

Attention Mechanism Based Intelligent Channel Feedback for mmWave Massive MIMO Systems

Yibin Zhang, Jinlong Sun, *Member, IEEE*, Guan Gui, *Senior Member, IEEE*, Yun Lin, *Member, IEEE*, Haris Gacanin, *Fellow, IEEE*, Hikmet Sari, *Life Fellow, IEEE*, and Fumiyuki Adachi, *Life Fellow, IEEE*

arXiv:2208.06570v2 [cs.IT] 2 Nov 2022

Abstract—The potential advantages of intelligent wireless communications with millimeter wave (mmWave) and massive multiple-input multiple-output (MIMO) are based on the availability of instantaneous channel state information (CSI) at the base station (BS). However, no existence of channel reciprocity leads to the difficult acquisition of accurate CSI at the BS in frequency division duplex (FDD) systems. Many researchers explored effective architectures based on deep learning (DL) to solve this problem and proved the success of DL-based solutions. However, existing schemes focused on the acquisition of complete CSI while ignoring the beamforming and precoding operations. In this paper, we propose an intelligent channel feedback architecture using eigenmatrix and eigenvector feedback neural network (EMEVNet). With the help of the attention mechanism, the proposed EMEVNet can be considered as a dual channel auto-encoder, which is able to jointly encode the eigenmatrix and eigenvector into codewords. Simulation results show great performance improvement and robustness with extremely low overhead of the proposed EMEVNet method compared with the traditional DL-based CSI feedback methods.

Index Terms—Attention mechanism, massive MIMO, mmWave, deep learning, channel feedback, beamforming, eigen features.

I. INTRODUCTION

Intelligent wireless communications with millimeter wave (mmWave) band and massive multiple-input multiple-output (MIMO) are considered as the key technology of future communication [1]–[3]. In sixth-generation (6G) of mobile communication, mmWave will play an indispensable and important role [4]–[7]. In addition, massive MIMO combined with the ultra-large bandwidth of mmWave will become a key technical for Internet of Everything (IoE) [8]–[10]. However, all these potential advantages, such as beamforming, power allocation and antenna selection techniques, will be achieved only when the base station (BS) can obtain accurate channel state information (CSI). In time-division duplexing

Yibin Zhang, Jinlong Sun, Guan Gui, and Hikmet Sari are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: 2021010208@njupt.edu.cn, sunjinlong@njupt.edu.cn, guiguan@njupt.edu.cn, hsari@ieee.org).

Yun Lin is with the College of Information and Communication Engineering, Harbin Engineering University, Harbin 150009, China (e-mail: linyun@hrbeu.edu.cn)

Haris Gacanin is with the Institute for Communication Technologies and Embedded Systems, RWTH Aachen University, Aachen 52062, Germany (e-mail: harisg@ice.rwth-aachen.de).

Fumiyuki Adachi is with the International Research Institute of Disaster Science (IRIDeS), Tohoku University, Sendai 980-8572 Japan (e-mail: adachi@ecei.tohoku.ac.jp).

(TDD) systems, the BS can infer the downlink CSI from the uplink CSI with the help of channel reciprocity. Unfortunately, channel reciprocity does not exist in frequency-division duplex (FDD) systems. Therefore, many scholars have been exploring how the BS obtains accurate downlink CSI in the mmWave FDD system in recent years. Compared with the traditional codebook feedback scheme, existing works try to utilize either machine learning (ML) or deep learning (DL) algorithms to obtain CSI at the BS.

Different from the existing codebook feedback scheme, many researches [11]–[19] devoted to exploring more accurate CSI feedback schemes. C. Wen *et al.* [11] innovatively proposed a DL based CSI compression and feedback method, i.e., CsiNet. The proposed CsiNet method compresses the downlink CSI estimated by the user equipment (UE), and then transmits the compressed codewords to the BS. And the BS obtains the accurate downlink CSI after decoding the codewords. J. Guo *et al.* [13] further explored the solution of multiple compression ratios on the basis of CsiNet, which can adapt to different channel environments. T. Wang *et al.* [14] explored the time correlation between CSI matrix and proposed to use long short-term memory (LSTM) algorithm to improve the performance of CsiNet. J. Guo *et al.* [12] summarized the researches using DL methods for CSI feedback in recent years, and claimed that the overhead of DL is too large to be deployed with conventional BS. Therefore, Y. Sun *et al.* [15], [16] explored an efficient algorithm for lightweight design, aiming to reduce the overhead of CsiNet. M. Chen *et al.* [19] proposed a DL-based implicit feedback architecture to inherit the low-overhead characteristic for wideband systems. Different from the CsiNet, J. Zeng *et al.* [17] explored a transfer learning-based fully convolutional network designed to satisfy different channel environments.

Although the DL-based CSI feedback schemes can help the BS to obtain more accurate downlink CSI, they still face the challenge of transmission overhead and utilization of spectrum resources. Furthermore, Z. Zhong *et al.* [20] pointed out that there is no full channel reciprocity in FDD system but partial channel reciprocity still exists. Hence, many researches [21]–[25] still insisted on exploring the solution of predicting downlink CSI from uplink CSI. Y. Yang *et al.* [21] proposed a sparse complex-valued neural network (SCNet) to approximate the mapping function between uplink and downlink CSI so as to reduce the transmission overhead. Meanwhile, they proposed intelligent algorithms based on meta-learning and transfer learning for multiple different wireless communication environments [22], in order to address the problem of limited

datasets in new scenarios. Considering that the CSI matrix can be seen as in-phase/quadrature (I/Q) signal, Y. Zhang *et al.* [24] introduced a complex network to excavate the implicit information between the two channels of I/Q signal and to improve the overall performance. Y. Yang *et al.* [25] developed a systematic framework based on deep multimodal learning to predict CSI by multi-source sensing information.

Neither of the feedback and prediction solutions described above is perfect. The CSI feedback scheme will cause extra spectrum overhead, and the accuracy of the CSI prediction scheme is not very good. Hence, J. Wang *et al.* [26] proposed a compromise solution called SampleDL. The SampleDL requires the user equipment (UE) to transmit sampled downlink CSI to assist the BS in improving the prediction accuracy. The sampleDL aims to combine the advantages of feedback and prediction, which may reduce the feedback overhead and improve the system performance. Recently, some researches have paid attention to the application of downlink CSI. They conduct more specific research for the following beamforming module at the BS, instead of improving the acquiring accuracy of CSI [27]–[31]. W. Liu *et al.* [27] focused on the application of eigenvectors and proposed EVCsiNet to compress and feedback eigenvectors. J. Guo *et al.* [29] explored the feedback schemes designed for beamforming (CsiFBnet) in both single-cell and multi-cell scenarios. Z. Liu *et al.* [30] proposed a novel deep unsupervised learning-based approach to optimize antenna selection and hybrid beamforming.

In this paper, we pay more attention on the eigenvector and eigenmatrix obtained by singular value decomposition (SVD) transformation. This paper proposes an attention mechanism based intelligent channel feedback method designed for beamforming at the BS. Considering the applications of downlink CSI at the BS, each UE is required to transmit useful and effective information to the BS rather than the downlink CSI. The main contributions of this paper are summarized as follows:

- We propose a CSI feedback architecture designed for beamforming, where SVD transformation is utilized as a pre-processing module for CSI matrix.
- We propose a two-channel compressed feedback network using residual attention mechanism, which is suitable for the joint coding of multi-channel heterogeneous data.
- We improve the reconstruction performance of codewords at the BS with respect to switching different auto-encoders for different channel types.
- Comparing with classical methods, the proposed method obtains better reconstruction performance with extremely low feedback overhead, to verify the robustness of our proposed architecture.

