Deep Channel Estimation and Feedback in Massive MIMO

Sagar Shrestha [1]

Abstract

In this report, a framework for downlink channel estimation and feedback for massive MIMO system in wireless communication is presented, based on readings on the recent developments in the field. The presented framework considers deploying a neural network trained by minimising rate-distortion loss, and directly encoding the pilot observation at the *user equipment* (UE) for limited feedback. Simulations are carried out that demonstrates the advantages of the proposed scheme.

1. Introduction

Massive *multi input multi output* (MIMO) is considered one of the key technology enablers for future wireless communication because of its high spectral efficiency, beamforming gains, and energy efficiency. One of the key components of massive MIMO is availability of downlink *channel state information* (CSI) at *base station* (BS) and *user equipment* UE. UE uses downlink CSI to decode received messages and BS uses downlink CSI to make efficient resource allocation among multiple users, beamforming, and precoding of downlink messages. Downlink CSI is estimated at the UE using pilot training. In *time division duplexing* (TDD), BS can also estimate downlink CSI by utilizing uplink CSI because of channel reciprocity. However in *frequency division duplexing* (FDD), CSI for uplink and downlink are very different and thus needs to be estimated at the UE and transmitted to the BS.

CSI estimation and feedback aims at accurate downlink CSI estimation at UE and recovery of the downlink CSI at BS via limited feedback from UE. The challenge of CSI estimation is that it has to estimate high dimensional channel matrix from limited pilot signals. CSI identification is impossible without assuming any prior on CSI because there is an infinite number of CSI perfectly consistent with the observations. The challenge of CSI feedback is that the need for transmitting back high dimensional CSI from UE to BS every coherence time interval imposes huge burden on the communication channel and can significantly degrade transmission rate. As such, CSI data needs to be heavily compressed before transmission while ensuring accurate reconstruction at the BS.

A plethora of techniques have been proposed to address CSI estimation and feedback independently or jointly. Mainly, sparsity of CSI in certain domains have been exploited for estimation from limited pilot observation (Nguyen & Ghrayeb, 2013) and reconstruction from limited feedback (Kuo et al., 2012). Estimation of key parameters of the channel model and feedback of the same is also considered in (Alevizos et al., 2018).

More recently, data-driven approaches have shown promising results in both estimation and feedback problems. In (Balevi et al., 2020b; Doshi et al., 2020), a *generative adversarial network* (GAN) prior is used to identify the CSI from limited pilot observations. (Balevi et al., 2020a) considers using unlearned neural network as a prior for CSI and (Dong et al., 2020) uses conditional GAN prior.

Similarly, for limited feedback problem, (Wen et al., 2018) uses a learned encoder at the UE to compress CSI into low dimensional latent representation and a decoder at the BS to recover CSI from the feedback of its encoded representation. In (Yang et al., 2019; 2020), the authors also consider quantization and encoding of latent represenation before feedback by directly training their network to minimize rate-distortion loss function that tries to minimize the number of feedback bits and the reconstruction error.

Taking inspiration from recent developments in data-driven image compression (Ballé et al., 2016; 2018) and similar to (Yang et al., 2020) for CSI feedback, the presented framework considers the limited feedback problem as the optimization of two competing objectives, the number of feedback bits and the quality of reconstruction. Unlike (Yang et al., 2020), proposed method does not cosider the availability of ground-truth CSI at the BS. Instead, encoding is applied directly to the received pilot observations which reduces error propagation from the CSI estimation stage.

[1] Department of Electrical Engineering and Computer Science, Oregon State University, Oregon, USA.. Correspondence to: Sagar Shrestha <shressag@oregonstate.edu>.

Project Report, Matrix Analysis, Winter 2021.

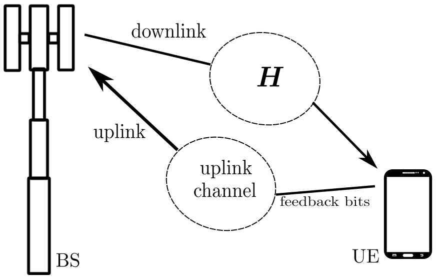


Figure 1. Illustration of uplink feedback channel and downlink channel between BS (cell tower) and UE (mobile phone) in wireless communication.

2. Background

A typical single user MIMO communication scenario in FDD mode is depicted in figure 1. The downlink channel $\boldsymbol{H}$ is estimated at the UE and feedback bits are sent to the BS via uplink channel. We consider that the BS is equipped with N_{t} transmit antennas and operate using *orthogonal frequency division multiplexing* (OFDM) with N_{c} subcarriers. For simplicity, the UE is assumed to have a single receive antenna. At the BS, data-bearing symbol $x \in \mathbb{C}$ is pre-coded by $\boldsymbol{v} \in \mathbb{C}^{N_{\mathrm{t}} \times 1}$ and transmitted to the UE. The received signal $\boldsymbol{u} \in \mathbb{C}^{N_{\mathrm{c}} \times 1}$ at the UE is given by

$$\boldsymbol{u} = \boldsymbol{H}\boldsymbol{v}x + \boldsymbol{w}, \tag{1}$$

where, $\boldsymbol{H} \in \mathbb{C}^{N_{\mathrm{c}} \times N_{\mathrm{t}}}$ is the downlink CSI matrix, $\boldsymbol{w} \in \mathbb{C}^{N_{\mathrm{c}} \times 1}$ is the additive noise. The BS requires an estimate of $\boldsymbol{H}$ to design precoding vector $\boldsymbol{v}$. Acquiring estimate of $\boldsymbol{H}$ at the BS also enables several advanced wireless communication tasks such as beamforming, resource allocation, etc. The UE requires $\boldsymbol{H}$ for decoding downlink messages. In the next subsections, we discuss the two challenges associated with this communication process.

