A Reconstructed Autoencoder Design for CSI Processing in Massive MIMO Systems

Venkataramani Kumar*, Dalyana Mercado-Perez†, Feng Ye†, Rose Qingyang Hu‡ and Yi Qian§
*Department of Electrical and Computer Engineering, University of Dayton, Dayton, OH, USA
†Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, WI, USA
‡Department of Electrical and Computer Engineering, Utah State University, Logan, UT, USA
§Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE, USA
Emails: *tiruchirappallinarv1@udayton.edu, †{dmercadopere, feng.ye}@wisc.edu, ‡rose.hu@usu.edu, §yi.qian@unl.edu

***Abstract*—Massive multiple input multiple output (MIMO) systems are integral to next-generation wireless technologies due to their ability to meet the growing demands of throughput and support a plethora of applications. An efficient operation of massive MIMO requires accurate channel state information (CSI). In a frequency division duplex (FDD) MIMO system, the base station can rely on CSI feedback that user equipment (UE) estimates from downlink CSI from orthogonal pilot sequences. Recently, artificial intelligence (AI), i.e., deep learning approaches, have been introduced to compress and reconstruct CSI matrices at UE and the base station, respectively. However, these existing approaches still rely on channel estimation at the UE side, which introduces additional errors in the autoencoder design. To address these issues, we propose to implement the autoencoder that processes the pilot sequences directly to avoid excessive processing errors. Moreover, a higher compression can be achieved due to the lower error. Evaluation results demonstrate that the proposed scheme can significantly reduce the communication overhead by using a higher compression ratio while maintaining high CSI reconstruction performance in addition to lower bit error rates compared to the existing deep learning approach.**

I. Introduction

Massive multiple-input multiple-output (MIMO) systems in millimeter-wave (mmWave) channels are the emerging technologies in the fifth and future generations of wireless networks due to their ability to serve multiple users simultaneously with higher throughput, spectral, and energy efficiency [1]. However, for the successful implementation of such systems, it is pertinent to ensure accurate and timely estimation of channel state information (CSI) [2]. To address the absence of channel reciprocity in frequency-division duplex (FDD) MIMO systems usually, a base station usually relies on CSI feedback from user equipment (UE) that can estimate downlink channel properties from orthogonal pilot sequences [3]. However, the complexity of a massive MIMO system can introduce extremely high overhead in communications if the full CSI matrix is fed back. Since the massive MIMO CSI matrix in mmWave channels demonstrates strong sparsity, conventional approaches apply compressive sensing to reduce the overall CSI representations [4]. Although the communication overhead can be reduced significantly, compressive sensing demands substantial computing resources and time in finding the spatial correlation in the compression process, and also in solving optimization problems in the decompression process [5]. In addition, compressive sensing relies on random projection instead of fully understanding the channel structure. Furthermore, the iterative algorithms reduce the speed of the reconstruction process resulting in additional latency [6]. Similarly, signal-to-noice ratio (SNR) feedback-based channel estimation scheme requires lower feedback overhead when compared to the channel feedback [7]. Another salient feature is its applicability to both time division duplex (TDD) and frequency division duplex (FDD) systems. However, these methods can be inefficient in a massive MIMO system. Recently, artificial intelligence (AI), i.e., neural network approaches, have been proposed as a more computationally efficient way to compress and reconstruct the downlink CSI presentations at the UE and the base station, respectively, in an FDD massive MIMO system [5], [8]. However, these classic autoencoder-based CSI processing methods that try to reconstruct the original inputs encounter a few issues. First, the traditional approaches require a CSI recovery from pilot sequences, e.g., by using least square and maximum-likelihood method [9], which introduce additional computational complexity at the UE side. Meanwhile, the CSI used as the ground truth may be inconsistent to the practical implementation due to the imperfect channel recovery from the pilot sequence at the UE side. Such inconsistency could further exacerbate the reconstruction error the base station. Furthermore, due to the relatively high dimension of the raw CSI representation, the classic autoencoder designs have complex structures with high floating-point operations per second (FLOPS).

