Scientific Research Publishing

Journal of Computer and Communications, 2023, 11, 143-153
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

A Novel Scheme for Separate Training of Deep Learning-Based CSI Feedback Autoencoders

Lusheng Xi[1], Yanan Yu[1], Jianzhong Yi[1], Chao Dong[1], Kai Niu[1], Qiuping Huang[2,3], Qiubin Gao[2,3], Yongqiang Fei[2,3]

[1]Key Laboratory of Universal Wireless Communication, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, China
[2]CICT Mobile Communications Technology Co., Ltd., Beijing, China
[3]State Key Laboratory of Wireless Mobile Communications, China Academy of Telecommunications Technology (CATT), Beijing, China
Email: xls318027@bupt.edu.cn

How to cite this paper: Xi, L.S., Yu, Y.N., Yi, J.Z., Dong, C., Niu, K., Huang, Q.P., Gao, Q.B. and Fei, Y.Q. (2023) A Novel Scheme for Separate Training of Deep Learning-Based CSI Feedback Autoencoders. *Journal of Computer and Communications*, **11**, 143-153.
https://doi.org/10.4236/jcc.2023.119009

Received: August 10, 2023
Accepted: September 25, 2023
Published: September 28, 2023

Abstract

In this paper, we introduce a novel scheme for the separate training of deep learning-based autoencoders used for Channel State Information (CSI) feedback. Our distinct training approach caters to multiple users and base stations, enabling independent and individualized local training. This ensures the more secure processing of data and algorithms, different from the commonly adopted joint training method. To maintain comparable performance with joint training, we present two distinct training methods: separate training decoder and separate training encoder. It's noteworthy that conducting separate training for the encoder can pose additional challenges, due to its responsibility in acquiring a compressed representation of underlying data features. This complexity makes accommodating multiple pre-trained decoders for just one encoder a demanding task. To overcome this, we design an adaptation layer architecture that effectively minimizes performance losses. Moreover, the flexible training strategy empowers users and base stations to seamlessly incorporate distinct encoder and decoder structures into the system, significantly amplifying the system's scalability.

Keywords

Autoencoder, Joint Training, Separate Training, CSI Feedback

1. Introduction

Deep learning techniques have gained immense popularity in various fields, including wireless communication physical layer [1]. Generative-based models like

Autoencoders [2] have emerged and demonstrated its remarkable ability in data compression and reconstruction. CSI feedback is a crucial communication indicator used to measure channel state information, especially in the frequency division duplexing (FDD) MIMO system [3]. Previous models, such as CsiNet [4], DCGAN [5], and TransNet [6] etc. have achieved significant performance in DL-Based CSI feedback task. However, these papers mainly focus on models of achieving high compression and reconstruction performance, which involves joint training of a single encoder-decoder pair. During joint training, the encoder and decoder collaborate and share information, which can potentially result in privacy leakage and leave the system more vulnerable to adversarial attacks [7]. Moreover, whether in a multi-user or multi-base station system, joint training requires encoders of all users and decoders of all base stations to be trained simultaneously. This results in the need to retrain all models when new users (or new base stations) join (or leave) the system. This not only demands a significant amount of computational resources but also makes it difficult to apply and deploy joint training in large-scale Multi-user Multi-base station system.

Separate training is a novel approach to solve these issues. With separate training, the encoder and decoder of multiple users receive independent and individualized local training and then apply our separate training strategy, the optimization between encoder and decoder is separated. This technique ensures that the encoder cannot learn any specific details related to the decoder, and vice versa. As a result, the privacy of the algorithm and model is better protected. Moreover, separate training is more flexible and allows for easier addition or removal of users or base stations from the system. It also permits greater personalization and customization of different model structures, making system more compatible and scalable.

Our system utilizes separate training in two different scenarios. In the first scenario, multiple User Equipment (UEs) simultaneously transmit CSI to a single Base Station (BS) in what is known as the $N\times 1$ case. For this scenario, we employ a separate training decoder that works in conjunction with multiple pretrained encoders of UEs. In the second scenario, a single UE transmits CSI to multiple BSs in what is known as the $1\times N$ case. For this scenario, our system employs a separate training encoder that works in conjunction with multiple pretrained decoders of BSs. These pretrained models have been obtained through independent one-to-one local joint training. They can provide prior information to facilitate separate training, while ensuring the confidentiality and privacy of the algorithmic models used.