II. SYSTEM MODEL AND PROBLEM FORMULATION

This section introduces the system model researched in this paper. First, The link-level channel model is introduced, which is based on 3rd Generation Partnership Project (3GPP) technical report. Then, we introduce SVD transform and its application to beamforming technology and precoding matrix

acquisition. Finally, the scientific issues to be addressed in this paper are described in detail.

A. Link-level Channel Model

Considering a typical mmWave FDD MIMO communication system, we assume that the BS is equipped with N_t antennas in the form of uniform linear array (ULA)¹ and the UE is equipped with N_r antennas ($N_t \gg N_r$). Meanwhile, the orthogonal frequency division multiplexing (OFDM) technique is applied to the link-level channel model. Then, the received signal at the UE can be expressed as,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (1)$$

where $\mathbf{H} \in \mathbb{C}^{N_{RB} \times N_r \times N_t}$ is the downlink CSI between the BS and the UE, \mathbf{n} denotes the noise vector. For an OFDM system, it is necessary to consider multiple subcarriers and OFDM symbols. In this paper, resource blocks (RBs) are used as the channel matrix resolution. Thus, N_{RB} represents the number of RBs used in link-level channel model. Considering a single RB and a pair of transmit and receive antennas, a common multi-path fading channel model [32] is used and can be expressed as,

$$\mathbf{H} = \sum_{n=1}^N \sum_{m=1}^M \sqrt{P_{n,m}} [c_{n,m} \cdot e^{(j2\pi v_{n,m} t)} \boldsymbol{\alpha}(\theta_{n,m})] \quad (2)$$

where N and M denotes the number of scattering clusters and ray pathes, respectively, $P_{n,m}$ represents the power of m -th ray in the n -th scattering cluster, $c_{n,m}$ is the coefficient calculated by field patterns and initial random phases, $\theta_{n,m}$ is the corresponding azimuth angle-of-departure (AoD) of ray path, $v_{n,m}$ stands for the speed and can be understood as Doppler shift parameter. Then, the steering vector $\boldsymbol{\alpha}(\theta_{n,m}) \in \mathbb{C}^{N_t \times 1}$ can be formulated as,

$$\boldsymbol{\alpha}(\theta) = \left[1, e^{-j2\pi \frac{d}{\lambda} \sin(\theta)}, \dots, e^{-j2\pi \frac{(N_t-1)d}{\lambda} \sin(\theta)} \right] \quad (3)$$

where d and λ are the antenna element spacing and carrier wavelength, respectively.

After channel model, we further discuss the probability distribution of LOS channel. Considering an urban macro (UMa) scenario defined by 3GPP TR38.901 [33], we assume that the plane straight-line distance from the UE to the BS is d_{2D} and the LOS probability is Pr_{LOS} . If $d_{2D} \leq 18$ m, then $\text{Pr}_{LOS} = 1$, else the Pr_{LOS} can be calculated via

$$\text{Pr}_{LOS} = \left[\frac{18}{d_{2D}} + \exp\left(-\frac{d_{2D}}{63}\right) \left(1 - \frac{18}{d_{2D}}\right) \right] \cdot \left[1 + 0.8 \cdot C(h_{UT}) \left(\frac{d_{2D}}{100}\right)^3 \exp\left(-\frac{d_{2D}}{150}\right) \right] \quad (4)$$

where the $C(h_{UT})$ can be found in (5), and the h_{UT} denotes the antenna height for the UE.

$$C(h_{UT}) = \begin{cases} 0, & h_{UT} \leq 13 \text{ m} \\ \left(\frac{h_{UT}-13}{10}\right)^{1.5}, & 13 \text{ m} \leq h_{UT} \leq 28 \text{ m} \end{cases} \quad (5)$$

¹We adopt the ULA model here for simpler illustration, nevertheless, the proposed approach does not restrict to the specifical array shape.

To sum up, NLOS channel is a more common scenario with the popularization of mmWave systems.

B. Applications of SVD Transformation

This subsection shows the advantages of SVD transformation and its application in wireless communication. In order to reduce the conflict between multi-rays and increase the channel capacity in massive MIMO system, the transmitter needs to use the beamforming technology to precode the data flow according to the quality of channel. A conventional precoding matrix is based on SVD transformation of CSI matrix.

Considering channel model mentioned above with CSI matrix as $\mathbf{H} \in \mathbb{C}^{N_{RB} \times N_r \times N_t}$. For simplicity of description, we discuss only one RB here², i.e. $RB = 1$ and $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$. First, the CSI matrix \mathbf{H} should carry on SVD transformation as

$$\mathbf{H} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^* \quad (6)$$

where $\mathbf{U} \in \mathbb{C}^{N_r \times N_r}$ and $\mathbf{V} \in \mathbb{C}^{N_t \times N_t}$ are the left-singular and the right-singular matrices³, respectively. What is more, $\mathbf{U}\mathbf{U}^* = \mathbf{I}_{N_r}$, $\mathbf{V}\mathbf{V}^* = \mathbf{I}_{N_t}$ ⁴. Note that $\Sigma = (\Lambda, 0)$ and Λ can be expressed as follows:

$$\Lambda = \begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sqrt{\lambda_{N_r}} \end{pmatrix}_{N_r \times N_r} \quad (7)$$

which represents the singular value matrix. And we define the eigenvalues of $\mathbf{H}\mathbf{H}^*$ as $\mathbf{s} = [\lambda_1, \lambda_2, \dots, \lambda_{N_r}]$. Next, the application of SVD transformation is introduced in detail. The unitary matrices \mathbf{V} and \mathbf{U} are used as precoding matrices for transmitter and receiver, respectively. When BS needs to sent the parallel data flow $\mathbf{x} = [x_1, x_2, \dots, x_{N_t}]^T$ to multiuser, right-singular matrix \mathbf{V} will be used for precoding: $\mathbf{x}_t = \mathbf{V} \cdot \mathbf{x}$. Thirdly, we consider a typical signal transmission model as

$$\mathbf{y} = \mathbf{H}\mathbf{x}_t + \mathbf{n} \quad (8)$$

where \mathbf{y} is the received data flow and \mathbf{n} denotes the noise vector. The channel matrix \mathbf{H} can be expressed by (6), and we can obtain

$$\begin{aligned} \mathbf{y} &= \mathbf{U}\Sigma\mathbf{V}^*\mathbf{V} \cdot \mathbf{x} + \mathbf{n} \\ &= \mathbf{U}\Sigma \cdot \mathbf{x} + \mathbf{n} \end{aligned} \quad (9)$$

Finally, the receiver will use \mathbf{U}^* for receiver combining, which can be expressed as

$$\begin{aligned} \mathbf{U}^*\mathbf{y} &= \mathbf{U}^*(\mathbf{U}\Sigma \cdot \mathbf{x} + \mathbf{n}) \\ &= \Sigma\mathbf{x} + \mathbf{U}^*\mathbf{n} \end{aligned} \quad (10)$$

The noise component in (10) will be filtered out by the receiver. Therefore, the receiver can recover the data flow \mathbf{x} by Σ .

In summary, as a transmitter, the BS should pay more attention to eigenmatrix \mathbf{V} , which can support the following precoding module. And eigenvector Σ should be applied to receiver combining the data flow when the BS is a receiver.

²This assumption is only for the brief illustration of SVD, and this solution is also applicable to OFDM systems.

³Both \mathbf{U} and \mathbf{V} will be called as eigenmatrix in the follows.

⁴ \mathbf{X}^* denotes conjugate transpose matrix of \mathbf{X} .

C. Problem Formulation

As analyzed above, the CSI matrix \mathbf{H} of a massive MIMO system is too complex to compress and reconstruct at the BS accurately. Meanwhile, considering the specific application of downlink CSI at the BS, we find that the BS prefers to obtain a perfect eigenmatrix \mathbf{V} (right-singular matrix) and eigenvector \mathbf{s} . What's more, the eigenmatrix \mathbf{V} is unitary matrix which is symmetric and easily compressible, and the eigenvector \mathbf{s} is a simple real-value vector. Therefore, this paper proposes to jointly compress the unitary matrix \mathbf{V} and the corresponding eigenvector \mathbf{s} , and feed back the codeword to the BS.