In this work, we intend to address the aforementioned challenges by reconstructing the CSI processing with a modified autoencoder design in a massive MIMO system. In the proposed approach, the modified autoencoder takes input of the received pilot sequences directly, instead of a fully recovered CSI representation, and extracts the features that are equivalent to the compressed CSI representations from the existing compressive sensing and the traditional autoencoder-based CSI processing methods, assuming perfect CSI recovery from the pilot sequences. Such an equivalence is validated as the CSI representations can be reconstructed at the base station through the decoder part of the modified autoencoder. Assuming imperfect CSI recovery from the pilot sequences,

the reconstructed process at the base station can eliminate the inconsistency between ground truth for training the autoencoder and actual inputs in practice. Note that the modified autoencoder design for CSI proposed in this work is different from the traditional ones in two aspects. First, the input to the encoder and the output from the decoding process are not intended to be the same. The input to the encoder is the received pilot sequence, while the output from the decoder is the reconstructed CSI representations. Second, the modified encoder is not necessarily for feature reduction, depending on the length of the input pilot sequence and the targeting compressed CSI representations. Nonetheless, the dimension of the extracted features would still be lower than that of a fully reconstructed CSI representation.

Our contributions in this work can be summarized as follows. The deep-learning assisted CSI processing in massive MIMO systems is reconstructed with a modified autoencoder design. By eliminating the full CSI recovery process at the UE side, the modified autoencoder enhances the consistency between the training and practical implementation. Evaluation results demonstrate that the proposed concept of reconstruction can achieve comparable accuracy with a much less computing complexity at higher compression ratio. The remainder of the paper is organized as follows. Section II describes the existing CSI estimation and feedback approaches. Section III describes the deep learning channel estimation techniques as well as our proposed scheme design. Section IV depicts the evaluation results. The conclusion and future works are outlined in Section V.

II. System Model and Preliminaries

A. Studied system model.

For better illustration, the notations used in the rest of the paper are listed in Table I. The operations, $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^H$, Tr$(\cdot)$, $|\cdot|$, $\mathbb{E}\{\cdot\}$, and $(\cdot)^\dagger$ denote the conjugate, transpose, conjugate transpose, trace, absolute operator, expectation and matrix pseudo-inverse, respectively.

TABLE I: Notations used in this work.

Notation	Remarks
N_t	Number of transmitting antennas
N_r	Number of receiving antennas
L	Length of channel taps
N_c	Number of subcarriers
y_n	Signal received at the n_{th} subcarrier
x_n	Transmitted symbol
$h_{r,m}$	Channel between the m^{th} transmit antenna and the r^{th} receive antenna
X_P	Pilot signals
Y_P	Received pilot signals
$\hat{H}$	Estimated channel state information

In this study, we consider a single-cell massive MIMO-Orthogonal frequency division multiplexing (OFDM) with N_t ($N_t >> 1$) antennas at the base station (BS) and receiver antenna at the user equipment (UE) as shown in Fig. 1. Since,

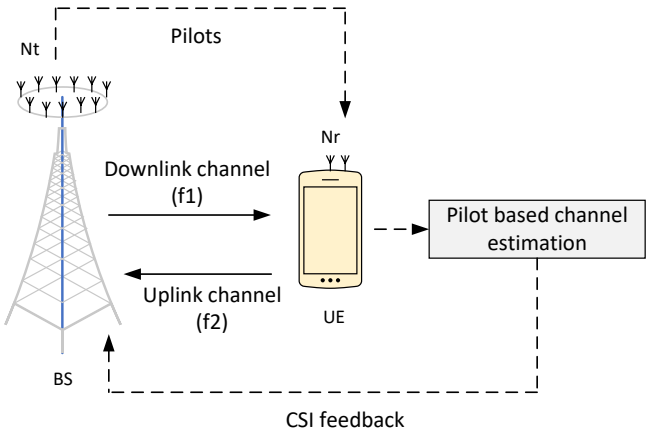

Fig. 1: Overview of the studied system.

the OFDM comprises N_c subcarriers, the received signal the n_{th} sub-carrier can be written as

