Learning Variable-Rate Codes for CSI Feedback

Heasung Kim, Hyeji Kim, and Gustavo de Veciana
Department of Electrical and Computer Engineering
University of Texas at Austin

***Abstract*—We observe that current Deep Learning (DL)-based Channel State Information (CSI) encoder and decoder architectures achieve a distortion which is highly channel-dependent. To exploit this, we propose a novel learning-based variable-rate coding scheme to reduce overheads associated with CSI feedback. To that end, we propose an architecture which combines (*a*) training an efficient predictor for the distortion rate tradeoffs achievable for a given channel, and (*b*) optimization of a decision logic which allocates rates based on the predicted distortion. We evaluate our approach on various wireless channel datasets including the 3GPP 3D channel model and COST2100 with Massive MIMO channel model, and show significant potential reductions of up to 20% in the CSI feedback overhead.**

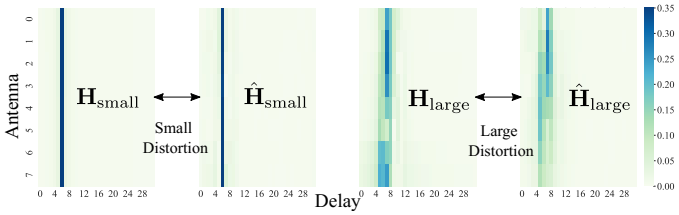

Fig. 1. The ground-truth CSI matrices $(\mathbf{H}_{\text{small}}, \mathbf{H}_{\text{large}})$ and the recovered CSI matrices $(\hat{\mathbf{H}}_{\text{small}}, \hat{\mathbf{H}}_{\text{large}})$ from a 64-bit autoencoder. Even using the same number of bits for CSI feedback, the achievable distortion varies greatly depending on the CSI pattern.

I. Introduction

CSI feedback plays a crucial role in the realization of wireless network coordination and performance. To increase the transmission efficiency, modern communication systems have adopted *implicit feedback* schemes, including information such as a Rank Indicator (RI), a Precoding Matrix Indicator (PMI), and a Channel Quality Indicator (CQI) [1]. Optimal codebook design for implicit feedback has been intensively studied with the goal of maximizing the achievable rate. The Grassmannian packing problem, i.e., the problem of designing a limited number of codewords to maximize the minimum distance between them was extensively studied in [2], [3]. In addition, Random Vector Quantization (RVQ), a method for selecting a codeword having the highest similarity in a randomly generated codebook, has been studied theoretically for various environments [4], [5]. However, because implicit CSI feedback alone cannot inform the Base Station (BS) of the channel gain and phase corresponding to each subcarrier, it gives the BS limited flexibility in choosing the beamforming vector to minimize interference and maximize rate.

In contrast to the implicit CSI, the *explicit CSI feedback* with raw channel information regardless of the precoding method gives the BS more flexibility for optimal precoding. However, in modern communication systems, since the BS communicates with the user equipment (UE) through multiple transmit antennas on tens to thousands of subcarriers, transmitting raw CSI is impractical in that it requires enormous computational overhead and radio resources to the UE.

To enable explicit CSI feedback with a linear complexity, the application of Deep Learning (DL) techniques has been proposed for the design of CSI feedback encoder and decoder [6]. DL-based approaches have outperformed existing methods based on the compressed sensing ideas in terms of time complexity and distortion minimization. In recent years, research aimed at improving performance by designing effective neural networks has been intensively conducted. In [7], the Long Short-Term Memory (LSTM) structure was utilized to exploit temporal correlation between successive channels. The authors of [8] proposed the LSTM-based structure of recurrent compression and decompression modules and designed a network using only a small number of parameters to reduce the overhead caused by the complex or deep architecture of the encoder/decoder. A convolutional LSTM-based structure [9] and the inception neural network-based structure [10] have also been proposed. In order to account for realistic wireless communication systems, neural structures capable of compressing CSI into binary information have been proposed [11], [12]. In [11], a structure that can generate codes with various feedback rates from the same encoder structure was proposed along with the quantization technique. A magnitude-adaptive phase quantization framework was proposed in [12], which is robust for high compression schemes.

Recent studies on DL-based CSI feedback have focused on maximizing performance under a given number of feedback bits, and have not provided a criterion for how many bits to allocate for a given CSI. The data-driven DL-based encoder/decoder are trained with the goal of maximizing only the average performance in a given dataset, and as a result, the distortion that occurs in the encoding and decoding process varies enormously across the CSI. Fig. 1 shows two CSI matrix samples compressed and decoded by a 64-bit autoencoder, which will be described in Section II. Even with compression using the same number of bits, some samples show very low distortion $(\mathbf{H}_{\text{small}})$, while some samples show large distortion $(\mathbf{H}_{\text{large}})$. This imbalance in compression distortion can bring

This work was supported by InterDigital, an affiliate of the 6G@UT center within the Wireless Networking and Communications Group at The University of Texas at Austin. This work work was supported in part by NSF Awards CNS-2008824, CNS-1910112, and ONR Award N00014-21-1-2379.