Notation: The channel matrix is denoted by H, and the channel eigenvectors is represented by $\mathbf{v}$, N_t and N_r represents the number of transmitting and receiving antennas, N_c is the number of subcarriers while S represents the number of subbands. The symbol $\mathbb{E}[\cdot]$ denotes statistical expectation.

feedback bits refers to the number of feedback bits, while B suggests the use of B-bit quantization and dequantization. $\mathcal{I}=\{\mathbf{l}_1,\cdots,\mathbf{l}_i,\cdots,\mathbf{l}_n\}$ denotes the latent

space vectors set while the split datasets of training is denoted by $\mathcal{V}=\{\mathbf{v}_1,\cdots,\mathbf{v}_i,\cdots,\mathbf{v}_n\}$.

2. System Model Design

2.1. System Model

We consider a 5G-NR massive MIMO transmission system with spatial-frequency CSI feedback [8]. We use 48 Resource Blocks (RBs) and divide them into $S=12$ subbands, each of which contains 4 RBs in the frequency domain. At the base station (BS), there are 32 transmitting antennas ($N_t=32$), while the User Equipment (UE) is equipped with 4 receiving antennas ($N_r=4$). In UE side, the received signal is denoted as:

$$y_i(f)=H_i(f)\cdot W\cdot s+n.$$

where $H(f)\in\mathbb{C}^{N_r\times N_t}$ denotes the UE's downlink channel response matrix at the i-th subcarrier, $i=1,2,\cdots,N_c$. W is the precoding matrix, s is the bitstream signal, and n denotes the additive noise. We calculate the convariance matrix of s-th subband:

$$R(s)=\frac{1}{N_S}\sum_{i=(s-1)\cdot N_S+1}^{s\cdot N_S}H_i^H(f)H_i(f),s=1,2,\cdots,S \tag{1}$$

where N_S denotes the number of subcarriers on each subband, which equals to N_c/S. The $R(s)$ reflects the average correlation of the channels across different subcarriers in s-th subband. For rank = 1, we obtain the eigenvector $v_s\in\mathbb{C}^{N_t\times 1}$ corresponding to the largest eigenvalue of this $R(s)$. Traverse all subbands, we get the all the eigenvectors of S subbands. The dimensionality of the CSI eigenvectors $\mathbf{v}$ is $N_t\times S\times 2=32\times 12\times 2=768$, where the factor of 2 represents the real and imaginary parts, respectively. In the UE uplink CSI feedback transmission, we employ uniform quantization with B bits of precision. The basic architecture of the autoencoder, which employs CSI eigenvectors for compression and reconstruction, is illustrated in **Figure 1**. Basic architecture of Autoencoder for CSI Eigenvectors compression and reconstruction where r and i denotes real and imaginary of eigenvectors, respectively. As CSI feature vectors serve as our data, we measure the similarity between the recovered vectors and the original vectors using the Square Cosine Similarity (SCS), which can be our metric of reconstruction. We employ a loss function, denoted as $\mathcal{L}$ in this paper, where $\mathcal{L}=1-SCS$: ($\mathbf{v}_{r,i}$ denotes eigenvectors). **Figure 2** presents the process of joint training and separate training (including separate training decoder

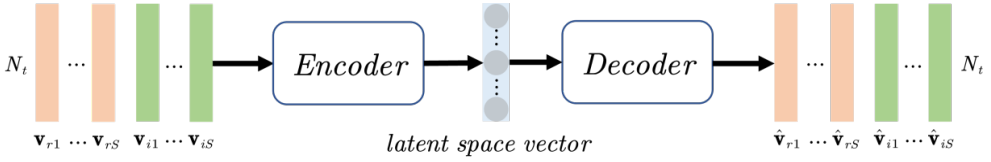


Figure 1. Basic architecture of Autoencoder for CSI Eigenvectors compression and reconstruction where r and i denotes real and imaginary of eigenvectors, respectively.