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

where $\hat{\mathbf{h}}_n^H$ and $\mathbf{v}_n$ refer to the channel frequency response vector and the precoding vector at the n_{th} subcarrier while x_n refers to the transmitted symbol and w_n refers to the white additive gaussian noise. The CSI matrix in the spatial-frequency domain can be represented in the matrix form as follows:

$$\hat{\mathbf{H}} = [\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, \ldots, \hat{\mathbf{h}}_n]^H \in \mathbb{C}^{N_t \times N_c}. \tag{2}$$

The BS computes the precoding vector $\mathbf{v}_n \in \mathbb{C}^{N_t \times 1}$ through singular value decomposition (SVD) based on CSI.

B. Preliminaries on Compression and Feedback Process

The downlink CSI matrix of such a system can be estimated in compression and feedback processes. To begin with, a block of pilot signals is transmitted by the BS to the UE in the downlink channel. Note that each time slot is subdivided into mini-time slots of which the first few are allocated for pilot sequences [10]. Given a downlink pilot sequence, it is assumed that the channel estimation can be performed by the UE using channel estimation methods such as least squares [11], as follows:

$$\hat{\mathbf{H}}_{\text{LS}} = \frac{1}{X_P} Y_P. \tag{3}$$

Note that the channel estimation process is required in most existing CSI feedback scheme designs. The $\hat{H}_{\text{LS}}$ is then compressed and fed back to the BS using the uplink channel. The number of feedback parameters is $(2N_cN_t)$. Excessive feedback requires greater resources such as bandwidth. The compression process is to reduce communication overhead. However, the computational complexity in the channel estimation increases proportionally with N_t [12]. In addition, the downlink channel estimation at UE can increase processing delay and reduces the time available for actual data transmission [13]. Depending on the compressing technology, the BS

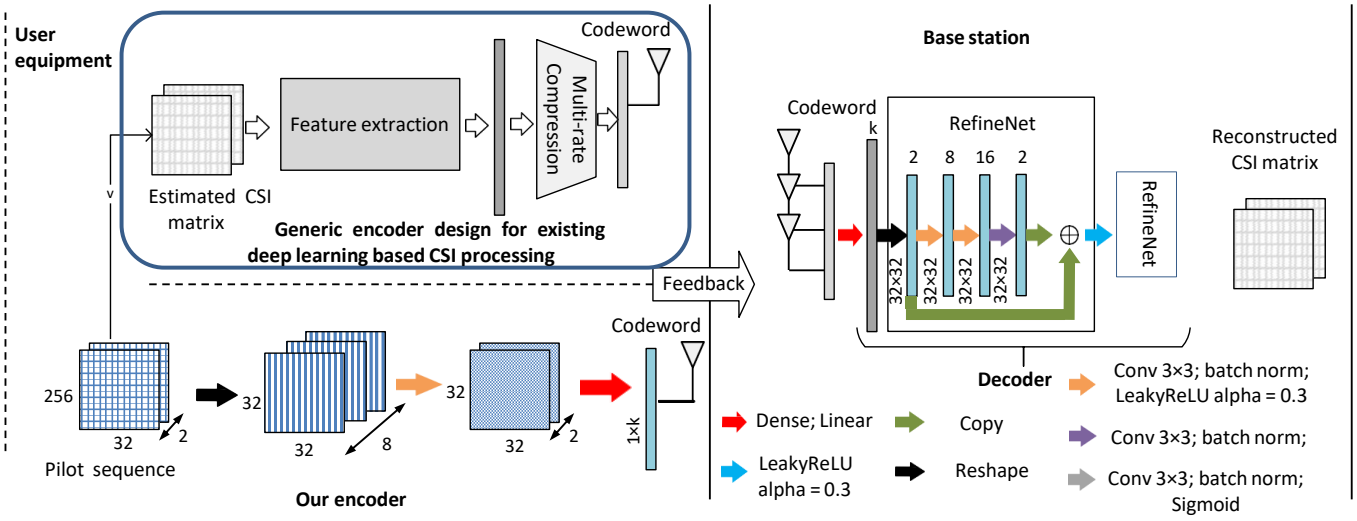

Fig. 2: Overview of the deep learning-based CSI estimation and feedback process.

reconstructs the downlink CSI matrix by decompressing the feedback information.