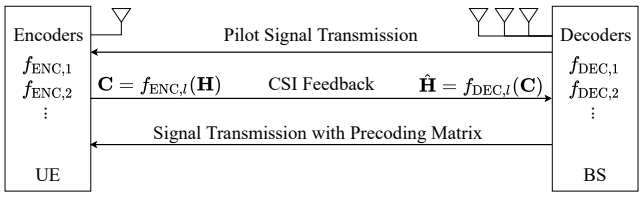

Fig. 2. A system model. The UE can feed back the CSI matrix $\mathbf{H}$ by selecting one of the encoders with various feedback bits. The BS has the estimated CSI matrix $\hat{\mathbf{H}}$ through the corresponding decoder. Based on the estimated CSI matrix, the BS designs a precoder and transmits a desired signal.

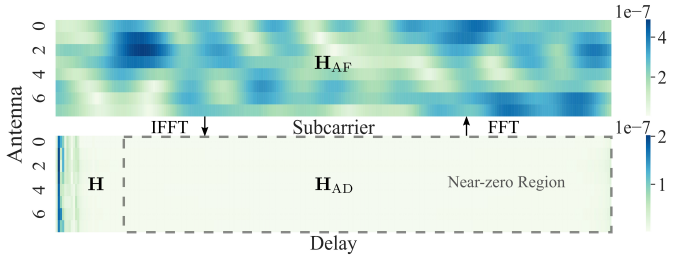

Fig. 3. The matrix in the first row represents the absolutes value of a randomly selected CSI matrix $\mathbf{H}_{\text{AF}} \in \mathbb{C}^{8\times 257}$. Through the IFFT in the frequency domain, we can obtain the information on the spatial-delay domain, $\mathbf{H}_{\text{AD}}$. To reduce computational load, the near-zero region is cropped and the remaining part is used as an ground-truth CSI $\mathbf{H} \in \mathbb{C}^{8\times 32}$.

undesired results, for example, an excessive number of feedback bits can be allocated for easy-to-compress CSI, or a small number of bits can be dedicated to the CSI matrix instances with large distortion.

To effectively solve this problem, we propose a novel approach, *Variable-Rate Code* (VRC) for CSI compression, that allocates an appropriate number of bits to compress a given CSI matrix based on distortion from lossy compression. To the best of our knowledge, this approach represents the first data-driven strategy to generate variable-rate codes for CSI feedback. Our main contributions are as follows:

- We design a lightweight distortion estimator that predicts how much distortion will occur when CSI is compressed and decoded by various DL-based autoencoders.
- We propose an evolutionary strategy-based codeword length selector that determines the optimal feedback codeword length. By allocating many bits for compression only when a large amount of distortion is predicted, we show that our proposed scheme significantly reduces the average codeword length required to achieve the desired distortion.
- To demonstrate the versatility of the proposed method, we show that the number of feedback bits can be effectively reduced by the proposed method for the existing autoencoders and dataset [6], as well as the autoencoder structure and dataset we provide.

II. System Model and Fixed-Rate CSI Feedback

We consider a Multiple Input Multiple Output (MIMO) channel where a BS communicates with a User Equipment (UE) over N_{sc} subcarriers. The BS and the UE have M antennas and a single antenna, respectively. On the n-th subcarrier, the AWGN channel output $y_n \in \mathbb{C}$ given the input signal $x_n \in \mathbb{C}$ is modelled as follows.

$$y_n = \mathbf{h}_n^{\text{H}} \mathbf{v}_n x_n + z_n, \qquad 1 \le n \le N_{sc}, \tag{1}$$

where $\mathbf{h}_n \in \mathbb{C}^{M\times 1}$ is a channel vector and $z_n \in \mathbb{C}$ is the corresponding AWGN. $\mathbf{v}_n$ is a normalized precoding vector whose power is 1, $||\mathbf{v}_n||^2 = 1$. We represent the CSI of all subcarriers as a single matrix as $\mathbf{H}_{\text{AF}} = [\mathbf{h}_1, \mathbf{h}_1, ..., \mathbf{h}_{N_{sc}}]$ and $\mathbf{H}_{\text{AF}} \in \mathbb{C}^{M\times N_{sc}}$. Note that $\mathbf{H}_{\text{AF}}$ is a CSI matrix in the spatial-frequency domain.

A. CSI Preprocessing

To reduce the computational overhead of the UE, the input dimension of the CSI encoder f_{ENC} needs to be reduced. One of the widely used efficient CSI preprocessing methods is transforming a CSI matrix by using the Fourier transform to handle CSI in the spatial-delay domain [6], [7]. For the sake of readability, we briefly describe the method and provide Fig. 3. CSI in the spatial-frequency domain can be transformed to spatial-delay domain via the Inverse Discrete Fourier Transform (IDFT) or Inverse Fast Fourier Transform (IFFT). The $M \times N_{\text{sc}}$ matrix in the first row of Fig. 3 represents the absolute values of a given channel $\mathbf{H}_{\text{AF}}$. The matrix below is the absolute value of the CSI in the spatial-delay domain $\mathbf{H}_{\text{AD}}$, where $\mathbf{H}_{\text{AD}} = \mathbf{H}_{\text{AF}}\mathbf{T}_{\text{IDFT}}$ and $\mathbf{T}_{\text{IDFT}}$ is the IDFT matrix. The CSI matrix in the spatial-delay domain $\mathbf{H}_{\text{AD}}$ contains some dominant column vectors in the delay domain, and the columns corresponding to the high delay components are near-zero vectors. By truncating the near-zero region and dealing with the cropped remaining matrix $\mathbf{H}$, which has a size of $M \times N_{\text{nz}}$, encoders and decoders can be designed to handle matrices with relatively small dimension[1].