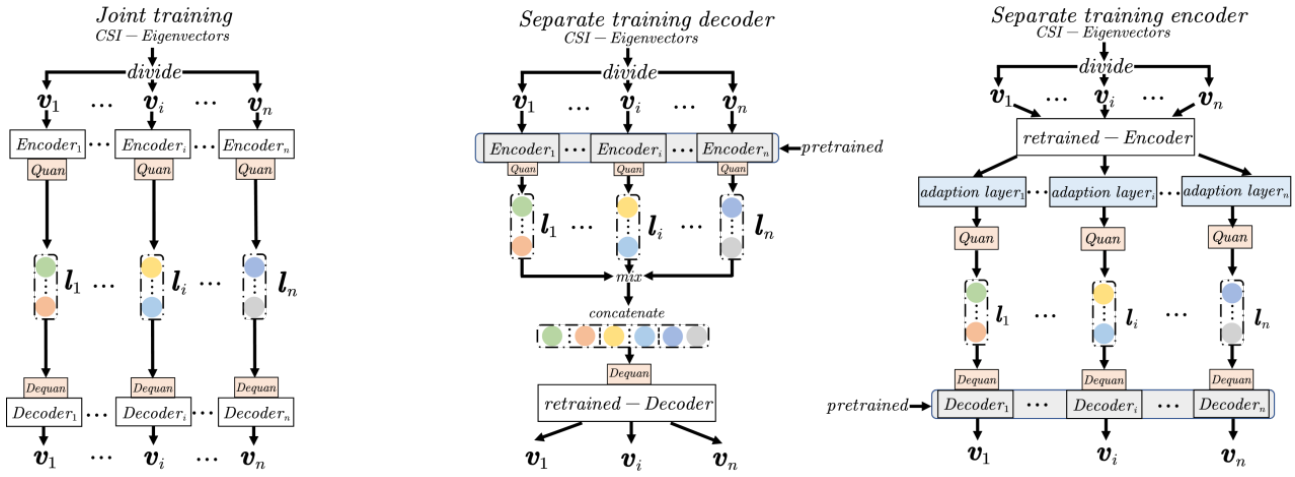


Figure 2. Joint training, Separate training decoder and Separate training encoder. *Quan* and *Dequan* refer to quantization and dequantization, the $\mathbf{l}_i \in \mathcal{I}(i=1,\cdots,n)$ are latent space vectors for split datasets $\mathbf{v}_i \in \mathcal{V}(i=1,\cdots,n)$ respectively.

and separate training encoder). We consider general of n-pairs of encoder-decoder, firstly we divide our dataset to $\mathcal{V}=\{\mathbf{v}_1,\cdots,\mathbf{v}_i,\cdots,\mathbf{v}_n\}$, of which are treated as independent encoder-decoder datasets to jointly train their respective encoder and decoder. We use $E_{\Theta_{E_i}}$ as the i-th encoder and $D_{\Theta_{D_i}}$ as the i-th decoder, the encoder and decoder can be parameterized by Θ_E and Θ_D, where refers to the weights and biases of neural network. And the $\mathcal{I}=\{\mathbf{l}_1,\cdots,\mathbf{l}_i,\cdots,\mathbf{l}_n\}$ are latent space vectors for $\mathcal{V}=\{\mathbf{v}_1,\cdots,\mathbf{v}_i,\cdots,\mathbf{v}_n\}$ respectively.

Joint training involves optimizing n autoencoder models (encoder-decoder pairs) simultaneously on their respective datasets, that is an unsupervised learning approach which simultaneously trains encoder and decoder models. Conversely, separate training involves retraining a single decoder in $N\times 1$ case and with n pre-trained encoders and retraining a single encoder with n pre-trained decoders in $1\times N$ case, as pre-trained encoders and decoders are from Equation (2) where $\mathbf{v}_i \in \mathcal{V}(i=1,\cdots,n)$.

$$\left\{\widehat{\Theta}_{E_i},\widehat{\Theta}_{D_i}\right\}=\underset{\Theta_{E_i}\Theta_{D_i}}{\arg\min}\,\mathbb{E}_{\mathbf{v}_i}\mathcal{L}\left(\mathbf{v}_i,D_{\Theta_{D_i}}\left[E_{\Theta_{E_i}}\left(\mathbf{v}_i\right)\right]\right) \tag{2}$$

2.2. Separate Training Decoder

When separate training decoder, we should retrain a decoder ($D_{\Theta_{D_r}}$) to minimize

$$\widehat{\Theta}_{D_r}=\sum_{i=1}^{n}\underset{\Theta_{D_r}}{\arg\min}\,\mathbb{E}_{\mathbf{v}_i}\mathcal{L}\left(\mathbf{v}_i,D_{\Theta_{D_r}}\left[E_{\Theta_{E_i}}\left(\mathbf{v}_i\right)\right]\right) \tag{3}$$

where $\mathbf{v}_i \in \mathcal{V}(i=1,\cdots,n)$. Since $E_{\Theta_{E_1}},\cdots,E_{\Theta_{E_i}},\cdots,E_{\Theta_{E_n}}$ are pre-trained. $\mathcal{I}=\{\mathbf{l}_1,\cdots,\mathbf{l}_i,\cdots,\mathbf{l}_n\}$ can be obtained by feeding the corresponding split datasets $\mathcal{V}=\{\mathbf{v}_1,\cdots,\mathbf{v}_i,\cdots,\mathbf{v}_n\}$ to respective encoder networks.

$$\mathbf{l}_i=E_{\Theta_{E_i}}\left(\mathbf{v}_i\right),\mathbf{v}_i\in\mathcal{V}(i=1,\cdots,n) \tag{4}$$

Take Equation (4) to Equation (3), we get

$$\widehat{\Theta}_{D_r} = \arg\min_{\Theta_{D_r}} \sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}_i}^{\mathbf{l}_i} \mathcal{L}\left(\mathbf{v}_i, D_{\Theta_{D_r}}\left[\mathbf{l}_i\right]\right) \tag{5}$$

where $\mathbf{v}_i \in \mathcal{V}, \mathbf{l}_i \in \mathcal{I}\left(i=1,\cdots,n\right)$.