Although downlink channel estimation is challenging, this topic is beyond the scope of this paper. We assume that perfect CSI has been acquired and focus on the feedback scheme. Considering a classical mmWave massive MIMO FDD system described above, we focus on the eigenmatrix $\mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}$ and the eigenvector $\mathbf{S} \in \mathbb{R}^{N_{RB} \times N_r}$. The UE needs to deploy an encoder to jointly encode \mathbf{V} and \mathbf{S} , which can be formulated as,

$$\varepsilon = f_{en}(\mathbf{V}, \mathbf{S}, \Theta_{en}) \quad (11)$$

where ε represents the codewords encoded by the UE, Θ_{en} is the weight parameter of the encoder, and $f_{en}(\cdot)$ stands for the framework of encoder. The role of the encoder is to extract high-dimensional features from \mathbf{V} and \mathbf{S} respectively, and match a suitable mapping function to convert them into codewords. When received the codewords ε , the BS needs to switch the corresponding decoder to interpret and obtain the required \mathbf{V} and \mathbf{S} . The decoder can be expressed as,

$$\widehat{\mathbf{V}}, \widehat{\mathbf{S}} = f_{de}(f_{en}(\mathbf{V}, \mathbf{S}, \Theta_{en}), \Theta_{de}) \quad (12)$$

where $\widehat{\mathbf{V}}, \widehat{\mathbf{S}}$ are the reconstructed eigenmatrix and eigenvector at the BS, $f_{de}(\cdot)$ denotes the framework of decoder, and Θ_{de} is the corresponding weight value.

This paper utilizes neural network (NN) based encoder and decoder to complete the compression and feedback of \mathbf{V} and \mathbf{S} . When training NN based auto-encoder, the loss function used by the optimizer is mean squared error (MSE) and can be expressed as,

$$MSE = \mathbb{E} \left[\Gamma \left(\|\mathbf{V} - \widehat{\mathbf{V}}\|_2^2, \|\mathbf{S} - \widehat{\mathbf{S}}\|_2^2 \right) \right] \quad (13)$$

where $\mathbb{E}(\cdot)$ stands for mathematical expectation, $\|\cdot\|_2^2$ denotes the Euclidean norm, and $\Gamma(\cdot)$ is a joint loss estimation function, weighted average function in general. The main problem explored is to solve the optimal weights of the NN-based encoder and decoder, which can be formulated by,

$$(\Theta_{en}^*, \Theta_{de}^*) = \arg \min_{\Theta_{en}, \Theta_{de}} \mathbb{E} \left[\Gamma \left(\|\mathbf{V} - \widehat{\mathbf{V}}\|_2^2, \|\mathbf{S} - \widehat{\mathbf{S}}\|_2^2 \right) \right] \quad (14)$$

where Θ_{en}^* and Θ_{de}^* are the optimal weights of encoder and decoder, respectively.

III. DL-BASED EMEV FEEDBACK ARCHITECTURE

This section describes the proposed DL-based eigenmatrix and eigenvector (EMEV) feedback architecture in detail. Based on the SVD transformation and its application for beamforming, we pay more attention to the eigenmatrix \mathbf{V}

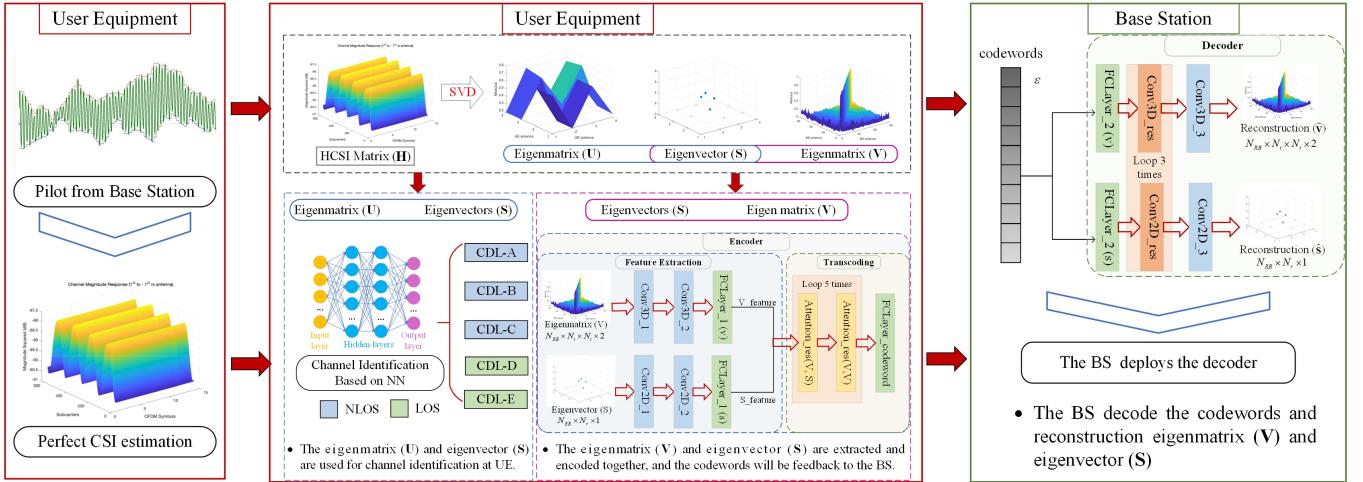


Fig. 1: Illustration of the overview of proposed EMEV feedback architecture. CSI matrix \mathbf{H} is estimated by pilot, and then be divided into eigenmatrix \mathbf{U} , \mathbf{V} and eigenvector \mathbf{S} by SVD transformation. \mathbf{U} and \mathbf{S} are used for channel identification, and the encoder is used to jointly encode \mathbf{V} and \mathbf{S} into codewords ε . The BS deploy the decoder to reconstruct $\hat{\mathbf{V}}, \hat{\mathbf{S}}$.

and eigenvector \mathbf{S} in this paper. First, the overview of proposed EMEV feedback architecture is shown. Then, the NN design for the EMEV auto-encoder is displayed and analyzed by different modules.

A. Overview of The Proposed Architecture

This part is the overview of DL-based EMEV feedback architecture. We aim to explore efficient feedback schemes for beamforming. As is shown in Fig. 1, the whole process starts when the UE estimates the real-time downlink CSI through pilot. And then the CSI matrix \mathbf{H} is divided into eigenmatrix \mathbf{U} , \mathbf{V} and eigenvector \mathbf{S} by SVD transformation. From the figure we can find that \mathbf{H} is complex and irregular, but \mathbf{U} and \mathbf{V} are unitary matrices and \mathbf{S} exhibits a scatter distribution. Furthermore, the power distributions of \mathbf{U} and \mathbf{V} are symmetric. Thirdly, the UE inputs \mathbf{U} and \mathbf{S} to the NN-based channel identification to obtain the exact channel type. The detailed NN-based channel identification is shown in our conference paper [34], called EMEV-IdNet. Considering the clustered delay line (CDL) channel model compliant with 5G new radio (NR) standards [33], we explores five common channel types, composed of three none line of sight (NLOS) channels and two line of sight (LOS) channels. Since the eigenmatrix distributions of five channel types are quite different [35], we cascade the channel identification before EMEV encoder to improve the system performance. After channel identification, the UE will joint-encode \mathbf{V} and \mathbf{S} into codewords and feedback to the BS. As is analyzed above, \mathbf{V} and \mathbf{S} will be able to meet the requirements of beamforming and communication for BS. Finally, the BS receives and decodes the codewords ε and reconstructs eigenmatrix $\hat{\mathbf{V}}$ and eigenvector $\hat{\mathbf{S}}$. The algorithm flow of proposed EMEV feedback architecture is described in **Algorithm 1**.