C. Deep Learning based CSI Processing Techniques.

The traditional channel estimation schemes described previously suffer from two major drawbacks namely computational complexity and channel overheads. Though the decomposition methods such as the QR-Gram Schmidt and QR-Givens Rotation method optimizes the complexity, they still require higher power and massive storage requirement [1]. Moreover, due to the non-convex nature of the estimation problems, it is challenging to develop analytical solutions. However, machine learning techniques such as the Grechberg-Saxton (GS) algorithm can be utilized to resolve non-convex and non-linear problems. But, the performance of such algorithms is unsatisfactory due to their higher iteration requirements to compute the channel estimate. Such methods cannot be frequently utilized in time-sensitive real-time applications [14]. The aforementioned challenges necessitate the need to utilize deep learning techniques to estimate the channel.

For example, CsiNet is one of the first CSI sensing and recovery mechanisms based on an autoencoder design [6]. As demonstrated in Fig. 2, the encoder design of CsiNet comprises a convolutional layer and a dense layer. the input to the encoder of CsiNet is the truncated sparsified $\hat{\mathbf{H}}$, e.g., from Eq. (2) in the angular domain, s.t.,

$$\mathbf{H} = \mathbf{F}_d \, \hat{\mathbf{H}} \, \mathbf{F}_a^H, \tag{4}$$

where $\mathbf{F}_d$ and $\mathbf{F}_a$ are $\hat{N}_c \times \hat{N}_c$ and $N_t \times N_t$ discrete Fourier transform (DFT) matrices, respectively. In specific, the real and imaginary components of $\mathbf{H}$ are applied separately as inputs to the encoder. The 3×3 dimensional kernel convolves with the inputs to generate two feature maps respectively. The vectors, which are the reshaped feature maps, are applied as inputs to the dense layer to generate an M-dimensional codeword. Such compression or the transformation into codewords is based on a compression ratio γ, e.g., 1/4 or 1/8 of the original input size. The M-dimensional codeword ($\mathbf{s}$) is transmitted back to the BS. The decoder in the BS comprises a dense layer and RefineNet units. The dense layer in the decoder generates two outputs each symbolizing the real and imaginary parts estimates of $\mathbf{H}$. The generated outputs are applied to the RefineNet units, a conglomeration of four convolutional layers, with a kernel size of 3. We can see that the second and the third layers of the RefineNet generate 8 and 16 feature maps while the final layer produces the reconstructed channel matrix $\mathbf{H}'$. Ideally, the RefineNets are optimized to ensure that their outputs are almost the same as that of the residual between its input and ground truth and can be expressed as follows:

$$\mathbf{H}'_{\text{res}} = \mathbf{H} - \mathbf{H}'_{\text{in}}, \tag{5}$$

where $\mathbf{H}$ refers to the original channel matrix and $\mathbf{H}'_{in}$ refers to initial estimated of $\mathbf{H}$ and $\mathbf{H}'_{res}$ is the expected residual. A batch normalization layer is provided to every layer in the network. It may be noted here that the rectified linear unit (ReLu) is the activation function, s.t., $\text{ReLu}(x) = \max(x, 0)$.

There have been several CSI processing schemes developed based on CsiNet [6], [15]. The salient features of CsiNet and its variants are as follows. The encoder entirely relies on the training data to understand the channel structure and with this knowledge, it compresses the representations into codewords. Similarly, the inverse transformation of the codewords into the channel matrices performed by the decoder is non-iterative and faster compared to the traditional approaches, e.g., compressive sensing. However, the error difference between the reconstructed and estimated channel increases with the compression ratios. Our proposed work envisions reducing this error difference even at more aggressive compression ratios.