The optimized function in Equation (5) implies that by utilizing the generated labels from prior joint training, we can convert the previous unsupervised learning approach of joint training autoencoders into a supervised problem. In this new approach, multiple latent space vectors serve as inputs, and CSI eigenvectors serve as outputs for our retrained-decoder, as illustrated in **Figure 2**. During separate training, we concatenate $\mathbf{l}_1,\cdots,\mathbf{l}_i,\cdots,\mathbf{l}_n$ and $\mathbf{v}_1,\cdots,\mathbf{v}_i,\cdots,\mathbf{v}_n$. Thus, the objective of separate training decoder is to minimize the overall loss in Equation (6).

$$\widehat{\Theta}_{D_r} = \arg\min_{\Theta_{D_r}} \mathbb{E}_{\mathbf{v}}^{\mathbf{l}} \mathcal{L}\left(\mathbf{v}, D_{\Theta_{D_r}}\left[\mathbf{l}\right]\right) \tag{6}$$

Equation (6) indicates the retrained-decoder $D_{\Theta_{D_r}}$ is to reconstruct the original CSI from the latent space vectors which offer prior information about encoders and are generated by $E_{\Theta_{E_1}},\cdots,E_{\Theta_{E_i}},\cdots,E_{\Theta_{E_n}}$. As a result, the retrained-decoder has no idea about the specific weights and biases of each encoders, which can help further protect the privacy of the individual encoder models and algorithms.

Figure 3 portrays the $N\times 1$ separate training decoder system $\left(N=3\right)$, where three User Equipments (UEs) apply CSI feedback to one Base Station (BS). Our model is transformer-based and was inspired by [6] with positional encoding. Our design incorporates three distinct encoder architectures: The first encoder (encoder1) employs a 6-layer transformer block, the second encoder (encoder2) employs a 5-layer transformer block, while the third encoder (encoder3) uses a simple fully connected (FC) layer. The decoder consists of a 6-layer transformer-decoder block.

2.3. Separate Training Encoder

As for separate training encoder, we should retrain a encoder ($E_{\Theta_{E_r}}$) to minimize (where $\mathbf{v}_i \in \mathcal{V}\left(i=1,\cdots,n\right)$.)

$$\widehat{\Theta}_{E_r} = \arg\min_{\Theta_{E_r}} \sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}_i} \mathcal{L}\left(\mathbf{v}_i, D_{\Theta_{D_i}}\left[E_{\Theta_{E_r}}\left(\mathbf{v}_i\right)\right]\right) \tag{7}$$

However, in separate training encoder, we cannot use the same label generation method like separate training decoder because the pre-trained models are

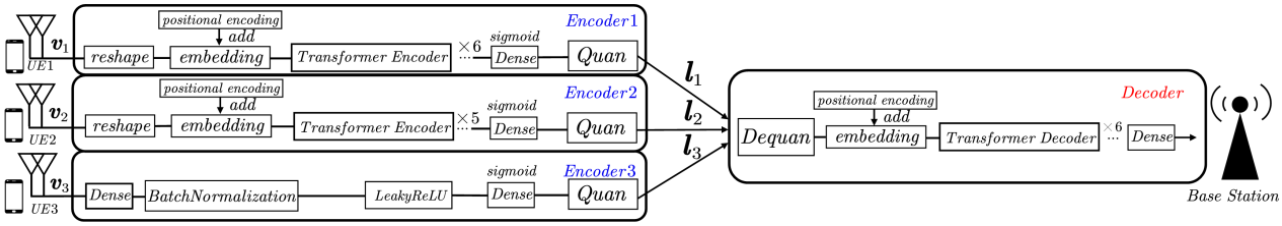


Figure 3. $N\times 1$ case for separate training decoder system $\left(N=3\right)$.

$D_{\Theta_{D_1}}, \cdots, D_{\Theta_{D_i}}, \cdots, D_{\Theta_{D_n}}$. The unidirectional architecture of the encoder-decoder model hinders the direct transformation of information from the decoder into labels for separate training encoder. Moreover, since the encoder is tasked with learning a compressed representation of underlying data features, it poses challenging to fit multiple pre-trained decoders. To overcome this limitation, we have developed an adaptation layer architecture that adapts non-linear transformations for multiple decoders as shown in **Figure 1**, which enables the re-trained encoder to be compatible with multiple decoders.