B. The Proposed DL-based EMEV Feedback Network

This subsection shows the overall framework of proposed DL-based EMEV feedback neural network, called EMEVNet.

Algorithm 1: The algorithm of proposed channel feedback architecture based on EMEV feature.

Input: $\mathbf{H} \in \mathbb{C}^{N_{RB} \times N_r \times N_t} \leftarrow$ CSI matrix;
Output: $\hat{\mathbf{S}} \in \mathbb{R}^{N_{RB} \times N_r} \leftarrow$ Reconstructed eigenvector;
 $\hat{\mathbf{V}} \in \mathbb{C}^{N_{RB} \times N_t \times N_t} \leftarrow$ Reconstructed eigenmatirx; $id \leftarrow$ Channel type;

1 Stage I: UE operations:
2 SVD transformation: Initialize $\mathbf{U} \in \mathbb{C}^{N_{RB} \times N_r \times N_r}$, $\mathbf{S} \in \mathbb{R}^{N_{RB} \times N_r}$, $\mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}$;
3 for $i = 1, \dots, N_{RB}$ **do**
 4 $\mathbf{U}_t, \mathbf{S}_t, \mathbf{V}_t = f_{svd}(\mathbf{H}(i, :, :))$
 5 **if** $\mathbf{U}_t \cdot \mathbf{S}_t \cdot \mathbf{V}_t == \mathbf{H}(i, :, :)$ **then**
 6 | $\mathbf{U}(i, :, :) = \mathbf{U}_t$; $\mathbf{S}(i, :) = \mathbf{S}_t$; $\mathbf{V}(i, :, :) = \mathbf{V}_t$.
 7 **end**
 8 **end**
 9 Save $\mathbf{U}, \mathbf{S}, \mathbf{V}$.
10 Channel identification: Load trained EMEV-IdNet and identify the channel type $id \leftarrow f_{id}(\mathbf{U}, \mathbf{S})$;
11 Encoder: Switch appropriate encoder by id and generate feedback codewords: $\varepsilon = f_{en}(\mathbf{V}, \mathbf{V})$
12 Stage II: BS operations:
13 Decoder: Switch appropriate decoder by id and reconstruct precoding matrix: $(\hat{\mathbf{V}}, \hat{\mathbf{S}}) = f_{de}(\varepsilon)$

Fig. 2 is the illustration of its framework.

As is described in Fig. 2, the EMEVNet is an auto-encoder which is combined with an encoder at the UE and a decoder at the BS. And the encoder can be further divided into feature extraction and transcoding modules. We design a feature extraction module with dual-channel input layer. Different convolution layers are used for different inputs, i.e. three-dimensional convolution layer (Conv3D) is for $\mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}$ and two-dimensional convolution layer (Conv2D) for $\mathbf{S} \in \mathbb{R}^{N_{RB} \times N_r}$. The high-dimensional feature maps after convolution layers will be compressed to

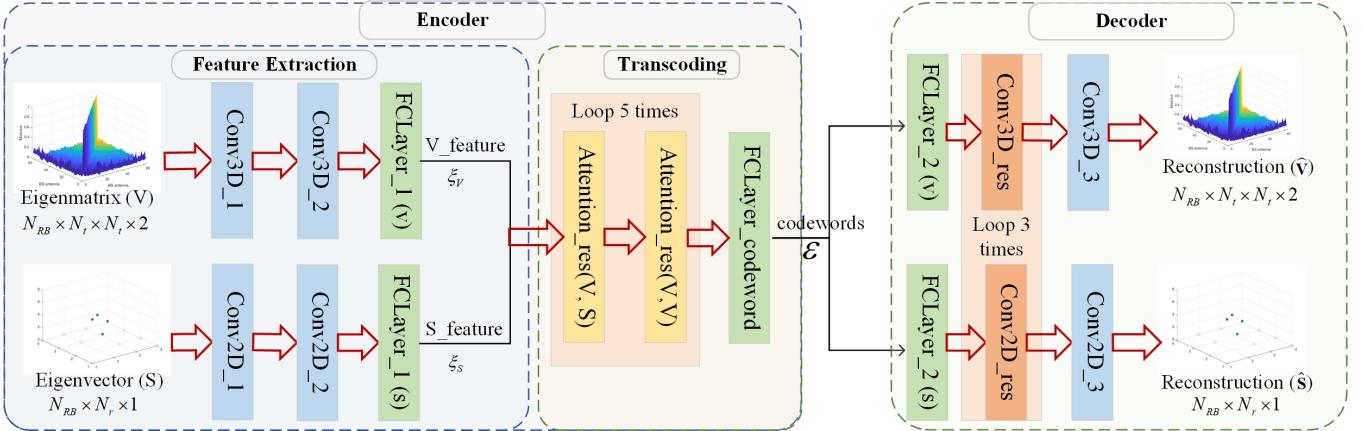


Fig. 2: Illustration of the overall framework of DL-based EMEVNet. Feature extraction module: Input are \mathbf{V} and \mathbf{S} ; Output ξ_V and ξ_S are high-dimensional feature of \mathbf{V} and \mathbf{S} respectively. Transcoding module: Input are ξ_V and ξ_S ; Output is codewords ε . Decoder module: Input is received codewords ε at the BS; Output are reconstructed $\hat{\mathbf{V}}$ and $\hat{\mathbf{S}}$.

one dimensional feature ξ_V and ξ_S by fully-connected layers. Thus, the feature extraction module can be described as,

$$(\xi_V, \xi_S) = \mathcal{L}_{fc} [\mathcal{L}_{conv} (\mathbf{V}, \mathbf{S}, \Omega_{conv}), \Omega_{fc}] \quad (15)$$

where $\mathcal{L}_{fc}(\cdot)$, $\mathcal{L}_{conv}(\cdot)$ represent fully-connected layer and convolution layer respectively, and Ω_{fc} , Ω_{conv} are their corresponding weight values. Then, ξ_V and ξ_S pass through the transcoding module and output codewords ε at specified system compression ratio. The attention mechanism⁵ based attention residual block and fully-connected layer are combined into a transcoding module, which is formulated as,

$$\varepsilon = \mathcal{L}_{fc} \left[\mathcal{L}_{att}^{(5)} (\xi_V, \xi_S, \Omega_{att}), \beta_{CR}, \Omega_{fc} \right] \quad (16)$$

where $\mathcal{L}_{att}^{(5)}$ indicates 5 loops of attention residual block, β_{CR} is the input system compression ratio, and Ω_{att} is the corresponding weight value. Therefore, the length of ε is determined by system β_{CR} , which can be defined as,

$$L_\varepsilon = \frac{L[\Re(\mathbf{V})] + L[\Im(\mathbf{V})] + L[\mathbf{S}]}{\beta_{CR}} + id \quad (17)$$

where $L[\cdot]$ denotes the length of the variable, $\Re(\cdot)$ and $\Im(\cdot)$ are the real and imaginary parts of complex numbers, L_ε is the length of codewords ε , and id represents the control symbol of channel identification result. Finally, the BS can reconstruct eigenmatrix $\hat{\mathbf{V}}$ and eigenvector $\hat{\mathbf{S}}$ from received ε by utilizing decoder. The decoder is composed of fully-connected layers, convolutional residual blocks and convolution layers, which can be written as,

$$(\hat{\mathbf{V}}, \hat{\mathbf{S}}) = \mathcal{L}_{conv} \left\{ \mathcal{L}_{res}^{(3)} [\mathcal{L}_{fc} (\varepsilon, \Omega_{fc}) \Omega_{res}], \Omega_{conv} \right\} \quad (18)$$

where $\mathcal{L}_{res}^{(3)}$ stands for 3 loops of convolutional residual block and Ω_{res} is its weight values. In summary, the concrete steps of EMEVNet algorithm are given in the **Algorithm 2**.