2.1. Channel Estimation

In FDD system, UE estimates the CSI matrix from pilot-based training. A set of signals known as 'pilot' sequence is decided upon by BS and UE before communication. During the training stage, BS sends this sequence to the UE. The received signal is given by

$$\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{P} + \boldsymbol{W}, \tag{2}$$

where $\boldsymbol{P} \in \mathbb{C}^{N_{\mathrm{t}} \times N_{\mathrm{p}}}$ is the pilot sequence, and $\boldsymbol{W} \in \mathbb{C}^{N_{\mathrm{t}} \times N_{\mathrm{p}}}$ is the additive noise. The task of channel estimation is to estimate $\boldsymbol{H}$ given $\boldsymbol{Y}$ and $\boldsymbol{P}$ in (2). We can rewrite (2) is vector notations as

$$\mathrm{vec}(\boldsymbol{Y}) = \mathrm{vec}(\boldsymbol{H}\boldsymbol{P}) + \mathrm{vec}(\boldsymbol{W}), \tag{3}$$

where vec$(.)$ is a vectorization operator given by

$$\mathrm{vec}(\boldsymbol{Y})((j-1)N_{\mathrm{p}} + i) = \boldsymbol{Y}(i, j).$$

It is known that $\mathrm{vec}(\boldsymbol{ABC}) = (\boldsymbol{C}^{\mathsf{T}} \otimes \boldsymbol{A})\mathrm{vec}(\boldsymbol{B})$. Rewriting $\boldsymbol{HP}$ in (2) as $\boldsymbol{I}_{N_{\mathrm{c}}}\boldsymbol{HP}$, this gives us

$$\mathrm{vec}(\boldsymbol{Y}) = (\boldsymbol{P}^{\mathsf{T}} \otimes \boldsymbol{I}_{N_{\mathrm{c}}})\mathrm{vec}(\boldsymbol{H}), \tag{4}$$

where $\boldsymbol{I}_{N_{\mathrm{c}}}\mathbb{R}^{N_{\mathrm{c}} \times N_{\mathrm{c}}}$ is an identity matrix. We can see from (4) that if $N_{\mathrm{p}} \leq N_{\mathrm{t}}$, the system is underdetermined, so there is an infinite number of $\boldsymbol{H}$ perfectly consistent with (4). This requires that some prior be imposed upon $\boldsymbol{H}$ so that the solution to (4) is unique. Sparsity in certain domain (angular-delay) is a commonly used prior for channel matrices. Corresponding sparsity based recovery methods such as Lasso regularization has been widely adopted. Recently, deeply learned priors have shown to outperform classical methods based on sparsity.

2.2. Limited Feedback

In FDD system, the CSI estimate needs to be fedback to the BS. The size of feedback message $N_{\mathrm{c}}N_{\mathrm{t}}$ grows with the number of transmitters and subcarriers, and is prohibitive in massive MIMO regime. This imposes restriction on the size of feedback message that can be used for CSI information. Essentially, we want the size of feedback message as small as possible and the reconstructed CSI at the base station as close as possible to the ground truth.

In this report, we consider the problem as a rate-distortion minimization. Rate-distortion is an information theoretic concept that deals with the problem of minmimising the rate R, number of bits used to represent or transmit a sample, and distortion D, the error in reconstruction of the sample. It is the theoretical foundation for lossy image compression. Since the feedback message has to be converted into bit representation before transmission, rate corresponds to the number of feedback bits per CSI estimate in our case. Recently deep neural networks learnt by directly minimizing rate-distortion loss has been widely investigated due to its superior performance compared to traditional compression techniques and this line of work has been adopted for limited feedback of CSI in (Yang et al., 2020).

In this report, we demonstrate the advantage of considering rate-distortion minimization as the optimization criteria and directly encoding pilot observation for limited feedback.