B. CSI Feedback Encoder and Decoder

The UE uses an encoder to compress $\mathbf{H}$ into binary information and sends it to the BS. BS recovers the information using a decoder. Fig. 2 represents the system model and the process. We refer to the entire structure combining encoder and decoder as an autoencoder, and exhibit the overall architecture in Fig. 4.

Encoding. The ground-truth channel $\mathbf{H}$ is used as input and fed to three independent convolution layers. The numbers and tuples in the convolution layer description refer to the filter dimension and filter size, respectively. When a convolution layer and a dense layer are connected, the output of the convolution layer is vectorized. The dense layers are represented with their depth and an activation function.

[1]In the Massive MIMO considered in [6], [7], where the number of transmitter antennas is large as $M \gg 1$, 2D-DFT is utilized to obtain a sparse matrix in the angular-delay domain. These approaches to obtain a sparse matrix exploit the duality between the spatial-frequency CSI and the angular-delay CSI which has been studied in depth by [13].

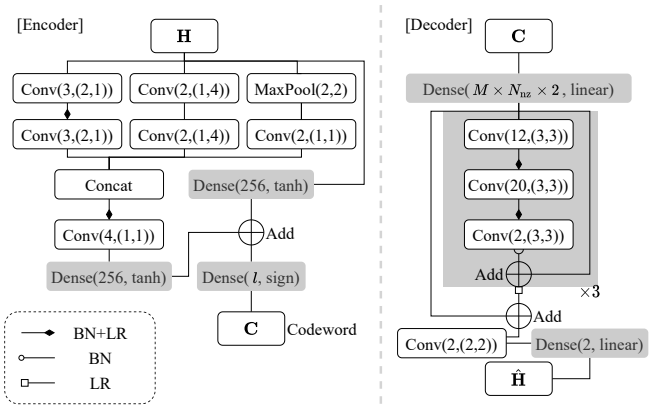

Fig. 4. The autoencoder architecture which consists of the encoder (left) and the decoder (right).

Quantization. To achieve quantization, the last dense layer of the encoder has a sign function as an activation function. Since the sign function, $sign(x) = 1$ for $x \geq 0$, else -1, is non-differentiable at 0, the derivative of the sign function at $x = 0$ is replaced by returning the gradient generated by backpropagation as it is. The encoder's output is $\mathbf{C} \in \{0,1\}^l$, which is an l-dimensional binary latent vector, and this binary information is transmitted to the BS over the feedback link. The encoder is denoted by f_{ENC} and we have $f_{\text{ENC}}(\mathbf{H}; \Theta_{\text{ENC}}) = \mathbf{C}$ where Θ_{ENC} is a set of all trainable encoder weights.

Decoding. The BS contains the decoder f_{DEC} and it expands the binary information $\mathbf{C}$ into an $M \times N_{\text{nz}} \times 2$-dimensional vector using the dense layer with a linear activation function. The expanded vector is reshaped into the size of the original channel matrix, and the CSI is restored by a residual decoding block consisting of three convolution layers. CsiNet's residual decoding block structure [6] was used for our decoder but we modified hyperparameters. Since the quantization process is included in the network architecture, we chose an encoder structure in which $\mathbf{H}$ is fed in multiple convolution layers so that the information of $\mathbf{H}$ can to be compressed with various resolutions.

C. Distortion Metrics and Loss Function

Mean squared error. The most widely used distortion metric is the mean squared error (MSE), which measures the distance between two matrices as follows

$$d_{\text{MSE}}(\mathbf{H}, \hat{\mathbf{H}}) = \mathbb{E}[||\mathbf{H} - \hat{\mathbf{H}}||^2] \approx \mathbb{E}[||\mathbf{H}_{\text{AD}} - \hat{\mathbf{H}}_{\text{AD}}||^2], \quad (2)$$

where $||\cdot||$ indicates the Frobenius norm which is an entry-wise matrix norm. The weights of the autoencoder can be updated so as to reduce the distance between the input and output of the autoencoder. Note that the DFT matrix is unitary so the MSE between a true channel matrix and an estimated channel matrix $\mathbf{H}_{\text{AD}} - \hat{\mathbf{H}}_{\text{AD}}$, which are on the spatial-delay domain, is equivalent to that on spatial-frequency domain.