⁵The attention mechanism will be described separately when analyzing the transcoding module.

Algorithm 2: The proposed EMEVNet algorithm for eigenmatrix and eigenvector feedback architecture.

Input: $\mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t} \leftarrow$ Eigenmatrix;
 $\mathbf{S} \in \mathbb{R}^{N_{RB} \times N_r} \leftarrow$ Eigenvector; $\eta \leftarrow$ Initial learning rate; $\tau \leftarrow$ Maximum epoch number;
 $\beta_{CR} \leftarrow$ System compression ratio;
Output: $\hat{\mathbf{V}} \rightarrow$ Reconstructed eigenmatrix; $\hat{\mathbf{S}} \rightarrow$ Reconstructed eigenvector; $(\Theta_{en}, \Theta_{de}) \rightarrow$ Trained autoencoder parameters;

1 Training stage:

- 2 Load $\mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}$, $\mathbf{S} \in \mathbb{R}^{N_{RB} \times N_r}$;
- 3 Randomly initialize NN weight parameters Θ_{en}, Θ_{de} ;
- 4 **for** $t = 1, \dots, \tau$ **do**
- 5 $(\xi_V, \xi_S) = \mathcal{L}_{fc} [\mathcal{L}_{conv} (\mathbf{V}, \mathbf{S}, \Omega_{conv}), \Omega_{fc}]$
- 6 $\varepsilon = \mathcal{L}_{fc} \left[\mathcal{L}_{att}^{(5)} (\xi_V, \xi_S, \Omega_{att}), \beta_{CR}, \Omega_{fc} \right]$
- 7 $\hat{\mathbf{V}}, \hat{\mathbf{S}} = \mathcal{L}_{conv} \left\{ \mathcal{L}_{res}^{(3)} [\mathcal{L}_{fc} (\varepsilon, \Omega_{fc}) \Omega_{res}], \Omega_{conv} \right\}$
- 8 $loss_t = \mathbb{E} \left[\Gamma \left(\|\mathbf{V} - \hat{\mathbf{V}}\|_2^2, \|\mathbf{S} - \hat{\mathbf{S}}\|_2^2 \right) \right]$
- 9 **if** $loss_t$ converges to $loss^*$ **then**
- 10 | break;
- 11 **end**
- 12 **if** $loss_t$ is not updated after 20 loops **then**
- 13 | $\eta = \eta \times 0.7$;
- 14 **end**
- 15 $\Omega \leftarrow \text{Adam}(\Omega, \eta, \nabla loss_t)$
- 16 **end**
- 17 $[\Theta_{en}^*, \Theta_{de}^*] \leftarrow [\Omega_{fc}, \Omega_{att}, \Omega_{conv}, \Omega_{res}]$;
- 18 Save $f_{en}(\beta_{CR}, \Theta_{en}^*)$, $f_{de}(\beta_{CR}, \Theta_{de}^*)$.

C. Analysis and Discussion of EMEVNet

In this subsection, feature extraction, transcoding and decoding modules are discussed one by one. We analyze the proposed NN from two aspects: time complexity and space complexity [36]. Time complexity refers to the number of operations of the model, which can be measured by FLOPs, i.e., the number of floating-point operations. The

TABLE I: The hyper-parameters setting and analysis of parameters and FLOPs for feature extraction module.

Layer name	Hyper-parameters	Activation	Output shape	Parameter size	FLOPs
Input(V)	-	-	$N_{RB} \times N_t \times N_t \times 2$	-	-
Input(S)	-	-	$N_{RB} \times N_r \times 1$	-	-
Conv3D_1	Filter = 2,	Leaky Relu	$N_{RB} \times N_t \times N_t \times 2$	$2 \times 2 \times 3^2$	$(N_{RB} \times N_t \times N_t \times 2) \times (2 \times 3^2)$
Conv2D_1	Kernel = 3.	Leaky Relu	$N_{RB} \times N_t \times 2$	$2 \times 2 \times 3^2$	$(N_{RB} \times N_r \times 2) \times (2 \times 3^2)$
Conv3D_2	Filter = 8,	Leaky Relu	$N_{RB} \times N_t \times N_t \times 8$	$8 \times 2 \times 3^2$	$(N_{RB} \times N_t \times N_t \times 8) \times (2 \times 3^2)$
Conv2D_2	Kernel = 3.	Leaky Relu	$N_{RB} \times N_r \times 8$	$8 \times 2 \times 3^2$	$(N_{RB} \times N_r \times 8) \times (2 \times 3^2)$
FCLayer_1(V)	Units = L_{ξ_V}	Relu	$L_{\xi_V} \times 1$	$[N_{RB} \times N_t \times N_t \times 8] \times L_{\xi_V}$	$2 \times [N_{RB} \times N_t \times N_t \times 8] \times L_{\xi_V}$
FCLayer_1(S)	Units = L_{ξ_S}	Relu	$L_{\xi_S} \times 1$	$[N_{RB} \times N_r \times 8] \times L_{\xi_S}$	$2 \times [N_{RB} \times N_r \times 8] \times L_{\xi_S}$

space complexity includes two parts: parameter quantity and output characteristic map. Meanwhile, the hyper-parameters and activation functions set for each layer are given in detail.

1) *Feature extraction module*: As is shown in Fig. 2, the feature extraction module includes two 2D convolution layers, two 3D convolution layers and two fully-connected layers. The time complexity of 2D and 3D convolution layers can be respectively expressed as,

$$\text{Time } \{Conv2D\} \sim O(S_M \cdot K^2 \cdot C_{in} \cdot C_{out}) \quad (19)$$

$$\text{Time } \{Conv3D\} \sim O(V_M \cdot K^3 \cdot C_{in} \cdot C_{out}) \quad (20)$$

where S_M, V_M are the area and volume of the output feature map of the convolution layer, respectively, K is the size of the convolution kernel, C_{in} denotes the number of input channels from the upper layer, and C_{out} represents the number of output channels. Considering the fully-connected layer, its time complexity can be formulated as,

$$\text{Time } \{fc\} \sim O(2 \times L_{in} \times L_{out}) \quad (21)$$

where L_{in} and L_{out} are the input and output tensor shape of fully-connected layer. After introducing the above time complexity, we continue to describe the space complexity. Eq. (22) to Eq. (24) are the space complexity expressions corresponding to 2D convolution layer, 3D convolution layer and fully-connected layer.

$$\text{Space } \{Conv2D\} \sim O(K^2 \times C_{in} \times C_{out}) \quad (22)$$

$$\text{Space } \{Conv3D\} \sim O(K^3 \times C_{in} \times C_{out}) \quad (23)$$

$$\text{Space } \{fc\} \sim O(L_{in} \times L_{out}) \quad (24)$$

Through the above analysis we can find that the convolution layer requires more FLOPs, but its space complexity is relatively low. On the contrary, the space complexity of fully-connected layer is larger, which means high demand for memory overhead. The hyper-parameters set in feature extraction module is listed in Tab. I, and the analysis of complexity is also included.

2) *Transcoding module*: Before the transcoding module, we want to briefly introduce the attention mechanism, which plays an important role in transcoding task. Attention mechanism [37] was proposed by the Bengio team and has been widely used in various fields of deep learning in recent years, such as in computer vision for capturing the receptive field on images, or for positioning in natural language processing (NLP) key token or feature. Fig. 3 shows the detailed tensor flow of the attention mechanism.

