNet	-17.36	0.99	-8.75	0.91
	NLCsiNet	-22.51	0.99	-9.02	0.93
	DS-CsiNet	-20.01	0.99	-10.38	0.93
	DS-NLCsiNet	**-24.99**	**0.99**	**-12.09**	**0.95**
$\frac{1}{8}$	CsiNet	-12.70	0.96	-7.61	0.88
	NLCsiNet	-13.60	0.98	-7.69	0.89
	DS-CsiNet	-16.00	0.99	-7.66	0.89
	DS-NLCsiNet	**-17.00**	**0.99**	**-7.96**	**0.90**
$\frac{1}{16}$	CsiNet	-8.65	0.93	-4.51	0.79
	NLCsiNet	-9.76	0.95	-4.80	0.80
	DS-CsiNet	-9.17	0.94	-4.94	0.82
	DS-NLCsiNet	**-12.93**	**0.97**	**-4.98**	**0.81**
$\frac{1}{32}$	CsiNet	-6.24	0.89	-2.81	0.67
	NLCsiNet	-7.58	0.92	-3.18	0.71
	DS-CsiNet	-7.38	0.91	-3.22	0.70
	DS-NLCsiNet	**-8.64**	**0.93**	**-3.35**	**0.73**

DS-NLCsiNet outperforms the existing DL-based CSI feedback methods in terms of both NMSE and ρ. Compared with CsiNet, DS-NLCsiNet also provides significant gains, which mainly benefits from the use of NL-blocks in both of the encoder and decoder, and DS-RefineNet in the decoder. In addition, the performance comparison of DS-CsiNet, NLC-siNet and DS-NLCsiNet demonstrates that NL-blocks and DS-RefineNet can indeed enhance the performance of CSI feedback network, respectively.

Figure 4 plots original and some reconstructed CSI images in Pseudo-gray at different compression ratios for different CSI feedback schemes. The CSI images are randomly extracted from the test dataset in indoor and outdoor scenarios, respectively. In the first column from left, we show the original images of the CSI matrix in the angular-delay domain after performing the 2D-DFT and truncation operation. In the right three columns, we demonstrate some reconstruction samples along with the corresponding pseudo-gray plots of the strength of recovered matrices, which are reconstructed by using different CSI feedback schemes. Obviously, NLCsiNet and DS-NLCsiNet both outperform CsiNet, especially at low compression ratios. In addition, NLCsiNet and DS-NLCsiNet can recover the CSI in a more accurate way, and can also retain some feature which might be lost in CsiNet feedback procedure.
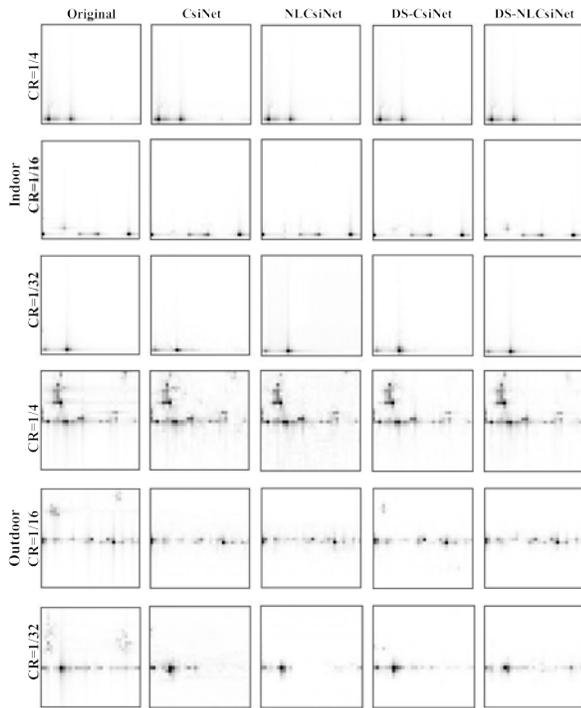
Fig. 4: The absolute value of original and reconstructed CSI images at different compression ratios (top: in indoor picocellular scenario; bottom: in outdoor rural scenario)

V. CONCLUSION

In this paper, we have proposed a novel DL-based CSI feedback scheme by utilizing non-local block and dense connectivity in feature extraction and RefineNet modules, respectively. Experimental results demonstrate that DS-NLCsiNet outperforms existing methods in terms of recovery accuracy and reconstruction quality. We believe this architecture has the potential for practical deployment on real FDD MIMO systems.

REFERENCES

[1] C. Li, P. Liu, C. Zou, F. Sun, J. M. Cioffi, and L. Yang, "Spectral-efficient cellular communications with coexistent one- and two-hop transmissions," *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 6765–6772, Aug 2016.

[2] M. S. Sim, J. Park, C. Chae, and R. W. Heath, "Compressed channel feedback for correlated massive MIMO systems," *Journal of Communications and Networks*, vol. 18, pp. 95–104, Feb 2016.

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

[4] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[5] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[6] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, 2017.

[7] 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, 2018.

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

[9] 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, pp. 416–419, April 2019.

[10] 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, 2018.

[11] X. Li and H. Wu, "Spatio-temporal representation with deep neural recurrent network in MIMO CSI feedback," *IEEE Wireless Communications Letters*, vol. 9, pp. 653–657, May 2020.

[12] Z. Lu, J. Wang, and J. Song, "Multi-resolution CSI feedback with deep learning in massive MIMO system," in *Proceedings of the IEEE International Conference on Communications*, June 2020.

[13] 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, pp. 2827–2840, Apr. 2020.

[14] J. Guo, X. Yang, C.-K. Wen, S. Jin, and G. Y. Li, "DL-based CSI feedback and cooperative recovery in massive MIMO," *arXiv preprint arXiv:2003.03303*, 2020.

[15] P. Liang, J. Fan, W. Shen, Z. Qin, and G. Li, "Deep learning and compressive sensing-based CSI feedback in FDD massive MIMO systems," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[16] J. Guo, J. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Compression and acceleration of neural networks for communications," *IEEE Wireless Communications*, vol. 27, pp. 110–117, Aug. 2020.

[17] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

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

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