3. Proposed Method

CSI estimation and feedback is generally carried out in tandem. First, CSI matrix $\boldsymbol{H}$ is estimated from pilot observation $\boldsymbol{Y}$. The estimate $\widehat{\boldsymbol{H}}$ is then compressed and fedback to the BS. It can be seen that the estimation error is propagated to the feedback stage, i.e. the BS tries to recover $\widehat{\boldsymbol{H}}$ and not $\boldsymbol{H}$. A better approach would be to directly generate compressed feedback message from $\boldsymbol{Y}$. The difference is illustrated in figure 2. In this report, we adopt parallel

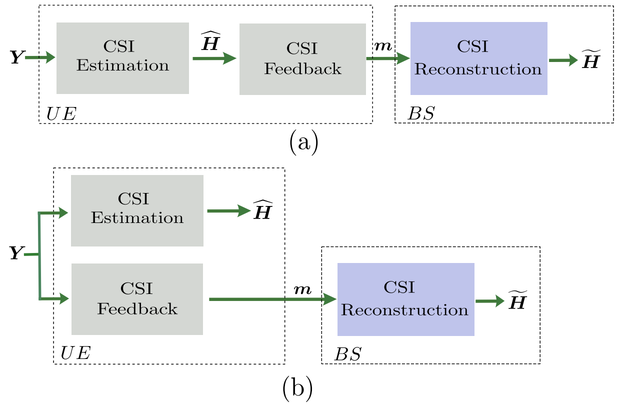


Figure 2. (a) CSI estimation and feedback in tandem. Here mutual information between $\widehat{\boldsymbol{H}}$ and $\boldsymbol{Y}$ is always less or equal to that between $\boldsymbol{Y}$ and $\widehat{\boldsymbol{H}}$. (b) CSI estimation and feedback in paralle. Here, feedback message $\boldsymbol{m}$ is directly designed from pilot observation $\boldsymbol{Y}$ instead of estimated CSI $\widehat{\boldsymbol{H}}$

estimation and feedback scheme in figure 2(b).

3.1. CSI Estimation

In CSI estimation, we are interested in recovering underlying CSI matrix from pilot observation as accurately as possible. For this purpose we can train a deep neural network to learn the mapping function from pilot observation to CSI. We generate L realizations of CSI matrices $\{\boldsymbol{H}_\ell\}_{\ell=1}^{L}$ using wireless channel simulators (see section 4 for details). We aim to find a mapping function $\boldsymbol{f}_\theta(.) : \mathbb{C}^{N_c \times N_p} \to \mathbb{C}^{N_c \times N_t}$ with the following training process:

$$\min_{\boldsymbol{\theta}} \sum_{\ell=1}^{L} \frac{1}{N_c N_p} \left\| \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{H}_\ell \boldsymbol{P}) - \boldsymbol{H}_\ell \right\|_F^2 , \tag{5}$$

where $\boldsymbol{P} \in \mathbb{C}^{N_c \times N_p}$ is a matrix consisting of fixed N_p pilot signals. Training is carried out offline and estimated mapping function $\boldsymbol{f}_{\widehat{\boldsymbol{\theta}}}$ is deployed in the UE. At test time, pilot observation $\boldsymbol{Y}_{\text{test}}$ is fed into the network to obtain an estimate of the channel, i.e.

$$\widehat{\boldsymbol{H}}_{\text{test}} = \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{Y}_{\text{test}}).$$

Since $\boldsymbol{H}$ and $\boldsymbol{Y}$ are complex matrices, and our neural network based mapping function can only deal with real numbers, we divide the real part, $\text{real}(\boldsymbol{Y})$, and imaginary part $\text{imag}(\boldsymbol{Y})$ into two input channels in our training and testing examples. Two output channels from the network similarly corresponds to $\text{real}(\widehat{\boldsymbol{H}})$ and $\text{imag}(\widehat{\boldsymbol{H}})$. Therefore the actual input to our network in practice $\in \mathbb{R}^{N_c \times N_p \times 2}$ and output $\in \mathbb{R}^{N_c \times N_t \times 2}$.

Proposed deep neural network based mapping function follows an autoencoder architecture. The network is termed

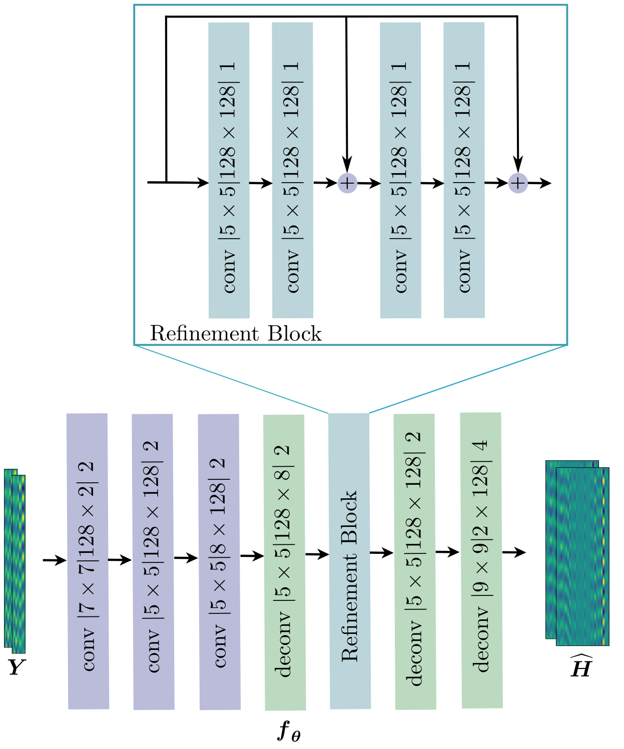

conv $|5\times5|128\times128|$ 1
conv $|5\times5|128\times128|$ 1
conv $|5\times5|128\times128|$ 1
conv $|5\times5|128\times128|$ 1
Refinement Block
$\boldsymbol{Y}$
conv $|7\times7|128\times2|$ 2
conv $|5\times5|128\times128|$ 2
conv $|5\times5|8\times128|$ 2
deconv $|5\times5|128\times8|$ 2
Refinement Block
deconv $|5\times5|128\times128|$ 2
deconv $|9\times9|2\times128|$ 4
$\widehat{\boldsymbol{H}}$
$\boldsymbol{f}_{\boldsymbol{\theta}}$