Figure 4 depicts the $1\times N$ separate training encoder system $(N=3)$, where one User Equipment (UE) apply CSI feedback to three Base Stations (BSs). Our design incorporates three distinct decoder architectures: the first decoder uses a 6-layer transformer block, the second decoder uses a 5-layer transformer block while the thrid decoder is a residual block consisting of 27 layers of $conv(128,(1,3))$ and one layer of $conv(2,(1,3))$. $conv(c,(h_i,w_i))$ represents the convolutional layer with c as the output channels of the CNN layer, and h_i and w_i as the height and width of the kernel size, respectively. The encoder consists of a 6-layer transformer block.

We have developed an adaptation layer based on multilayer perceptron (MLP) that utilizes mixed activation functions. As shown in **Figure 5** MLP-Based Adaption Layer architecture., our approach differs from previous encoders, as it is not directly connected to the quantization layer. Instead, the output of the last fully connected (FC) layer, which has a dimension of $\lfloor feedback\ bits / B \rfloor$ and is activated by a sigmoid function, passes through two fully connected layers forming the adaptation layer. These layers are associated with the relu and tanh activation functions, respectively, providing distinct non-linear transformations. The shapes of these two output layers are denoted by ζ. Before connecting to the quantization layer, we also include another FC layer with an output shape of $\lfloor feedback\ bits / B \rfloor$ and a sigmoid activation function.

Therefore, we propose a general separate training algorithm for the encoder with adaptation layer architecture, which is presented in **Table 1** Algorithm of Separate training encoder with adaption layer. The adaptation layer, denoted by A_{Θ_A}, is parameterized by Θ_A, which includes n different weights and biases adaption layers, namely $A_{\Theta_{A_1}}, \cdots, A_{\Theta_{A_i}}, \cdots, A_{\Theta_{A_n}}$.

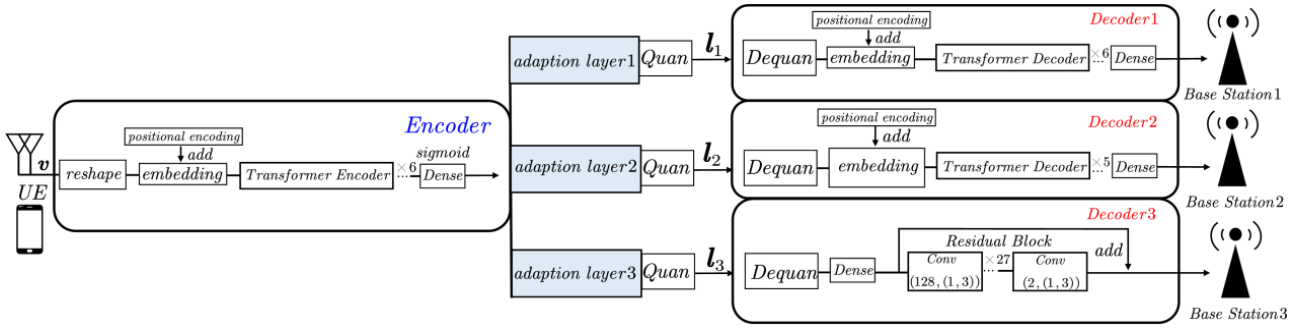


Figure 4. $1\times N$ case for separate training encoder system $(N=3)$.

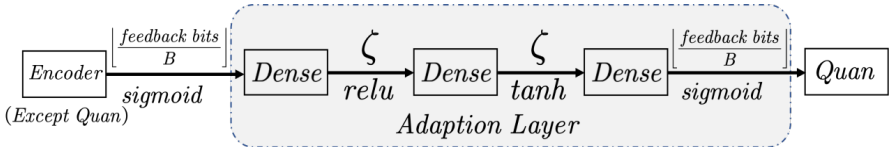


Figure 5. MLP-Based Adaption Layer architecture.

Table 1. Algorithm of Separate training encoder with adaption layer.

1: Initialize the encoder parameters Θ_{E_r}, weights and biases of n adaption layers: $A_{\Theta_{A_i}} = \{w_{A_i}, b_{A_i}\}(i = 1, \cdots, n)$. Freeze all parameters of $D_{\Theta_{D_1}}, \cdots, D_{\Theta_{D_n}}$.