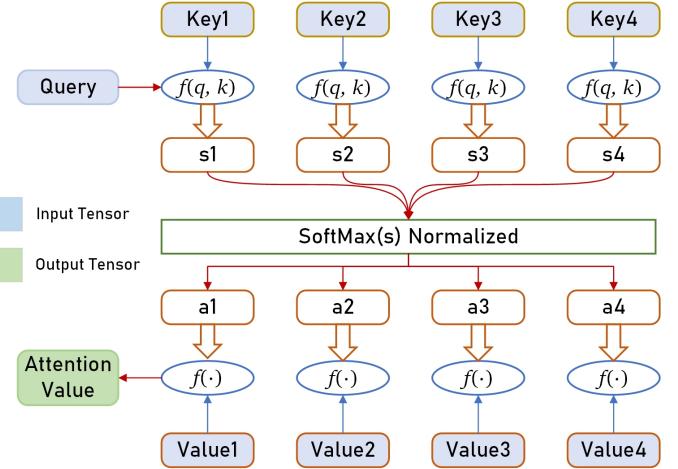


Fig. 3: Illustration of attention mechanism. The query (q), key (k), and value (v) are input tensors, and attention value (z) is output tensors.

First, attention distribution s between the input query vector q and the keyword vector k need to be known, which can be calculated via,

$$s_i = f(q, k_i) = \begin{cases} q^T k_i \\ q^T W k_i \\ [q^T k_i] / \sqrt{d} \\ v \cdot \tanh(Wq + U k_i) \end{cases} \quad (25)$$

where W , U and v are trainable weight coefficients in neural network, and d denotes the input dimension. Then, the attention distribution s is normalized to attention score a , which can be written as,

$$a_i = \text{softmax}[f(q, k_i)] = \frac{e^{f(q, k_i)}}{\sum_j e^{f(q, k_j)}} \quad (26)$$

Finally, the output z of attention mechanism is the result of weighted average of vector v , which is shown as,

$$z = \text{Attention}(q, k, v) = \sum_i a_i \cdot v_i \quad (27)$$

In short, the attention mechanism is actually designed to give larger weight to the parts that need attention, highlighting important information and ignoring other content. L. Chen *et al.* [38] explored the combination of CNN and attention mechanism, which gave excellent performance. Y. Cui *et al.* [39] applied the attention mechanism to the CSI feedback solution and demonstrated its performance improvement.

TABLE II: The hyper-parameters setting and analysis of parameters and FLOPs for transcoding module.

Layer name	Hyper-parameters	Activation	Output shape	Parameter size	FLOPs
Input(ξ_V)	–	–	$L_{\xi_V} \times 1$	–	–
Input(ξ_S)	–	–	$L_{\xi_S} \times 1$	–	–
Attention_res(\mathbf{V}, \mathbf{S})	Head_num = 2, Key_dim = 3.	–	$L_{\xi_V} \times 1$	$2 \times (L_{\xi_V}^2 + L_{\xi_S}^2)$	$8 \times (L_{\xi_V}^2 + L_{\xi_S}^2)$
Attention_res(\mathbf{V}, \mathbf{V})	–	–	$L_{\xi_V} \times 1$	$2 \times (L_{\xi_V}^2 + L_{\xi_V}^2)$	$8 \times (L_{\xi_V}^2 + L_{\xi_V}^2)$
FCLayer_codewords	Units = L_ε	Linear	$L_\varepsilon \times 1$	$(L_\varepsilon \times 1) \times (L_\varepsilon \times 1)$	$2 \times (L_\varepsilon \times 1) \times (L_\varepsilon \times 1)$

TABLE III: The hyper-parameters setting and analysis of parameters and FLOPs for decoder module.

Layer name	Hyper-parameters	Activation	Output shape	Parameter size	FLOPs
Input(ε)	–	–	$L_\varepsilon \times 1$	–	–
FCLayer_2(\mathbf{V})	Units = $[N_{RB} \cdot N_t \cdot N_t \cdot 2]$	Linear	$[N_{RB} \cdot N_t \cdot N_t \cdot 2]$	$L_\varepsilon \times [N_{RB} \cdot N_t \cdot N_t \cdot 2]$	$2 \times L_\varepsilon \times [N_{RB} \cdot N_t \cdot N_t \cdot 2]$
FCLayer_2(\mathbf{S})	Units = $[N_{RB} \cdot N_r \cdot 1]$	Linear	$[N_{RB} \cdot N_r]$	$L_\varepsilon \times [N_{RB} \cdot N_r]$	$2 \times L_\varepsilon \times [N_{RB} \cdot N_r]$
Conv3D_res	Filter = [2,8,2],	–	$N_{RB} \times N_t \times N_t \times 2$	$(2+8+2) \times 2 \times 3^2$	$(N_{RB} \times N_t \times N_t \times 12) \times (12 \times 3^2)$
Conv2D_res	Kernel = 3.	–	$N_{RB} \times N_t \times 2$	$(1+8+2) \times 2 \times 3^2$	$(N_{RB} \times N_r \times 11) \times (11 \times 3^2)$
Conv3D_3	Filter=2, Kernel=3	Tanh	$N_{RB} \times N_t \times N_t \times 2$	$2 \times 2 \times 3^2$	$(N_{RB} \times N_t \times N_t \times 2) \times (2 \times 3^2)$
Conv2D_3	Filter=2, Kernel=3	Linear	$N_{RB} \times N_r$	$1 \times 2 \times 3^2$	$(N_{RB} \times N_r \times 1) \times (2 \times 3^2)$

As is shown in Fig. 4, transcoding module is combined with two attention residual blocks and a fully-connected layer. The attention residual block receives two parallel input tensors \mathbf{X} and \mathbf{X}_{key} . If $\mathbf{X} = \mathbf{X}_{key}$, then we call it self-attention residual block, which pays more attention to self hidden information. If $\mathbf{X} \neq \mathbf{X}_{key}$, the attention residual block will embed the \mathbf{X}_{key} 's feature information in \mathbf{X} , which is called cross-attention residual block. First, we design a cross-attention residual block to embed the information of ξ_S into ξ_V . And then a self-attention residual block is followed to explore the self hidden features of ξ_V . The time complexity and space complexity of

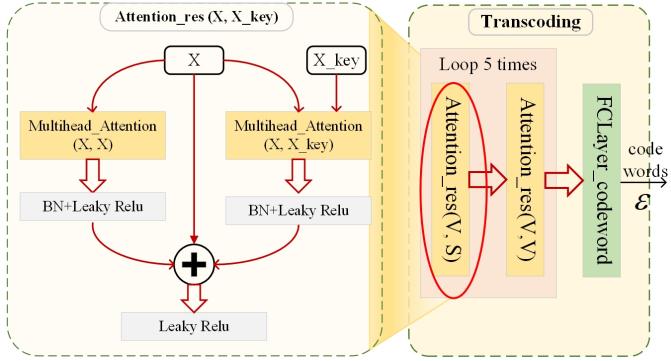


Fig. 4: Illustration of the transcoding module and attention residual block.

multi-head attention layer can be respectively formulated as,

$$Time \{att\} \sim O(L_{in}^2 \times d_{in}) \quad (28)$$

$$Space \{att\} \sim O(4 \times L_{in}^2 \times d_{in}) \quad (29)$$

where L_{in} is the length of input tensor, and d_{in} denotes the value of hyper-parameter *head_num*. The hyper-parameters set in transcoding module is listed in Tab. II, and the analysis of complexity is also included.