Figure 3. Parameterization of `CENet`.Each layer is followed by batch normalization and PReLU activation except for the last layer.

as *Channel Estimation Network* (`CENet`) and depicted in figure 3. It is a fully convolutional network with 3 layers in the encoder and 7 layers in the decoder. Downsampling and upsampling are achieved by strided convolution. The decoder consists of 3 upsampling layers and 4 refinement layers termed as *refinement block*. Output size of *refinement block* is the same as the input.

In figure 3, $\text{conv}|7 \times 7|128 \times 2|2$ represents affine convolution with filter size fo 7×7; (output $\times$ input) channels of 128×2; the trailing integer refers to the stride length in convolution operation; and 'deconv' refers to transposed convolution.

In figure 3, the two channel input to the `CENet` corresponds to the real and imaginary part of the pilot observation $\boldsymbol{Y}$. The two channel output of `CENet` corresponds to the real and imaginary part $\widehat{\boldsymbol{H}}$.

3.2. CSI Feedback

The BS has a more challenging task compared to UE in recovering downlink CSI. To reduce feedback overhaed, the UE can only send a limited number of bits to reconstruct CSI at each coherence time. Therefore any representation of CSI has to be encoded by a limited number of bits. If the representation is a vector of complex numbers, it will undergo quantization before encoding.

The feedback framework presented in this report is similar to (Yang et al., 2020), except that we directly encode $\boldsymbol{Y}$ insdead of $\boldsymbol{H}$ for feedback. Essentially, we want to train a deep autoencoder as `CENet`, but take into consideration that the compressed representation will undergo quantization and binary encoding before passing through the decoder. Additionally, we want to train the autoencoder in such a way that the quantized representation has a small entropy so that the expected length of feedback code is small.

Let $\boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}(.) : \mathbb{C}^{N_{\mathrm{c}} \times N_{\mathrm{p}}} \to \mathbb{R}^k$ denote the encoder that takes pilot observation $\boldsymbol{Y}$ as input and produces a representation $\boldsymbol{z} = \boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}(\boldsymbol{Y})$. We then apply uniform quantization to $\boldsymbol{z}$. The quantization operator $\mathsf{QC}(.) : \mathbb{R}^D \to \mathbb{N}^k$ outputs the nearest integer corresponding to the real input. Let $\widehat{\boldsymbol{z}} = \mathsf{QC}(\boldsymbol{z})$. Let $\mathcal{C}$ be the set of uniquely decodable binary codewords used to encode the feedback message. $\widehat{\boldsymbol{z}}$ is then encoded into codeword $\boldsymbol{m} \in \mathcal{C}$ by an entropy coder represented by $\mathsf{EC}(.) : \mathbb{N}^k \to \mathcal{C}$, i.e. $\boldsymbol{m} = \mathsf{EC}(\widehat{\boldsymbol{z}})$. Then this codeword is transmitted to the BS. We assume that $\boldsymbol{m}$ is received at the BS without error.

At the BS, entropy decoder, represented by $\mathsf{ED}() : \mathcal{C} \to \mathbb{N}^k$, decodes feedback bits $\boldsymbol{m}$ back into $\widehat{\boldsymbol{z}}$. Note that ED and EC represent lossless encoding and decoding. Therefore, $\widehat{\boldsymbol{z}}$ can be exactly reconstructed at the BS. $\widehat{\boldsymbol{z}}$, then passes through a neural network based decoder $\boldsymbol{g}_{\boldsymbol{\theta}_{\mathrm{d}}}(.) : \mathbb{R}^k \to \mathbb{C}^{N_{\mathrm{c}} \times N_{\mathrm{t}}}$ to produce the estimate of downlink CSI, $\widetilde{\boldsymbol{H}} = \boldsymbol{g}_{\boldsymbol{\theta}_{\mathrm{d}}}(\widehat{\boldsymbol{z}})$.

The entire process is illustrated in figure 4. Note that EC and ED needs an estimate of the *probability mass function* (pmf) of $\widehat{\boldsymbol{z}}$. We denote the pmf estimator by $\boldsymbol{p}_{\boldsymbol{\beta}}(.) : \mathbb{N}^k \to \mathbb{R}$ that evaluates the probability of $\widehat{\boldsymbol{z}}$.