2: **for** each epoch **do**

3: Sample $\mathbf{v}_i \in d_i (i = 1, \cdots, n)$

4: Traverse $\theta \in \{\Theta_{E_r}, \Theta_{A_1}, \cdots, \Theta_{A_i}, \cdots, \Theta_{A_n}\}$

5: Apply Backpropagation to update θ for all adaption layers and encoder.

$$\theta \leftarrow \theta - \eta \nabla_\theta \left\{ \sum_{i=1}^{n} \mathbb{E}_{\mathbf{v}_i} \mathcal{L}\left(\mathbf{v}_i, D_{\Theta_{D_i}}\left[A_{\Theta_{A_i}}\left(E_{\Theta_{E_r}}(\mathbf{v}_i)\right)\right]\right)\right\} \tag{8}$$

where $\mathbf{v}_i \in \mathcal{V}\{\mathbf{v}_1, \cdots, \mathbf{v}_i, \cdots, \mathbf{v}_n\}(i = 1, \cdots, n)$

By Equation (8), we can observe that the separate training of $E_{\Theta_{E_r}}$ involves feed forwarding and gradient receiving from $D_{\Theta_{D_1}}, \cdots, D_{\Theta_{D_i}}, \cdots, D_{\Theta_{D_n}}$, while keeping the decoders' weights and biases private. The adaptation layer architecture is designed to minimize the performance loss when adapting an encoder to multiple pre-trained decoders, ensuring that the system can maintain high performance levels while preserving the privacy of the decoder models. By doing so, the method offers greater flexibility and security compared to joint training. Moreover, Equation (8) also reveals that the whole data $\mathcal{V}$ is to optimize $E_{\Theta_{E_r}}$ while $A_{\Theta_{A_1}}, \cdots, A_{\Theta_{A_i}}, \cdots, A_{\Theta_{A_n}}$ is updated by their respective split datasets $\mathcal{V} = \{\mathbf{v}_1, \cdots, \mathbf{v}_i, \cdots, \mathbf{v}_n\}$, which proves that the design of adaption layer $A_{\Theta_{A_1}}, \cdots, A_{\Theta_{A_i}}, \cdots, A_{\Theta_{A_n}}$ is to help the encoder $E_{\Theta_{E_r}}$ learn the common underlying features of the data while adaption layer coordinates the compatibility between the $E_{\Theta_{E_r}}$ and $D_{\Theta_{D_1}}, \cdots, D_{\Theta_{D_i}}, \cdots, D_{\Theta_{D_n}}$.

3. Results and Analysis

3.1. Dataset and Hyperparameters Configuration

Our dataset was generated based on the Clustered Delay Line (CDL) channel at Urban Macro(UMa) scenario with a carrier frequency of 2 GHz, bandwidth of 10 MHz, and carrier spacing of 15 KHz, as per the Scenarios and Requirements for AI-enhanced CSI from 3GPP Release 16 discussion. The UE receiver settings included 80% indoor (3 km/h) and 20% outdoor (30 km/h). The CSI eigenvectors are acquired as described in Section 2.1. Each dataset ($\mathcal{V} = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$) consists of 100,000 training and 12,000 validation samples, and 60,000 testing samples are used to evaluate the performance of models from both joint training

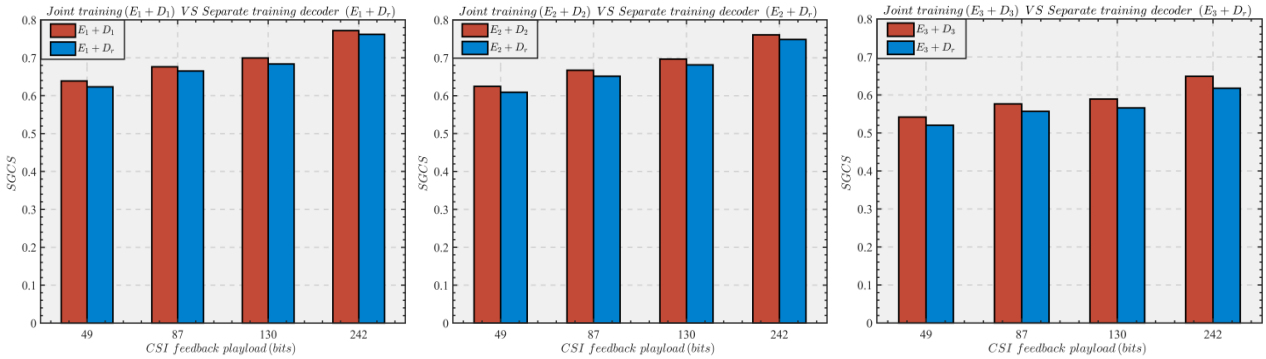
and separate training. The transformer's linear embedding dimension was set to $e_{dim} = 384$, the shapes of these two output adaption layers are $\zeta = 4e_{dim} = 1536$. Moreover, we set the number of attention heads for multi-head attention to be 8. We trained our models using 2-bit uniform quantization ($B = 2$) and Adam optimization algorithm. The initial learning rate was set to 1e−4 and decreased by half every 40 epochs. We trained our models on a single NVIDIA 2080 Ti GPU with a batch size of 64 and up to a maximum of 200 epochs. We utilized the loss function \$1−SCS\$, as specified in Section 2.1. The more detailed data description and open source codes are available at https://github.com/xls318027/CSI-Separate-training.