Sine error. Distortion can be defined in various ways depending on the purpose of optimization. If the goal is to maximize the user's signal-to-noise ratio (SNR), we can directly maximize the SNR through supervised learning by converting the output of the neural network into information on the spatial-frequency domain. To directly maximize the SNR as opposed to reducing the MSE between the ground-truth and estimated channels, we propose a span-free distortion metric sine error (SE) d_{sine}, where $d_{\text{sine}}(\hat{\mathbf{H}}, \mathbf{H}) = d_{\text{sine}}(\hat{\mathbf{H}}, c\mathbf{H}), \forall c \in \mathbb{R}$ as follows

$$\mathbf{H}_{\text{AF}} \approx \mathbf{T}_{\text{DFT}} ZP(\mathbf{H}), \hat{\mathbf{H}}_{\text{AF}} \approx \mathbf{T}_{\text{DFT}} ZP(\hat{\mathbf{H}}), \quad (3)$$

$$\hat{\mathbf{H}}_{\text{AF}}^{\text{nml}} = \mathbf{T}_{\text{DFT}} ZP(\hat{\mathbf{H}}) \text{diag}\left(\frac{1}{||\hat{\mathbf{h}}_{\text{AF},1}||_2}, ..., \frac{1}{||\hat{\mathbf{h}}_{\text{AF},N_{\text{sc}}}||_2}\right), \quad (4)$$

$$d_{\text{sine}}(\mathbf{H}, \hat{\mathbf{H}}) = -\mathbb{E}[|\text{Tr}(\mathbf{H}_{\text{AF}}^H \hat{\mathbf{H}}_{\text{AF}}^{\text{nml}})|], \quad (5)$$

where $ZP(\cdot)$ is $M \times (N_{\text{sc}} - N_{\text{nz}})$-size zero-matrix padding to recover the spatial-delay domain information (Fig. 3). $\mathbf{T}_{\text{DFT}}$ is the discrete Fourier transform matrix and $\hat{\mathbf{H}}_{\text{AF}}^{\text{nml}}$ is a column-wise normalized precoding matrix. $\text{diag}(\cdot)$ is a diagonal matrix with the input sequence as its diagonal elements and $\hat{\mathbf{h}}_{\text{AF},n}$ is the n-th column of $\hat{\mathbf{H}}_{\text{AF}}$. This loss function allows the output of the neural network in the spatial-frequency domain to have normalized beamforming vectors for each subcarrier and enables updating in the direction in which the vector and the inner product of the actual channel are maximized. Using this metric, we can measure the distortion in the spatial-frequency domain even when the autoencoder receives a low-dimensional $M \times N_{\text{nz}}$ size matrix as an input.

D. Autoencoder Training

The autoencoder for the fixed-rate CSI feedback is trained to minimize distortion according to the desired metric and the problem is defined as follows.

$$\underset{\Theta_{\text{ENC}}, \Theta_{\text{DEC}}}{\text{minimize}} \quad \mathbb{E}[d(\mathbf{H}, \hat{\mathbf{H}})] \quad \text{(P1a)}$$

$$\text{subject to} \quad \hat{\mathbf{H}} = f_{\text{DEC}}(\mathbf{C}; \Theta_{\text{DEC}}), \mathbf{C} = f_{\text{ENC}}(\mathbf{H}; \Theta_{\text{ENC}}) \quad \text{(P1b)}$$

To solve this problem, we use a gradient-based update with respect to $\Theta_{\text{AE}} = \Theta_{\text{ENC}} \bigcup \Theta_{\text{DEC}}$, $\nabla_{\Theta_{\text{AE}}} \mathbb{E}[d(\mathbf{H}, \hat{\mathbf{H}})]$. In particular, we initialize the autoencoder by starting learning with the MSE loss function in (2) and finally solve the problem P1 by updating the weights again with the loss function in function (5). Detailed hyperparameter settings and learning techniques are covered in Section V.

III. Variable-Rate Code for CSI Feedback

In this section, we assume that we have N_{AE} CSI autoencoders each with a different number of feedback bits, $\{l_1, l_2, ..., l_{N_{\text{AE}}}\} = \mathcal{L}$. We design a variable-rate coding scheme by selecting an appropriate autoencoder according to the channel characteristics. More specifically, we can choose the optimal autoencoder for each CSI matrix by solving the following two problems. The first is the *distortion estimation problem*. We aim to estimate the distortion resulting from

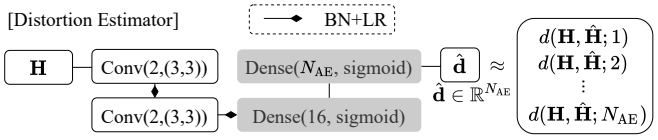

Fig. 5. The architecture of the distortion estimator $\hat{D}$. This shallow neural network can estimate the distortions $\hat{\mathbf{d}}$ from the N_{AE} autoencoders.