Let $R_{\natural}$ denote the expected number of feedback bits per channel matrix. Then we have,

$$R_{\natural} = \mathbb{E}_{\boldsymbol{Y}}[\mathrm{len}(\boldsymbol{m})],$$

where $\mathbb{E}_{\boldsymbol{Y}}[.]$ denotes the expectation over the underlying distribution of pilot observation. We know that the expected length of a codeword using optimal coding schemes is upper and lower bounded by the entropy of the message to be encoded. Therefore we can approximate $R_{\natural}$ by R, such that,

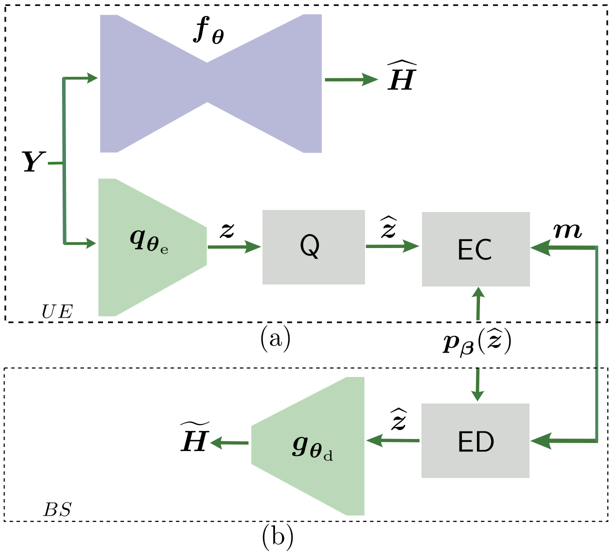
$$R = \mathbb{E}_{\boldsymbol{Y}}[-\log_2 \mathsf{P}_{\widehat{z}}(\widehat{\boldsymbol{z}})],$$

where $\mathsf{P}_{\widehat{\boldsymbol{z}}}(\widehat{\boldsymbol{z}})$ is the *probability mass function* (pmf) of $\widehat{\boldsymbol{z}}$.

Let D denote the expected *mean squared error* (MSE) of reconstruction, i.e.

$$D = \mathbb{E}_{\boldsymbol{Y}}\left[\frac{1}{N_{\mathrm{p}} N_{\mathrm{c}}} \|\widetilde{\boldsymbol{H}} - \boldsymbol{H}\|_{\mathrm{F}}^2\right],$$

where, $\widetilde{\boldsymbol{H}} = \boldsymbol{g}_{\boldsymbol{\theta}_{\mathrm{d}}}(\mathsf{Q}(\boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}(\boldsymbol{Y})))$ and $\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{P}$. We wish to find parameters $\boldsymbol{\theta}_{\mathrm{d}}$ and $\boldsymbol{\theta}_{\mathrm{e}}$ as follows:


$\boldsymbol{f}_{\boldsymbol{\theta}}$
$\widehat{\boldsymbol{H}}$
$\boldsymbol{Y}$
$\boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}$
$\boldsymbol{z}$
Q
$\widehat{\boldsymbol{z}}$
EC
$\boldsymbol{m}$
UE
$\boldsymbol{p}_{\boldsymbol{\beta}}(\widehat{\boldsymbol{z}})$
$\widetilde{\boldsymbol{H}}$
$\boldsymbol{g}_{\boldsymbol{\theta}_{\mathrm{d}}}$
$\widehat{\boldsymbol{z}}$
ED
BS

(a)

(b)

Figure 4. Illustration of limited feedback process and CSI estimation. CSI estimation is carried out by a single forward pass on `CENet` represented by $\boldsymbol{f}_{\boldsymbol{\theta}}$. For feedback the UE compresses pilot observation $\boldsymbol{Y}$ through a learned encoder $\boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}$ applies uniform quantization Q and encodes into bits using estimated pmf $\boldsymbol{p}_{\boldsymbol{\beta}}$. At the BS, it is decoded using the same pmf and the CSI is reconstructed using a learned decoder $\boldsymbol{g}_{\boldsymbol{\theta}_{\mathrm{d}}}$

$$\min_{\boldsymbol{\theta}_{\mathrm{d}}, \boldsymbol{\theta}_{\mathrm{e}}} R + \lambda D,$$

where $\lambda > 0$ controls the rate distortion tradeoff.

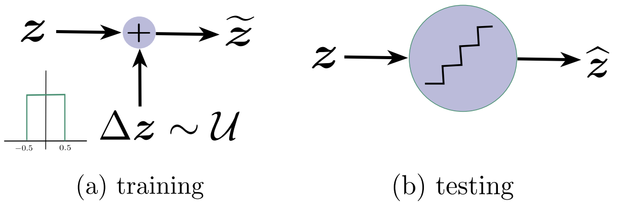
Since computing the expectation exactly is not feasible, we estimate it empirically with I training pairs $\{\boldsymbol{Y}_i, \boldsymbol{H}_i\}_{i=1}^n$ as follows:

$$\min_{\boldsymbol{\theta}_{\mathrm{d}}, \boldsymbol{\theta}_{\mathrm{e}}} \sum_{i=1}^{I} \underbrace{-\log_2\left(\mathsf{P}_{\widehat{\boldsymbol{z}}}(\mathsf{Q}(\boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}(\boldsymbol{Y}_i)))\right)}_{\widehat{R}} + \lambda \underbrace{\|\boldsymbol{g}_{\boldsymbol{\theta}_{\mathrm{d}}}(\mathsf{Q}(\boldsymbol{q}_{\boldsymbol{\theta}_{\mathrm{e}}}(\boldsymbol{Y}_i))) - \boldsymbol{H}_i\|_{\mathrm{F}}^2}_{\widehat{D}}. \qquad (6)$$

Minimizing loss function in (6) has two problems. First, there is a quantization operation which is non-differentiable, so gradient based minimization cannot work with it. Second, the rate term requires knowing the pmf of $\widehat{\boldsymbol{z}}$. We briefly discuss methods for dealing with these problems in the following subsections.