We consider 4 different CSI feedback payload bits of 49, 87, 130 and 242. The output shape of encoder before quantization layer is $\lfloor feedback\ bits / B \rfloor$, which implies the compression ratio η of our model is 24/768, 43/768, 65/768 and 121/768, respectively. Our performance metric is Square Generalized Cosine Similarity (*SGCS*) where denotes the average square cosine similarity of S subbands: (r, i for real and imaginary parts of eigenvectors while s denotes s-th subbands.)

$$SGCS = \mathbb{E}\left\{ \frac{1}{S} \sum_{s=1}^{S} \left(\frac{\left\| \left(\hat{\mathbf{v}}_{r,i}^{s} \right)^{H} \mathbf{v}_{r,i}^{s} \right\|}{\left\| \hat{\mathbf{v}}_{r,i}^{s} \right\| \left\| \mathbf{v}_{r,i}^{s} \right\|} \right)^{2} \right\} \tag{9}$$

3.2. Performance Comparison Result between Joint Training and Separate Training

Figure 6 presents the results of comparing joint training and separate training decoder on different feedback bits. The retrained decoder D_r exhibits an average decrease in SGCS performance of 0.0131, 0.0146, and 0.0239 on E_1, E_2, and E_3, compared to the respective joint training decoders D_1, D_2, and D_3. This corresponds to a decrease of 1.90%, 2.15%, and 3.99% in terms of percentage. The results illustrate that the method of separately training the decoder in



(a) $E_1 + D_1$ VS $E_1 + D_r$ (b) $E_2 + D_2$ VS $E_2 + D_r$ (c) $E_3 + D_3$ VS $E_3 + D_r$

Figure 6. SGCS performance between joint training and separate training decoder.

Equation (6) provides a slight decrease in SGCS performance compared with joint training. This minor decrease shows that separate training decoder algorithm makes different users independently and freely train their respective encoder models without having to retrain all models as in the case of joint training of $N \times 1$ system when new users join or leave.

Figure 7 presents comparing joint training and separate training encoders on different feedback bits, with and without adaptation layer. The retrained encoder E_r with respective adaption layer A_1, A_2, A_3 exhibited an average decrease in SGCS performance of 0.0142, 0.0201, and 0.0212 on D_1, D_2 and D_3, compared to the respective joint training encoders E_1, E_2, and E_3. This represents a decrease of 1.94%, 2.77%, and 3.17% in terms of percentage.

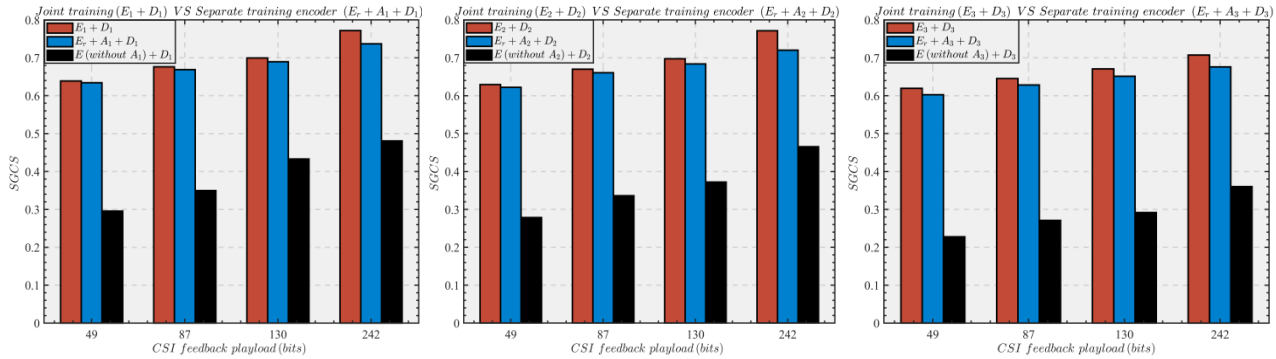
In contrast, ablation experiment without adaptation layer indicates the performance of $E(without\ A_i)+D_i\ (i=1,2,3)$ decreased significantly by 0.3070, 0.3290, and 0.3731, representing a decrease of 44.48%, 47.98%, and 56.72% respectively.

Our experimental results demonstrate that the use of the adaption layer design algorithm presented in **Table 1**. For separately training encoder models offers only a marginal reduction in SGCS compared to joint training. However, the non-adaption method leads significantly worse performance. Those results shows that not only the separate training encoder enables each user to selectively switch between communicating base stations without altering the decoders at individual base stations but also our proposed algorithm with adaption layer could significantly mitigates the performance loss of separate training.