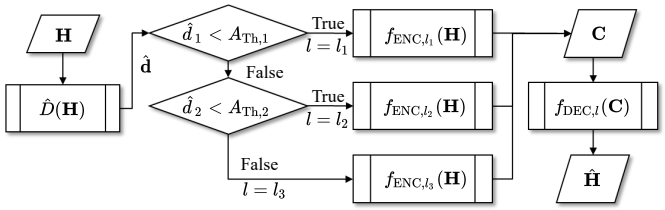

Fig. 6. The entire flowchart including CL where $N_{\text{AE}} = 3$. $\hat{D}$ estimates the distortion of a given CSI matrix $\mathbf{H}$ for each autoencoder, $\hat{d}_1, \hat{d}_2$, and $\hat{d}_3$. Based on the decision boundaries, $A_{\text{TH},1}$ and $A_{\text{TH},2}$, one of the encoders are selected and codeword $\mathbf{C}$ is generated.

each autoencoder without passing the CSI matrix through the autoencoders, in order to minimize the computational complexity of the UE. The second problem is a *codeword length allocation problem* with a codeword length selector that determines how many bits to devote to compress a given CSI matrix based on the estimated distortion value. The estimated distortion becomes a clear criterion for how easy or difficult to compress a given CSI instance. Based on this information, we propose algorithms to minimize the average number of feedback bits.

A. Design of Distortion Estimator

Distortion d resulting from the autoencoder with the codeword length l is denoted by $d(\mathbf{H}, \hat{\mathbf{H}}; l)$. We build a lightweight neural network, a distortion estimator $\hat{D}$ as depicted in Fig. 5, to estimate the true distortion from each autoencoder. The distortion estimator aims to approximate multiple distortion values $\mathbf{d} = (d(\mathbf{H}, \hat{\mathbf{H}}; l_1), ..., d(\mathbf{H}, \hat{\mathbf{H}}; l_{N_{\text{AE}}}))$ by taking only a given sample as input. The set of weights of the distortion estimator Θ_{D} can be updated in direction of minimizing the mean-squared error between the true and estimated distortion, $\mathbb{E}[||(\hat{D}(\mathbf{H}; \Theta_{\text{D}}) - \mathbf{d})||^2]$. The estimated distortion is denoted by $\hat{\mathbf{d}}$ and $\hat{d}_i$ is the i-th element of the vector $\hat{\mathbf{d}}$. This function allows us to rapidly estimate the distortion caused by compression without going through the autoencoder directly. The estimated distortion gives us a clear criterion for how many bits we need to compress a given CSI matrix.

B. Design of Feedback Codeword Length Controller

We can obtain a variable-rate coding scheme by designing a controller that allocates an appropriate codeword length l to a given CSI matrix. The purpose of this subsection is to design the codeword length controller based on the distortion estimator $\hat{D}$. We denote the function as CL (Codeword Length controller) and Θ_{CL} represents the set of all the parameters for CL. It takes a CSI matrix $\mathbf{H}$ as an input and allocates an appropriate length as $CL(\mathbf{H}; \Theta_{\text{CL}}) = l$ for the matrix and $l \in \{l_1, l_1, ..., l_{N_{\text{AE}}}\}$. Exploiting the distortion estimator $\hat{D}$, we design CL to have a simple and natural logic as follows: it measures the distortion of the CSI matrix and allocates more feedback bits if the distortion is large for a low number of bits. The detailed logic is specified in Algorithm 1. A CSI matrix observed by the UE, $\mathbf{H}$, is an input of the function. Through the distortion estimator $\hat{D}$, the distortion values $\hat{\mathbf{d}} = (\hat{d}_1, ..., \hat{d}_{N_{\text{AE}}})$, which are expected to be caused from the available N_{AE} autoencoders, are estimated. Then, one can check whether the estimated distortion $\hat{d}_1$, which is expected to occur when the autoencoder corresponding to the smallest bit is used, is smaller than the parameter $A_{\text{Th},1}$. If an autoencoder $(f_{\text{ENC},l_1}, f_{\text{DEC},l_1})$ with a small number of bits is expected to cause distortion smaller than the desired threshold $A_{\text{Th},1}$, CL allocates l_1 for the CSI matrix. If the estimated distortion is greater than the threshold, $\hat{d}_1 > A_{\text{Th},1}$, then the estimated distortion that will occur from the autoencoder corresponding to the next larger bit is checked. The same logic can be repeated with all the autoencoders that the UE has at its disposal. That is, one can continuously check whether the distortion which is below a certain level is observed even when compressed through autoencoder using as few bits as possible.

The thresholds $A_{\text{Th},1}, ..., A_{\text{Th},N_{\text{AE}}-1}$ can be optimized according to the desired goal. Our ultimate goal is to minimize the average number of feedback bits (P2a) while achieving a distortion no more than D_{TH} using the given autoencoders (P2b) and following Algorithm 1 (P2c).

Algorithm 1 Function $CL(\cdot)$: CodeLength Selector

1: Input: $\mathbf{H}$
2: Parameters: $\Theta_{\text{CL}} = (A_{\text{Th},1}, A_{\text{Th},2}, ..., A_{\text{Th},N_{\text{AE}-1}})$
3: Estimate distortion values $\hat{D}(\mathbf{H}; \Theta_{\text{D}}) = \hat{\mathbf{d}} \in \mathbb{R}^{N_{\text{AE}}}$
4: **for** $i = 1$ to $N_{\text{AE}} - 1$ **do**
5: **if** $\hat{d}_i, < A_{\text{Th},i}$ **then**
6: Return l_i (Use l_i-bit AE), (Algorithm Ends)
7: **end if**
8: **end for**
9: Return $l_{N_{\text{AE}}}$ (Use $l_{N_{\text{AE}}}$-bit AE), (Algorithm Ends)