III. Simplified CSI Processing Scheme

A. Overview of the Proposed Scheme Design

An overview of the proposed simplified CSI processing scheme is shown in Fig. 2. Three major functionalities can be seen in our model namely the pilot processing, feature extraction, and codeword generation. The first step is pilot processing. The BS transmits a block of pilots proportional to the N_t to the UE. Then, the UE executes the encoding part to generate a compressed codeword. Compared to the existing deep-learning-based approaches, our design mainly simplifies the encoder part by eliminating the CSI estimation process at the UE side. The direct processing of pilots ensures that the channel is directly constructed at the BS thereby reducing the resource utilization at the UE. Another salient feature of our proposed scheme is the consideration of original channels instead of the estimated channel matrix as the ground truth. Such consideration will reduce the channel estimation error between the estimated and reconstructed channels. The codeword is fed back to the BS which is then decoded it to construct the entire channel matrix.

B. Simplified Encoder Design

Without reconstructing CSI matrix at the UE side, the inputs to the encoder are the reshaped received pilot sequences from the BS. The first layer, which is the convolution layer, extracts the features required to construct a complete CSI matrix using a filter of size 3×3. Let $\mathbf{P} = [\mathbf{p}_1,\ \mathbf{p}_2,\ \mathbf{p}_3,\ \ldots,\ \mathbf{p}_n]$ refer to the received pilot sequence block while $\mathbf{S} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n]$ denote the vector of extracted features f_i, $i \in \mathbb{N}$. The fully connected layer with k neurons compresses the $\mathbf{S}$ to a lower dimension output referred to as codeword $\mathbf{s}$, denoted as follows:

$$\mathbf{s} = f_{\text{en}}(\mathbf{S}). \tag{6}$$

The γ of the encoder can be calculated as

$$\gamma = M/(2N_tN_c), \tag{7}$$

where the denominator $2N_tN_c$ is referred to as the feedback parameters.

C. Decoder Design

For a better analysis of the proposed concept of eliminating CSI estimation at the UE side, the decoder part at the BS side generally follows the existing scheme design. The first layer in the decoder is the fully-connected layer with $2 \times N_t \times N_c$ neurons. The codeword $\mathbf{s}$ when applied as input to this layer results in the generation of two matrices of dimension $N_c \times N_t$. These two matrices serve as an initial estimate of the real and imaginary parts of the original channel matrix $\mathbf{H}$. The RefineNets fine tunes the initial estimate to reconstruct the channel matrix $\mathbf{H}$', s.t.,

$$\mathbf{H}' = f_{\text{de}}(\mathbf{s}). \tag{8}$$

It may be noted here that RefineNet provides twin benefits. To begin with, the output size of the RefineNet is the same as that of the original channel matrix $\mathbf{H}$. In addition to this, it alleviates the vanishing gradient problem arising due to the multiple non-linear transformations by providing shortcut connections [6]. A batch normalization layer is provided to every convolution layer and a sigmoid activation layer is used to scale the outputs within the range [0,1].

D. Modified Loss Function

The end-to-end learning is adapted to tune the kernel and bias values of both the encoder and decoder. The set of parameters can be defined mathematically as, $\Theta = \{\Theta_{\text{en}}, \Theta_{\text{de}}\}$. The input is $\mathbf{H}_i$, while the output is the recovered channel at the BS is $\hat{\mathbf{H}}_i$. Mathematically, $\hat{\mathbf{H}}_i = f(\mathbf{H}_i;\ \Theta)$ [6]. It may be noted here that both inputs and outputs to this model are normalized and their values lie in the range [0, 1]. Since the output from the decoder part does not map directly to the input to the encoder part, the loss function is modified to minimize the mean squared error (MSE) between the reconstructed channel matrix and the ground truth of channel matrix as follows:

$$L(\Theta) = \frac{1}{T}\sum_{i=1}^{T} ||f(\mathbf{S}_i;\Theta) - \mathbf{H}_i||_2^2, \tag{9}$$

where $\|\cdot\|_2$ is the Euclidean norm and T is the total number of samples utilized in the training set.