3.3. Influence of Feedback Bits, Compatibility of Different Models

The performance of the separately training decoder shows a relatively small impact with different feedback bits of 49, 87, 130, and 242, resulting in a performance decrease of 0.0171, 0.0156, 0.0180, and 0.0178, respectively. This corresponds to a percentage decrease of 2.92%, 2.48%, 2.77%, and 2.56%.

These results suggest that the concatenation of latent space vectors for the



(a) $E_1+D_1 VS\ E_r+A_1+D_1$ $VS\ E(without\ A_1)+D_1$ (b) $E_2+D_2 VS\ E_r+A_2+D_2$ $VS\ E(without\ A_2)+D_2$ (c) $E_3+D_3 VS\ E_r+A_3+D_3$ $VS\ E(without\ A_3)+D_3$

Figure 7. SGCS performance among joint training, separate training encoder with and without adaption layer.

separate training decoder is only slightly influenced by the feedback bits. In contrast, the separate training encoder is significantly impacted, exhibiting performance decreases of 0.0095, 0.0113, 0.0143, and 0.0391, respectively, which correspond to percentage decreases of 1.52%, 1.71%, 2.08%, and 5.19%. This demonstrates that as feedback bits increase, the output shape of our fixed adaptation output shape ζ, with a size of 1536, also decreases in performance due to the increased value of $\lfloor \textit{feedback bits} / B \rfloor$. Thus, when considering $1 \times N$ adaption layer, we should make set appropriate adaption layer settings for achieving considerable performance.

Both separate training decoder and separately training encoder demonstrate that when the encoder and decoder have different architectures, it results in a more noticeable performance decrease, as observed in the cases of $E_3 + D_r$ for separately training decoder and $E_r + A_3 + D_3$ for separately training encoder. Therefore, there should be a consideration of the trade-off between model compatibility and complexity.

4. Conclusion

Our paper has introduced a new method for training DL-based CSI feedback autoencoders separately. We have proposed the use of concatenated latent space vectors for separate training decoder and a unique adaption layer for separate training encoder. Through a series of comprehensive comparative experiments, we have shown that separate training can achieve similar performance to joint training while providing additional benefits such as improving protection of model and algorithm privacy and enhancing system scalability due to its flexible mechanism of independently local training and sufficient separate training strategies.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant (No. 92067202), Grant (No. 62071058) and CICT Mobile Communication Technology Co., Ltd.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Wang, T., Wen, C.K., Wang, H., Gao, F., Jiang, T. and Jin, S. (2017) Deep Learning for Wireless Physical Layer: Opportunities and Challenges. *China Communications*, **14**, 92-111. https://doi.org/10.1109/CC.2017.8233654

[2] Wang, W., Huang, Y., Wang, Y. and Wang, L. (2014) Generalized Autoencoder: A Neural Network Framework for Dimensionality Reduction. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 490-497. https://doi.org/10.1109/CVPRW.2014.79

[3] Nasser, A., Elsabrouty, M. and Muta, O. (2018) Alternative Direction for 3D Orthogonal Frequency Division Multiplexing Massive MIMO FDD Channel Estimation and Feedback. *IET Communications*, **12**, 1380-1388. https://doi.org/10.1049/iet-com.2017.0916

[4] Wen, C.K., Shih, W.T. and Jin, S. (2018) Deep Learning for Massive MIMO CSI Feedback. *IEEE Wireless Communications Letters*, **7**, 748-751. https://doi.org/10.1109/LWC.2018.2818160

[5] Tolba, B., Elsabrouty, M., Abdu-Aguye, M.G., Gacanin, H. and Kasem, H.M. (2020) Massive MIMO CSI Feedback Based on Generative Adversarial Network. *IEEE Communications Letters*, **24**, 2805-2808. https://doi.org/10.1109/LCOMM.2020.3017188

[6] Cui, Y., Guo, A. and Song, C. (2022) TransNet: Full Attention Network for CSI Feedback in FDD Massive MIMO System. *IEEE Wireless Communications Letters*, **11**, 903-907. https://doi.org/10.1109/LWC.2022.3149416

[7] Liu, Q., Guo, J., Wen, C.K. and Jin, S. (2020) Adversarial Attack on DL-Based Massive MIMO CSI Feedback. *Journal of Communications and Networks*, **22**, 230-235. https://doi.org/10.1109/JCN.2020.000016

[8] Guo, J., Wen, C.K., Jin, S. and Li, G.Y. (2022) Overview of Deep Learning-Based CSI Feedback in Massive MIMO Systems. *IEEE Transactions on Communications*, **70**, 8017-8045. https://doi.org/10.1109/TCOMM.2022.3217777