3) *Decoder module*: In this part we will discuss the decoder module in EMEVNet. The decoder will be deployed at the BS to reconstruct eigenmatrix $\hat{\mathbf{V}}$ and eigenvectors $\hat{\mathbf{S}}$. Fig. 5 shows the detailed decoder module and the convolution residual block applied in it. We design two different branches

to reconstruct $\hat{\mathbf{V}}$ and $\hat{\mathbf{S}}$ respectively. Two different fully-connected layers are utilized to extract the high-dimensional features of \mathbf{V} and \mathbf{S} from codewords ε . And then we design convolutional residual blocks to reconstruct $\hat{\mathbf{V}}$ and $\hat{\mathbf{S}}$. Since

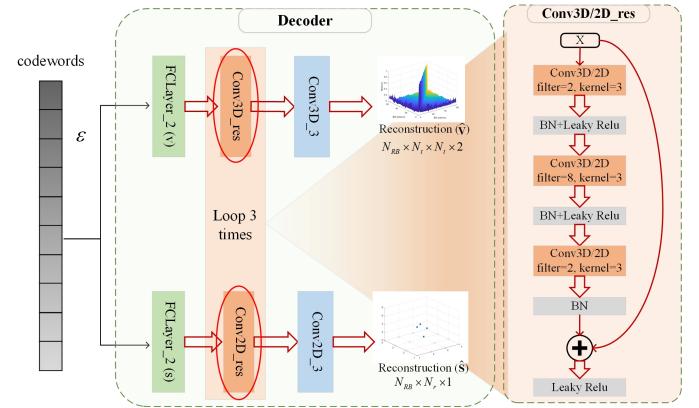


Fig. 5: Illustration of the decoder module and convolutional residual block.

this part depends on convolution layer and fully-connected layer, and their complexity analysis is mentioned above. The hyper-parameters set in transcoding module is listed in Tab. III, and the analysis of complexity is also included.

IV. SIMULATION RESULTS AND DISCUSSIONS

This section shows the simulation experiments of proposed EMEV feedback NN (EMEVNet) in detail, including simulation platform, datasets generation, parameter settings and performance evaluation. And then we give some analysis and discussion about the results, including feasibility analysis, superiority analysis and robustness analysis. Meanwhile, this section exhibits all numerical results corresponding to simulation experiments. The datasets used in this paper and the simulation codes can be found at Github⁶.

⁶Github link: <https://github.com/CodeDwan/EMEV-feedback>

A. Parameters Setting

1) *Simulation platform*: All the simulations and experiments are carried out on the workstation with CentOS 7.0. The workstation is equipped with two Intel(R) Xeon(R) Silver 4210R CPU and four Nvidia RTX 2080Ti GPU, it also has 256GB Random Access Memory (RAM).

2) *Datasets generation*: With the help of MATLAB 5G Toolbox and Communication Toolbox, we define a standard CDL channel object and carry on the link-level simulation. The dataset used in our experiments are extracted from the link-level simulator. Tab. IV shows the alternative parameter and default values in the data generator. Both UE and BS antennas follow uniform panel array (UPA) distribution.

TABLE IV: Simulation experiment parameter setting

Channel environment	NLOS			LOS	
	CDL-A	CDL-B	CDL-C	CDL-D	CDL-E
NRB	13				
Center frequency	28 GHz				
Subcarrier spacing	60 KHz				
UE speeds	{4.8, 24, 40, 60} km/h				
Delay spreads	129 ns	634 ns	634 ns	65 ns	65ns
BS antenna	UPA [8,8] = 64				
UE antenna	UPA [2,2] = 4				

We totally generate 60,000 data samples for each CDL channel using MATLAB. The 60,000 samples of each channel type are divided into 50,000 and 10,000. We utilize 50,000 samples \mathbb{D}_{sp} to train specific network \mathbb{N}_{sp} for each channel. For comparison, we mix 10,000 samples of five CDL channels to obtain \mathbb{D}_{mix} , and train a general network \mathbb{N}_{mix} . The operations of datasets and neural networks can be respectively expressed as,

$$\mathbb{N}_{sp}^* \leftarrow \mathbb{D}_{sp}^* = \left\{ \mathbf{H}_*^{(50k)} \right\} \quad (30)$$

$$\mathbb{N}_{mix} \leftarrow \mathbb{D}_{mix} = \left\{ \mathbf{H}_A^{(10k)}, \mathbf{H}_B^{(10k)}, \dots, \mathbf{H}_E^{(10k)} \right\} \quad (31)$$

where $\mathbf{H}_*^{(50k)}$ denotes the 50,000 samples of each channel environment represented by different subscripts, and $\mathbf{H}_A^{(10k)}$ means the 10,000 samples of CDL-A channel.

3) *Setting of compression ratios*: This paper explores the EMEV feedback solution. The UE needs to compress \mathbf{V} and \mathbf{S} first, and then feed back the compressed codewords ε to the BS. The length of codewords L_ε is related to the overhead of feedback, which will affect the spectral efficiency of communication system. And the L_ε is decided by compression ratio β_{CR} . Since the size of channel matrix \mathbf{H} , eigenmatrix \mathbf{V} and eigenvectors \mathbf{S} are different, we detailed define the setting of compression ratio β_{CR} in this part. For unity and comparison, almost all researches utilize the compression ratio of \mathbf{H} as the measurement standard. Hence, the length of codewords L_ε can be expressed as,

$$\begin{aligned} L_\varepsilon &= \frac{\Re(\mathbf{H}) + \Im(\mathbf{H})}{\beta_h} + 1 \\ &= \frac{N_{RB} \times N_r \times N_t \times 2}{\beta_h} + 1 \end{aligned} \quad (32)$$

where β_h is the compression ratio of \mathbf{H} . In order to guarantee the effectiveness of the comparative simulation experiments,

fixed L_ε is given in the follow discussion. To ensure the same L_ε , the compression ratio of EMEVNet can be defined as,

$$\begin{aligned} \beta_{emev} &\approx \frac{\Re(\mathbf{V}) + \Im(\mathbf{V}) + \Re(\mathbf{S})}{\Re(\mathbf{H}) + \Im(\mathbf{H})} \beta_h \\ &= \frac{N_{RB} \times (N_t \times N_t \times 2 + N_r)}{N_{RB} \times N_r \times N_t \times 2} \beta_h \end{aligned} \quad (33)$$

where β_{emev} represents the compression ratio of \mathbf{V}, \mathbf{S} . Referring to the datasets parameters settings in Tab. IV, we can assume $\beta_{emev} = 16\beta_h$ in this paper.

4) *Setting of NN training*: To ensure that the NN can converge to the best performance, we have tried many times for the NN training parameters, and finally defined as follows: Maximum epoch number $\tau = 500$; Initial learning rate $\eta = 1 \times 10^{-3}$; Early stop patience is 50 epoches. For the fairness evaluation, above NN training parameters are consistent in different experiments. The length of codewords and the corresponding compression ratio are set as follows:

- $L_\varepsilon = [416, 208, 104, 52, 26, 13, 6]$ means the length of codewords.
- $\beta_h = [16, 32, 64, 128, 256, 512, 1024]$ represents the compression ratio of CSI matrix \mathbf{H} .
- $\beta_{emev} = [256, 512, 1024, 2048, 4096, 8192, 16384]$ denotes the system compression ratio of EMEVNet.