3.2.1. Quantization

Quantization operation has zero derivative everywhere except at the transitions where the derivative does not exist. We follow the method in (Ballé et al., 2016) and replace the quantization operation with uniform noise $\Delta \boldsymbol{z} \sim$



(a) training (b) testing

Figure 5. Quantization in training and testing. (a) During training, uniform noise is added to the compressed representation. (b) During testing, uniform quantization is restored as usual.

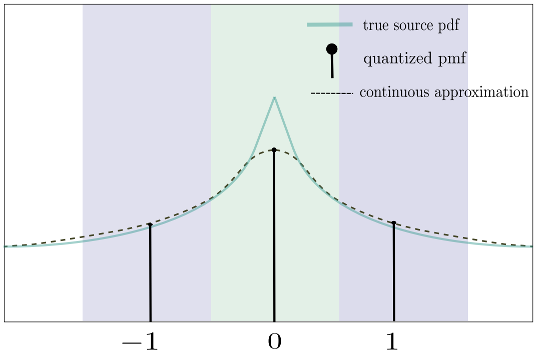


Figure 6. Effect of quantization and uniform noise addition for a given source distribution. Continuous line denotes the pdf of true source a. After adding uniform noise, the pdf of $a + \Delta a$ is represented by dotted line. Quantized source $\mathsf{Q}(a)$ has pmf denoted by stem plot.

$\mathcal{U}(-0.5, 0.5)$ in training phase. This is illustrated in figure 5. The resulting $\widetilde{\boldsymbol{z}} = \boldsymbol{z} + \Delta\boldsymbol{z}$ becomes continuous approximation of $\widehat{\boldsymbol{z}}$. During testing, we quantize using Q as usual.

3.2.2. Estimation of Pmf of $\widehat{\boldsymbol{z}}$

In order to evaluate loss function in (6), we wish to find a way to obtain the pmf of $\widehat{\boldsymbol{z}}$. It has been argued in (Ballé et al., 2016) that pdf of $\widetilde{\boldsymbol{z}}$ is a continuous approximation of the pmf of $\widehat{\boldsymbol{z}}$. To illustrate this concept, consider a distribution over a single real number (for e.g. $a \in \mathbb{R}$). Figure 6 shows a case where the pdf of source a is given by continuous line. Stem plots represent the pmf of $\widehat{a} = \mathsf{Q}(a)$. And dotted line represents the pdf of $\widetilde{a} = a + \Delta a$, where $\Delta a \sim \mathcal{U}(-0.5, 0.5)$. It has been shown that $\widetilde{a}$ is a continuous approximation of $\widehat{a}$. Hence pdf of $\widetilde{a}$ can be approximated by a neural network, and evaluating the probability density at integer locations correspond to the evaluating pmf of $\widehat{a}$. Therefore, we can train a neural network based pdf approximator $\boldsymbol{p_\beta}$ that can utilized to evaluate the probability of $\widehat{\boldsymbol{z}}$. Note that $\boldsymbol{p_\beta}$ should be trained alongside our primary feedback network. In this project, we use off-the-shelf library for end-to-end compression in `Pytorch` (Bégaint et al., 2020) that provides pmf estimation network and procedures. We adopt the implementation of design from (Ballé et al., 2018). In the following discussion, we assume that $\boldsymbol{p_\beta}$ provides estimation of the pmf of $\widehat{\boldsymbol{z}}$ and is learned alongside the compression autoencoder.

Figure 6 shows the CSI *FeedBack Network* (`FBNet`) architecture. The notations for network parameters follow the same pattern as in `CENet`. $\boldsymbol{q_{\theta_\mathrm{e}}}$ and $\boldsymbol{g_{\theta_\mathrm{d}}}$ are obtained through training process in (6). $\boldsymbol{p_\beta}$ is obtained from pmf estimation process described earlier. At test time, feedback bits $\boldsymbol{m}$ is obtained by applying $\boldsymbol{q_{\theta_\mathrm{e}}}$ to pilot observation $\boldsymbol{Y}$, quantizing the output using uniform quantization and entropy coding into bits by utilizing $\boldsymbol{p_\beta}$.

4. Simulations

We use COST2100 channel model (Liu et al., 2012) to generate training and test examples for the proposed framework. Pilot signal $\boldsymbol{P}$ is generated by sampling its entries from standard normal distribution. For entropy coding, we use *range asymmetric numeral system* (Duda, 2013) that has high compression optimality at high speed, and fits the need of wireless communication.