IV. Evaluation results

A. Dataset and Evaluation Settings.

The dataset utilized for training and testing the model comprised pilot sequences and the original channel matrices. In our work, we utilized the open-source channel matrix dataset [6]. The dataset was generated for two scenarios namely indoor and outdoor using the COST 2100 channel. Each channel was of dimension $32\times32\times2$. In this work, we utilized indoor datasets alone since the related work demonstrated better performance here [6]. The dataset and settings are chosen the same for comparing directly with the existing deep-learning CSI processing scheme CsiNet [6], interchangeable with the 'benchmarking approach' in this section. Based on the channel matrices, the corresponding pilot sequences were generated using the classic approach [16]. Without further notice, the settings utilized for the pilot generation are summarized in Table II. The deep learning-based autoencoder designs were implemented in the Pytorch environment on a workstation running an 8-core Intel(R) Xeon(R) CPU @ 2.40 GHz, 32 GB RAM, and an Nvidia GTX 1080 Ti GPU card.

B. Evaluation Results

Our simplified autoencoder model was trained and tested on the dataset comprising 50,000 samples. The dataset was bifurcated into training and testing datasets each with 30,000 and 20,000 samples, respectively. The pilot sequences originally of size $256 \times 32 \times 2$ were reshaped to $32 \times 32 \times 8$ and applied as inputs to the first convolution layer of the encoder. The input was convoluted with a kernel of size 3. The fully-connected layer transforms the vectors obtained from the previous layer into codewords of different dimensions (k)

resulting in different compression ratios (γ). For consistency with the existing approach, the compression ratios considered in this work were 1/4, 1/16, and 1/32. In the decoding part, the codeword from the encoder is applied as input to the fully-connected or dense layer to transform it back to the vectors of suitable size. The multiple convolutional layers with a uniform kernel size of 3, processes, refines and reconstruct the entire channel. The epochs, learning rate, and batch size are set as 1000, 0.001, and 500 respectively. The existing approach for comparison is built using standard settings of autoencoder, where the output is mapped to the input for minimum error. In the testing phase, it is assumed that the UE executes the encoder part based on the received pilot sequence, or the estimated CSI from the pilot sequence, using the simplified and benchmarking approaches, respectively.

Fig. 3 depicts the normalized mean square error (NMSE) of both simplified and benchmark approaches at different compression ratios and at multiple SNR values. In general, a simple examination of the results provides the following inferences. First, the performance decreases with the decrease in the SNR values. For instance, at SNR and γ of -20 dB, and 1/4 respectively the mean MSE of the simplified approach stands at around -5 dB when compared to about -30 dB at 20 dB SNR. Second, the performance decreases with the increase in the compression ratio regardless of the SNR. For instance, at an SNR of 20 dB, the MSE values of the simplified approach are approximately -35 dB, -30 dB, and -27 dB at γ of 1/4, 1/8, and 1/32 respectively. In other words, the accuracy of the channel reconstruction process decreases with the increase in the compression ratio, and decrease in the SNR. In terms of performance, our simplified approach outperforms the benchmarking approach. For instance, at an SNR of 20 dB and γ of 1/4, the NMSE of our simplified approach is approximately -35 dB when compared to - 28dB of benchmarking approach. The relatively better performance of our approach can be attributed to the direct processing of pilot signals. Fig. 4 depicts the bit error rate (BER) analysis of both approaches at different SNRs. It may be noted here that the transmitted data symbols are precoded using the zero-forcing (ZF) method to reduce the BER [17]. The precoding improves

TABLE II: Simulation settings for the MIMO system.

Settings	**Remarks**
$N_t \times N_r$	32 × 32
Pilot spacing (P)	4
Pilot length	1 × 32
Subcarriers (K)	1024
Guard Interval (G)	0.25
Modulation (Mod)	QPSK
Channel taps (L)	2
Normalized signal power	1
SNR	[-20, -15, -10, 0, 10, 15, 20]
Compression ratio γ	1/4, 1/16, 1/32
Carrier frequency	2.412 GHz
# Samples	50,000

the accuracy and reliability of the transmitted symbols. From BER evaluation results, it is feasible to draw two inferences. To begin with, the BER decreases with the increase in SNR regardless of the compression ratio. For instance, at γ of 1/4, the BER of our simplified approach decreases from 0.037 at -10 dB SNR to 0.026 at 10 dB. Furthermore, the decrease in the BER decreases with the increase in the compression ratio. For instance, at 15 dB the BER of our simplified approach is 0.022, 0.037, and 0.0461 respectively. In the future, we intend to evaluate the BER by applying different precoding schemes such as maximum ratio transmission (MRT).