$$\underset{\Theta_{\text{CL}}}{\text{minimize}} \quad \mathbb{E}[\mathbf{L}] \tag{P2a}$$

$$\text{subject to} \quad \mathbb{E}[d(\mathbf{H}, f_{\text{DEC},l=\mathbf{L}}(f_{\text{ENC},l=\mathbf{L}}\mathbf{H}))] < D_{\text{TH}}, \tag{P2b}$$

$$\mathbf{L} = CL(\mathbf{H}; \Theta_{\text{CL}}) \tag{P2c}$$

CL can radpily determine the codeword length using only the parameters Θ_{CL}. Since the cardinality of the set Θ_{CL} is equal to $N_{\text{AE}} - 1$, this is a low-dimensional optimization problem. *Covariance matrix adaptation evolution strategy* (CMA-ES) [14], which is naturally suited to this case, is used to optimize the $N_{\text{AE}} - 1$ variables for problem P2. CMA-ES is a numerical optimization method based on an evolutionary strategy and a

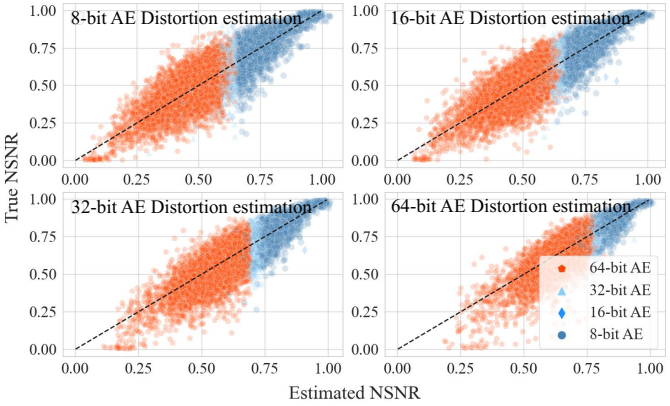

Fig. 7. The actual negative distortion (vertical axis, True NSNR) caused by the autoencoder and the estimated negative distortion by $\hat{D}$. The more concentrated the data is on the dash line of each subplot, the higher the accuracy of the distortion estimator. For AE that devotes higher bits, both the predicted distortion and the actual distortion show small values. CL allocates a larger number of bits to samples where large distortion is expected.

derivative-free optimization method. It is well known that the technique can effectively search the domain of the variables to be optimized with the efficient recombination and mutation techniques. In this way, we obtain the thresholds that can lead the UE to use the smallest average number of bits while maintaining the average distortion below the desired level.

Fig. 6 shows the flowchart of the proposed algorithm that consists of three steps, the distortion estimation, codeword length selection, and encoding-decoding process.

IV. Numerical Results

A. Simulation Settings

We consider a system with a center frequency of 2.53 GHz. The number of subcarriers used by each UE is $N_{\text{sc}} = 257$, and the subcarrier spacing is 15000 Hz. The BS has $M = 8$ transmit antennas and the UE has a single antenna. The location of the UE is uniformly distributed on the disc with a radius of 500m centered at the BS. The speed and the direction of the UE are uniformly distributed over [0,2] km/h and [0,2π], respectively. To generate the dataset, we use the 3D channel model of the 3GPP technical report 36.873 [15] through Quasi Deterministic Radio Channel Generator (QuaDRiGa) [16]. The parameter setting for the channel realization follows the *Terrestrial Urban Macrocell parameters extracted from measurements in Berlin* [17].

We construct $N_{\text{AE}} = 4$ autoencoders with the feedback codewords of lengths $l \in \{8, 16, 32, 64\}$ (bits). We let the autoencoder takes a 8×32 size complex CSI matrix as an input, i.e., $N_{\text{nz}} = 32$. To update the weights of the each autoencoder as depicted in Fig. 4, we adopt the Cosine Annealing Learning Rate (CALR) [18] technique together with the Adam optimizer [19] where the learning rate varies over [1e-3,1e-5]. The number of data samples for training, validation and testing is 100,000, 20,000, and 10,000, respectively. The batch size is set to 100, and the training is done over 600 epochs.

TABLE I
Performance comparison for the fixed-rate AEs with MSE, SE, and VRC

Average of Codeword Length / Average NSNR / Standard Deviation of NSNR			
MSE	SE	VRC-$\hat{\mathbf{d}}$	VRC-$\mathbf{d}$
8 / 0.50 / 0.20	8 / 0.56 / 0.18	8 / 0.56 / 0.18	8 / 0.56 / 0.18
16 / 0.55 / 0.18	16 / 0.57 / 0.17	12.27 / 0.58 / 0.17	10.44 / 0.58 / 0.16
32 / 0.61 / 0.17	32 / 0.63 / 0.16	30.63 / 0.63 / 0.15	21.44 / 0.63 / 0.13
64 / 0.69 / 0.15	64 / 0.70 / 0.14	49.61 / 0.68 / 0.13	51.97 / 0.69 / 0.14