Two scenarios are considered for training and testing: indoor picocellular at 5.3GHz band and outdoor rural at 300MHz. Default parameter values in the official implementation of (Liu et al., 2012) is used throughout the experiments. $20 \times 20m^2$ and $400 \times 400m^2$ regions are used for indoor and outdoor environments. For each channel realization, the BS is placed at the center of the region and the UE is randomly placed anywhere inside the region. N_t is set to 32 and N_c is set to 128. For easy comparison data open-sourced from (Wen et al., 2018) is used that was generated using COST2100 model using the aforementioned settings. We use a fixed set of 8 pilot signals throughout the experiments. Note that with these configurations, size of bottleneck in `CENet` is 128 and that of `FBNet` is 32.

`CENet` and `FBNet` are trained using the same 100,000 training examples with validation set of 5,000. Both networks use the same parameters: batch size of 128, ADAM optimizer initialized with learning rate of 0.001, and trained for 200 epochs.

For both networks, we evaluate the quality of reconstruction using *normalized mean square error* (NMSE) estimate defined as follows:

$$\mathrm{NMSE} = \frac{\|\boldsymbol{H} - \widehat{\boldsymbol{H}}\|_\mathrm{F}^2}{\|\boldsymbol{H}\|_\mathrm{F}^2}$$

To gauge the performance for downstream task such as beamforming, we evaluate the cosine similarity between channel gain vectors for each subcarrier. Let $\boldsymbol{h}_j$ denote the channel gain vector for j-th subcarrier. Then $\boldsymbol{H} =$

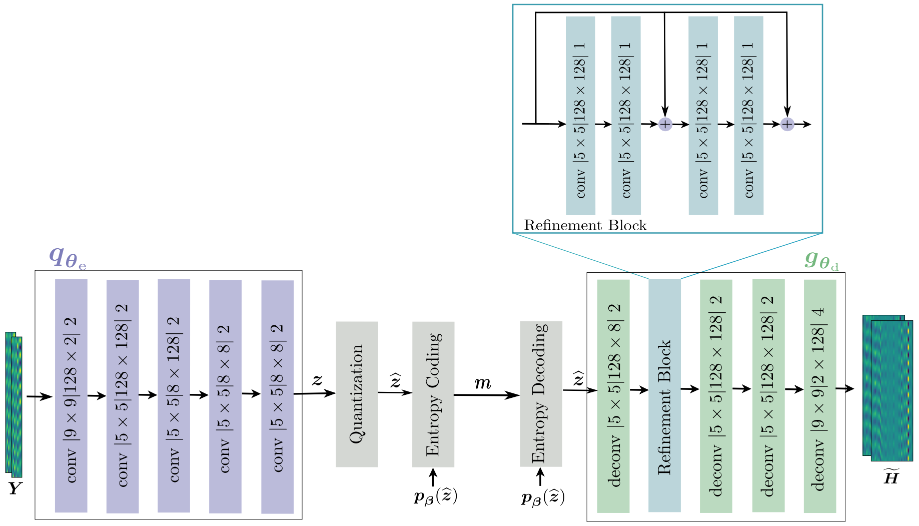


Figure 7. Illustration of network architecture of `FBNet`. tNotations follow figure 3. At UE, pilot observation $\boldsymbol{Y}$ is compressed by pre-trained encoder $\boldsymbol{q}_{\boldsymbol{\theta}_\mathrm{e}}$, quantized uniformly and converted into bits utilizing a learned pmf $\boldsymbol{p}_{\boldsymbol{\beta}}$ for quantized vector $\widehat{\boldsymbol{z}}$. At BS, $\boldsymbol{p}_{\boldsymbol{\beta}}$ is used to recover quantized representation $\widehat{\boldsymbol{z}}$. Finally, the CSI estimate is obtained by passing $\widehat{\boldsymbol{z}}$ through a pre-trained decoder $\boldsymbol{g}_{\boldsymbol{\theta}_\mathrm{d}}$.

$\begin{bmatrix}\boldsymbol{h}_1 & \boldsymbol{h}_2 & \dots \boldsymbol{h}_{N_\mathrm{c}}\end{bmatrix}^\mathsf{T}$ and $\widehat{\boldsymbol{H}} = \begin{bmatrix}\widehat{\boldsymbol{h}}_1 & \widehat{\boldsymbol{h}}_2 & \dots \widehat{\boldsymbol{h}}_{N_\mathrm{c}}\end{bmatrix}^\mathsf{T}$. We denote by $\boldsymbol{\rho}$ the correlation between the channel gain vectors given by:

$$\boldsymbol{\rho} = \frac{1}{N_\mathrm{c}} \sum_{n=1}^{N_\mathrm{c}} \frac{|\widehat{\boldsymbol{h}}_n^* \boldsymbol{h}_n|}{\|\widehat{\boldsymbol{h}}_n\|_2 \|\boldsymbol{h}_n\|_2}$$

Performance of limited feedback also depends upon the number of feedback bits sent per CSI realization. Since we are using a fixed size $\boldsymbol{H}$, i.e. $N_\mathrm{c} \times N_\mathrm{t}$, we directly evaluate the number of bits need to represent a channel matrix given by $|\boldsymbol{m}|$.