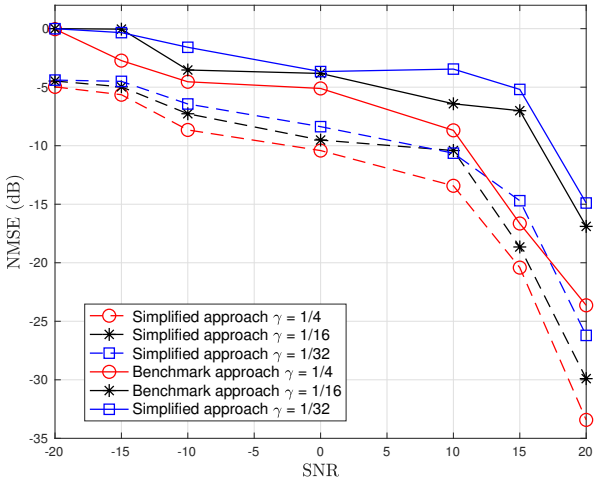

Fig. 3: NMSE of channel recontructions.

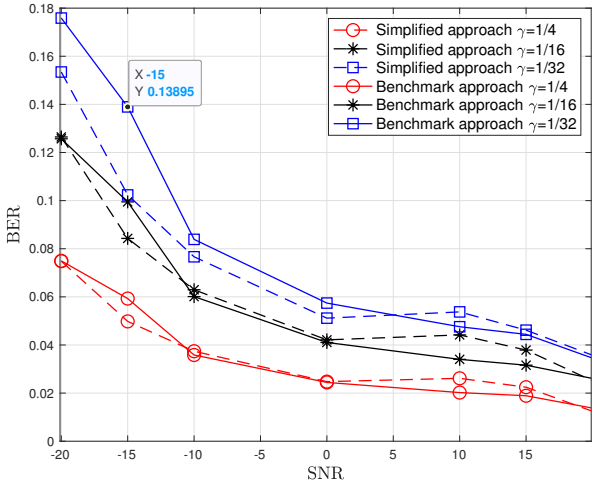

Fig. 4: BER demonstration using the reconstructed channels.

Table III further compares the parameters, and FLOPS of our simplified approach with that of the benchmarking approach. Note that the number of FLOPs required by the simplified approaches are higher than that of the benchmarking ones at the same compression rates. It is mainly because the

TABLE III: Results of simplified autoencoder design and the benchmarking autoencoder design.

Compression Ratio (γ)	Flops (M)			Parameters (K)			NMSE (dB) at SNR = 20 dB	
	Simplified Encoder	Benchmark Encoder	Decoder	Simplified Encoder	Benchmark Encoder	Decoder	Simplified autoencoder	Benchmark autoencoder
1/4	1.70	0.59	94.89	524.36	518.60	13.20	-33.43	-23.63
1/16	1.31	0.33	94.63	262.22	259.09	13.42	-29.07	-16.89
1/32	1.24	0.20	94.50	131.14	129.55	26.53	-26.20	-14.89

higher input dimension of the pilot sequence comparing to the CSI matrix in the proposed approach. Although a higher FLOP count indicates a greater complexity, the benchmarking approaches require an extra step of CSI reconstruction, e.g., using Eq. (3), which introduces additional complexity that can be more than the extra FLOPs in the proposed approach. The exact complexity analysis will be conducted in the future work. Meanwhile, the parameters in both these models are nearly comparable. For example, the number of parameters in our simplified encoder is 524.36 K which is comparable to 518.16 K in the CsiNet.