B. Performance Analysis

Distortion Estimation and VRC. The four subplots in Fig. 7 show the distortion estimation results of our four fixed-rate AEs. The vertical axis represents the actual Normalized SNR (NSNR)[2] and the horizontal axis represents the NSNR predicted by the distortion estimator. Note that NSNR can be interpreted as a negative distortion. For autoencoders with a more the number of feedback bits, we can see that the distortion values of the samples cluster in the upper right, which means better compression performance. Markers and colors indicate how many bits the CL has allocated for each of the 10,000 test CSI instances when $D_{\text{TH}} = 0.63$ with $\hat{\mathbf{d}}$ for P2. For example, the CSI samples marked with a red pentagram indicate that they were assigned to a 64-bit autoencoder. We observe that the higher the distortion predicted (the lower the NSNR predicted), the more the bits are allocated.

Table 1 shows the performance comparison of the autoencoders trained using MSE, sine error (SE), and VRC.

MSE vs. SE loss. The autoencoders trained with SE perform better than the autoencoders trained with MSE in terms of NSNR. The performance gap between the autoencoders using SE and MSE narrows as the length of the codeword increases. VRC is constructed using the four SE-based autoencoders that show better performance in NSNR.

Fixed-rate vs. Variable-rate. Compared to the fixed-rate AEs trained with SE, VRC-$\hat{\mathbf{d}}$ which selects the rate adaptively based on the estimated distortion $\hat{\mathbf{d}}$ requires relatively fewer bits to achieve similar reliability (NSNR). VRC-$\mathbf{d}$ indicates the result from solving P2 with the true distortion $\mathbf{d}$ instead of the estimated distortion $\hat{\mathbf{d}}$ (line 3, Algorithm 1). We can see that it require fewer bits than VRC-$\hat{\mathbf{d}}$. That is, when the UE knows the actual distortion value, i.e. by performing both the encoding and decoding, the number of bits required can be reduced more efficiently.

When the VRC is used, the desired performance can be achieved by variously adjusting D_{TH} and at the same time, the minimum number of bits can be achieved by varying the rates. Among the distortion performance achievable using the VRC, Table I shows the performance of the VRCs most similar to the ones obtained from fixed-rate autoencoders in terms of the distortion. Note that if the distortion decreases monotonically in the length of the latent vector, no additional gain can be obtained through distortion estimation and code

[2]We normalized the channel of each subcarrier and also normalized the precoding vector to only observe the change in SNR according to csi feedback. Transmission power and noise variance were assumed to be 1. We refer to the SNR measured in this environment as NSNR.

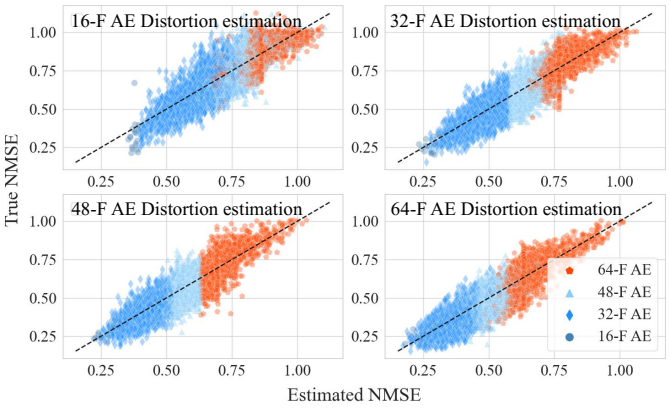

Fig. 8. The difference in compression distortion among samples is also observed in the COST2100 dataset (outdoor setting) using CsiNet [6]. $\hat{D}$ can estimate distortion of CsiNet and our proposed scheme allocates more floating point numbers to the CSI instances that are expected to generate large distortion.

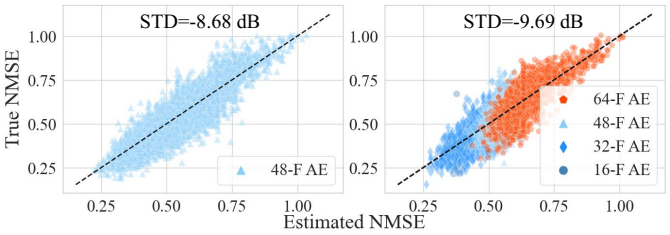

Fig. 9. True and estimated distortion achieved by 48-F AE (left) and VRC-$\hat{\mathbf{d}}$ (right). 48-F CsiNet with CA uses 48 bits to achieve -2.58dB and VRC-$\hat{\mathbf{d}}$ uses 46.69 bits to achieve -2.61dB. Moreover, The standard deviation (imbalance) of the distortion was mitigated from -8.68dB to -9.69dB.

length allocation. However, since some CSI instances show similar distortion at different compression rates, VRCs exploit them to reduce the required number of bits.

From Table I, we observe that VRC can reduce distortion imbalance among the CSI instances while achieving better or similar overall performance.

Performance comparisons on COST2100 dataset with CsiNet. We reproduce CsiNet with 4 different compression ratios, for example, 32-F indicates that the AE compresses CSI as 32 floating point numbers. CALR method is adopted (CsiNet with CA) and all other hyperparameter settings and performance metric follow the original paper [6]. For this experiment, normalized MSE is used for the autoencoder training.