We compare proposed method with `CsiNet` proposed in (Wen et al., 2018) which does not consider the effect of quantization during feedback. `CsiNet` is an autoencoder whose encoder is used at the UE for compression of CSI matrix and decoder at the BS for recovery. In the original work, availability of ground truth CSI at the UE is assumed and feedback of the CSI is considered. However, in our experiments we use `CsiNet` on the pilot observation. In order to make a fair comparison, we use the same architecture proposed for `FBNet` ignoring the quantization and entropy coding (only the composition of $\boldsymbol{g}_{\boldsymbol{\theta}_\mathrm{d}}$ and $\boldsymbol{q}_{\boldsymbol{\theta}_\mathrm{e}}$) trained with MSE loss equivalent to (5). In the following experiments we use `CsiNet` with 8-dimensional latent vector with 32-bit floating point represention per dimension.

To evaluate the performance gain from proposed method with respect to the cascaded approach illustrated in figure 2(a), `FBNet` is trained as a regular autoencoder that takes in a CSI matrix and outputs CSI estimate. Then CENet is cascaed with this `FBNet` to form `CascadeNet`.

Model	NMSE(dB)	ρ	$\lvert\boldsymbol{m}\rvert$
`CENet`	-6.39	0.69	NA
`FBNet`	-4.85	0.63	118
`CsiNet`	-2.20	0.46	256
`CascadeNet`	-2.67	0.53	128

Table 1. Evaluation of NMSE, correlation and number of feedback bits $|\boldsymbol{m}|$ of different methods in outdoor environment with $\lambda = 10^5$.

In table 1 and 2, we test all the methods with the same 500 test examples and take the average.

Table 1 shows performance in outdoor environment for all the methods considered. As expected, `CENet` has the smallest NMSE and largest correlation because `CENet` is used at the UE for channel estimation and does not have limit on the bottleneck layer. `FBNet` performs well with only 118 bits in average required to represent a CSI matrix compared to 256 bits for `CsiNet` with poorer reconstruction. There is a huge gain with respect to `CascadeNet` which performs poorer than `FBNet` in CSI reconstruction which shows that cascading the two stages result in error propagation and degraded overall performance.

Table 2 shows performance in indoor environment for all the methods considered. Performance hierarchy basically follows the same pattern as in outdoor environment. Correlation score, however, seems to be similar for all the approaches.

Model	NMSE(dB)	ρ	$\|\boldsymbol{m}\|$
CENet	-3.43	0.74	NA
FBNet	-3.17	0.73	155
CsiNet	-1.73	0.73	256
CascadeNet	-2.15	0.71	168

Table 2. Evaluation of NMSE, correlation and number of feedback bits $|\boldsymbol{m}|$ of different methods in indoor environment with $\lambda = 10^5$.

5. Conclusion

In this report, a framework for channel estimation and limited feedback. The presented framework is based on readings on recent research on the field. Advantage of including rate-distortion as the optimization criteria is demonstrated through experiments. Also, a modification is proposed on the existing work in (Yang et al., 2020) by directly applying encoding to pilot observation. Performance gain from this scheme is also demonstrated in experiments.

References

Alevizos, P. N., Fu, X., Sidiropoulos, N. D., Yang, Y., and Bletsas, A. Limited feedback channel estimation in massive mimo with non-uniform directional dictionaries. *IEEE Transactions on Signal Processing*, 66(19): 5127–5141, 2018.

Balevi, E., Doshi, A., and Andrews, J. G. Massive mimo channel estimation with an untrained deep neural network. *IEEE Transactions on Wireless Communications*, 19(3): 2079–2090, 2020a.

Balevi, E., Doshi, A., Jalal, A., Dimakis, A., and Andrews, J. G. High dimensional channel estimation using deep generative networks. *IEEE Journal on Selected Areas in Communications*, 39(1):18–30, 2020b.

Ballé, J., Laparra, V., and Simoncelli, E. P. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.

Dong, Y., Wang, H., and Yao, Y.-D. Channel estimation for one-bit multiuser massive mimo using conditional gan. *IEEE Communications Letters*, 2020.

Doshi, A., Balevi, E., and Andrews, J. G. Compressed representation of high dimensional channels using deep generative networks. In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5. IEEE, 2020.

Duda, J. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013.

Kuo, P.-H., Kung, H., and Ting, P.-A. Compressive sensing based channel feedback protocols for spatially-correlated massive antenna arrays. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 492–497. IEEE, 2012.

Liu, L., Oestges, C., Poutanen, J., Haneda, K., Vainikainen, P., Quitin, F., Tufvesson, F., and De Doncker, P. The cost 2100 mimo channel model. *IEEE Wireless Communications*, 19(6):92–99, 2012.

Nguyen, S. L. H. and Ghrayeb, A. Compressive sensing-based channel estimation for massive multiuser mimo systems. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2890–2895. IEEE, 2013.

Wen, C.-K., Shih, W.-T., and Jin, S. Deep learning for massive mimo csi feedback. *IEEE Wireless Communications Letters*, 7(5):748–751, 2018.

Yang, Q., Mashhadi, M. B., and Gunduz, D. Distributed deep convolutional compression for massive mimo csi feedback. *arXiv preprint arXiv:2003.04684*, 2020.

Yang, Y., Gao, F., Li, G. Y., and Jian, M. Deep learning-based downlink channel prediction for fdd massive mimo system. *IEEE Communications Letters*, 23(11):1994–1998, 2019.