V. Conclusion and future works

The evolution of the MIMO relies on the ability to estimate accurate CSI with lower computational complexity, reduced and transmission overheads. The traditional approaches offer channel estimation but suffer from high computational complexity with the increase in the transmit antennas in a MIMO system. The deep-learning assisted CSI processing schemes have been introduced to address the issues. However, the standard concept autoencoder used in these existing approaches introduces inevitable error due to CSI reconstruction at the UE side. In this work, we proposed a simplified structure that processes the received pilot sequences directly as the inputs, which is compressed into a low-dimension codeword. The codeword is decompressed at the decoder to retrieve the channel. The modified autoencoder structure can mitigate the CSI reconstruction error in the process. The proposed model was validated on the same open-source dataset used by the benchmarking approach. The evaluation results demonstrated that the simplified approach can achieve a much lower reconstruction error, hence a lower BER compared to the existing one. Moreover, the simplified approach achieved the same level of reconstruction accuracy with all the tested compression ratios, while the benchmarking approach returned much lower reconstruction accuracy given a higher compression ratio. In other words, the transmission overhead in the feedback process can be effectively reduced with the schemes developed in this work at a reduced NMSE when compared to the benchmarking approach. In future work, more practical scenarios will be evaluated to further optimize the simplified autoencoder design. Evaluations will also be conducted on hardware testbeds.

Acknowledgment

This project was supported by the U.S. National Science Foundation under the grants NSF ECCS-2336234, ECCS-2139508, and ECCS-2139520.

References

[1] C. Zhang, X. Liang, Z. Wu, F. Wang, S. Zhang, Z. Zhang, and X. You, "On the low-complexity, hardware-friendly tridiagonal matrix inversion for correlated massive mimo systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6272–6285, 2019.

[2] M. Boloursaz Mashhadi and D. Gündüz, "Deep learning for massive mimo channel state acquisition and feedback," *Journal of the Indian Institute of Science*, vol. 100, no. 2, pp. 369–382, 2020.

[3] M. B. Mashhadi and D. Gündüz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for mimo-ofdm systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6315–6328, 2021.

[4] J. Ling, D. Chizhik, A. Tulino, and I. Esnaola, "Compressed sensing algorithms for ofdm channel estimation," *arXiv preprint arXiv:1612.07761*, 2016.

[5] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Overview of deep learning-based csi feedback in massive mimo systems," *IEEE Transactions on Communications*, vol. 70, no. 12, pp. 8017–8045, 2022.

[6] 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.

[7] S. Abeywickrama, T. Samarasinghe, C. K. Ho, and C. Yuen, "Wireless energy beamforming using received signal strength indicator feedback," *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 224–235, 2017.

[8] S. Han, Y. Oh, and C. Song, "A deep learning based channel estimation scheme for ieee 802.11p systems," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[9] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5g wireless communications: A deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2018.

[10] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive mimo systems," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1228–1231, 2019.

[11] A. K. Gizzini, M. Chafii, A. Nimr, and G. Fettweis, "Enhancing least square channel estimation using deep learning," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[12] Y. Zeng and R. Zhang, "Optimized training design for wireless energy transfer," *IEEE Transactions on Communications*, vol. 63, no. 2, pp. 536–550, 2014.

[13] H. Suzuki, R. Kendall, C. K. Sung, and D. Humphrey, "Efficient and accurate channel feedback for multi-user mimo-ofdma," in *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE, 2017, pp. 1–6.

[14] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep learning based channel estimation for mimo systems with received snr feedback," *IEEE Access*, vol. 8, pp. 121 162–121 181, 2020.

[15] 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, 2020.

[16] I. Barhumi, G. Leus, and M. Moonen, "Optimal training design for mimo ofdm systems in mobile wireless channels," *IEEE Transactions on signal processing*, vol. 51, no. 6, pp. 1615–1624, 2003.

[17] A. Wiesel, Y. C. Eldar, and S. Shamai, "Zero-forcing precoding and generalized inverses," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4409–4418, 2008.