The distortion difference among the CSI instances is also observed for CsiNets trained for the COST2100 outdoor dataset [6]. Fig. 8 shows that the distortion resulting from CsiNet with the various number of feedback bits is predictable. As a more the number of feedback bits is allowed, we observe that the predicted distortion and actual distortion samples are clustered in the lower left (low distortion). The markers distinguished by color and shape show what autoencoder the proposed method uses to compress each sample when $D_{\text{TH}} = -2.59$ dB with $\hat{\mathbf{d}}$. The higher distortion is predicted,

TABLE II
Performance comparison for CsiNet, CsiNet with CA, and VRC

Average of Codeword Length / Average NMSE (dB) / Standard Deviation (dB)			
CsiNet[6]	CsiNet with CA	VRC-$\hat{\mathbf{d}}$	VRC-$\mathbf{d}$
-	16 / -1.26 / -8.39	16 / -1.26 / -8.39	16 / -1.26 / -8.39
32 / -1.93	32 / -2.14 / -8.34	31.02 / -2.10 / -8.51	31.98 / -2.20 / -9.23
-	48 / -2.58 / -8.68	46.69 / -2.61 / -9.69	41.80 / -2.60 / -10.05
64 / -2.81	64 / -3.10 / -8.61	61.10 / -3.00 / -9.02	62.67 / -3.09 / -8.69

the proposed method allocates a larger number of floating numbers for compression. Fig. 9 shows that VRC-$\hat{\mathbf{d}}$ can achieve better performance while mitigating the performance imbalance among the CSI instances. From Table II, it is observed that VRC-$\hat{\mathbf{d}}$ and VRC-$\mathbf{d}$ performs better than the fixed-rate approaches in terms of feedback efficiency and distortion balance.

V. Conclusion and Future Directions

This paper brings attention to the possible benefit of using variable-rate coding for CSI feedback by observing the significant difference in distortion among CSI instances under fixed-rate data-driven DL-based CSI feedback schemes. We devise a way to exploit these differences by designing distortion estimator that can estimate distortion due to the lossy compression using various fixed-rate feedback autoencoders. Based on the estimated distortion values, the proposed method selects the number of feedback bits by choosing an appropriate autoencoder for each CSI instance. The experimental results show that the proposed scheme reduces the number of bits required to achieve the desired average distortion while reducing the distortion imbalance among the CSI instances. The variable-rate feedback approach is expected to reduce the feedback overhead and imbalance, especially in the coordinated multipoint environment where users may be required to send CSI feedback associated with multiple BSs.

References

[1] "Evolved Universal Terrestrial Radio Access (E-UTRA) physical layer procedures," 3rd Generation Partnership Project (3GPP), Technical Specification Group Radio Access Network 36.213, 01 2017, version 14.1.

[2] T. Strohmer and R. W. Heath Jr, "Grassmannian frames with applications to coding and communication," *Applied and computational harmonic analysis*, vol. 14, no. 3, pp. 257–275, 2003.

[3] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE transactions on information theory*, vol. 49, no. 10, pp. 2735–2747, 2003.

[4] W. Santipach and M. L. Honig, "Asymptotic performance of mimo wireless channels with limited feedback," in *IEEE Military Communications Conference, 2003. MILCOM 2003.*, vol. 1. IEEE, 2003, pp. 141–146.

[5] A.-Y. et al., "On the performance of random vector quantization limited feedback beamforming in a miso system," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 458–462, 2007.

[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] T. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based csi feedback approach for time-varying massive mimo channels," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 416–419, 2018.

[8] C. Lu et al., "Mimo channel information feedback using deep recurrent network," *IEEE Comm. Letters*, vol. 23, no. 1, pp. 188–191, 2018.

[9] X. Li and H. Wu, "Spatio-temporal representation with deep neural recurrent network in mimo csi feedback," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 653–657, 2020.
[10] Z. Lu, J. Wang, and J. Song, "Multi-resolution csi feedback with deep learning in massive mimo system," in *IEEE International Conference on Communications*. IEEE, 2020, pp. 1–6.
[11] 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.
[12] 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, 2020.
[13] B. Wang, F. Gao, S. Jin, H. Lin, and G. Y. Li, "Spatial-and frequency-wideband effects in millimeter-wave massive mimo systems," *IEEE Transactions on Signal Processing*, vol. 66, no. 13, pp. 3393–3406, 2018.
[14] N. Hansen et al., "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.
[15] "Study on 3D channel model for LTE," 3rd Generation Partnership Project (3GPP), Technical Specification Group Radio Access Network 36.873, 06 2017, version 12.5.0.
[16] S. Jaeckel et al., "Quadriga: A 3-d multi-cell channel model with time evolution for enabling virtual field trials," *IEEE transactions on antennas and propagation*, vol. 62, no. 6, pp. 3242–3256, 2014.
[17] ——, "Quadriga - quasi deterministic radio channel generator, user manual and documentation," *Fraun-hofer Heinrich Hertz Institute*, vol. 2.6.1, 2021.
[18] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.
[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.