5) *Performance evaluation*: The ultimate purpose of our architecture is to help BS reconstruct the eigenmatrix \mathbf{V} and eigenvector \mathbf{S} . Therefore, we utilize normalized mean square error (NMSE) and cosine similarity (ρ) to measure the reconstruction accuracy, which can be respectively defined as,

$$\begin{aligned} NMSE(\mathbf{V}, \hat{\mathbf{V}}) &= \mathbb{E} \left\{ \|\mathbf{V} - \hat{\mathbf{V}}\|_2^2 / \|\mathbf{V}\|_2^2 \right\} \\ &= 10 \log \left(\mathbb{E} \left\{ \|\mathbf{V} - \hat{\mathbf{V}}\|_2^2 / \|\mathbf{V}\|_2^2 \right\} \right) \end{aligned} \quad (34)$$

$$\begin{aligned} NMSE(\mathbf{S}, \hat{\mathbf{S}}) &= \mathbb{E} \left\{ \|\mathbf{S} - \hat{\mathbf{S}}\|_2^2 / \|\mathbf{S}\|_2^2 \right\} \\ &= 10 \log \left(\mathbb{E} \left\{ \|\mathbf{S} - \hat{\mathbf{S}}\|_2^2 / \|\mathbf{S}\|_2^2 \right\} \right) \end{aligned} \quad (35)$$

$$\rho(\mathbf{V}, \hat{\mathbf{V}}) = \mathbb{E} \left\{ \frac{\langle \mathbf{V}^*, \hat{\mathbf{V}} \rangle}{\|\mathbf{V}\|_2 \|\hat{\mathbf{V}}\|_2} \right\} \quad (36)$$

$$\rho(\mathbf{S}, \hat{\mathbf{S}}) = \mathbb{E} \left\{ \frac{\langle \mathbf{S}^*, \hat{\mathbf{S}} \rangle}{\|\mathbf{S}\|_2 \|\hat{\mathbf{S}}\|_2} \right\} \quad (37)$$

where $\langle a, b \rangle$ is the scalar product in Euclidean space. Generally, NMSE is converted to logarithmic domain and the smaller value represents the better performance, e.g. $NMSE = -20$ dB shows better performance than -10 dB. The range of cosine similarity is $\rho \in [-1, 1]$, which can measure the similarity between reconstructed and original matrix. ρ close to 1 indicates better reconstruction performance.

B. Feasibility Analysis

This subsection will show some simulation results and discuss the feasibility of our proposed architecture. In order to verify the feasibility of proposed EMEVNet, this part utilize

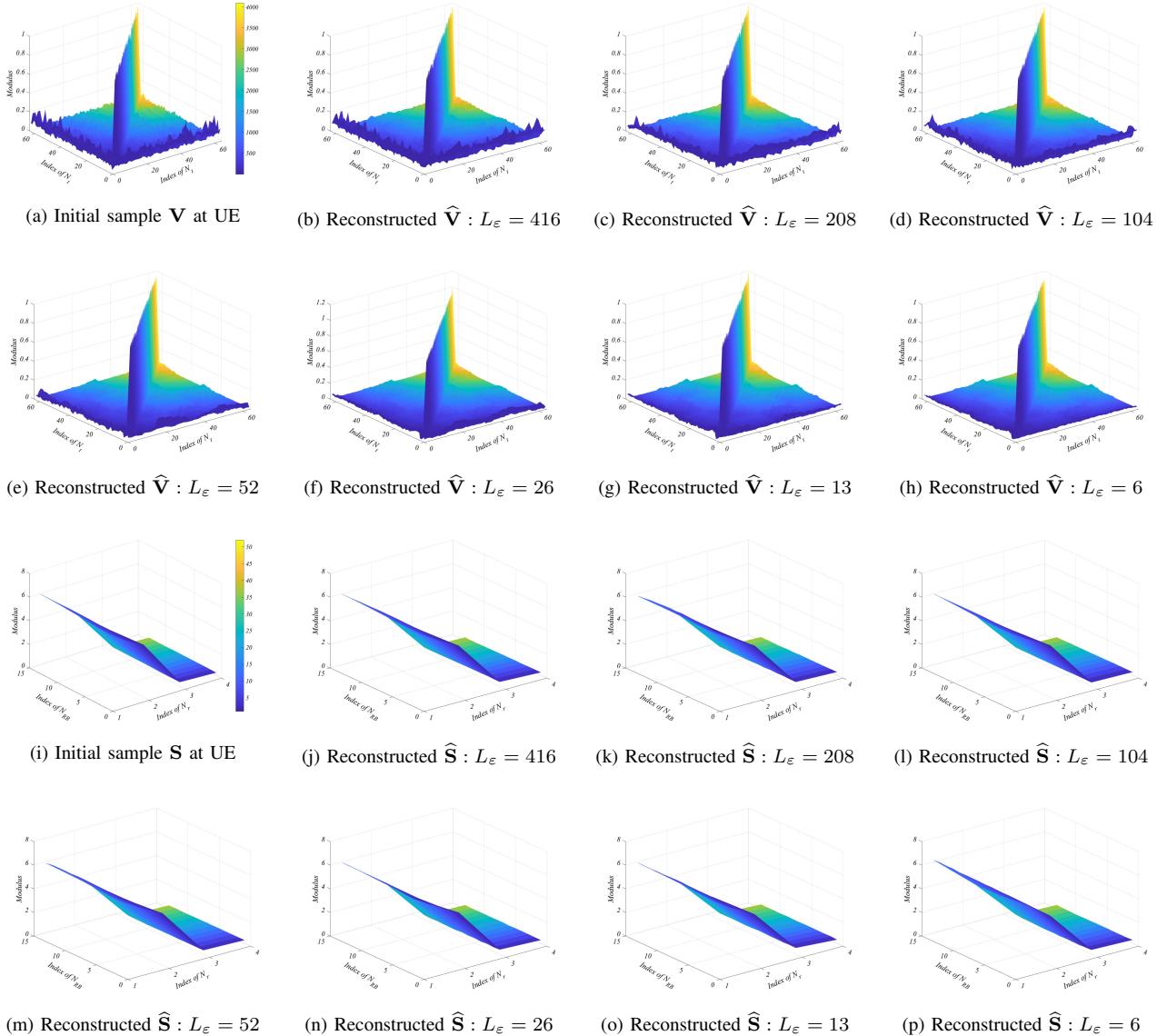


Fig. 6: Illustration of partial visualization feasibility experimental results. Fig. 6a and Fig. 6i show the perfect eigenmatrix \mathbf{V} and eigenvector \mathbf{S} at the UE. Fig. 6b to Fig. 6h describe the reconstructed $\widehat{\mathbf{V}}$ from $L_\varepsilon = 416$ to $L_\varepsilon = 6$. Fig. 6j to Fig. 6p show the reconstructed eigenvector $\widehat{\mathbf{S}}$ with the same L_ε .

50,000 samples of CDL-A channel, i.e. $\mathbb{D}_{sp}^A = \{\mathbf{H}_A^{(50k)}\}$, as simulation datasets. We set the training, validation and testing datasets obeying the ratio of 70 : 15 : 15. Before training EMEVNet, the datasets \mathbb{D}_{sp}^A need to carry out SVD transformation according to **Algorithm 1**. And the generated \mathbf{V}, \mathbf{S} are the input of EMEVNet. After NN training stage according to **Algorithm 2**, we can obtain the trained \mathbb{N}_{sp}^A , which is the specific EMEVNet for CDL-A channel. In this part, we have trained and tested seven different compression ratios set as **Section IV-A4**.

The experimental results can be found in Fig. 6. Considering that $\mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}$ is a 3D complex-valued matrix, so we split one RB for more intuitive exhibition. The program randomly selects the third resource block, i.e. $\mathbf{V}(3) \in \mathbb{C}^{N_t \times N_t}$. Then, the Euclidean norm of $\mathbf{V}(3)$ is applied for

convenient plotting, which can be interpreted as the power distribution. Fig. 6a shows the initial \mathbf{V} sample at UE and Fig. 6b to Fig. 6h show the reconstructed $\widehat{\mathbf{V}}$ at BS with different compression ratio from $L_\varepsilon = 416$ to $L_\varepsilon = 6$. As for eigenvector $\mathbf{S} \in \mathbb{R}^{N_{RB} \times N_r}$, it is a 2D real-valued matrix. We directly plot the figure without any processing. Fig. 6i shows the initial \mathbf{S} at UE and Fig. 6j to Fig. 6p show the reconstructed $\widehat{\mathbf{S}}$ at BS with the same settings as mentioned above. After many experiments, the shape of \mathbf{V} of different RBs is almost the same. They all present diagonal power distribution, and the edges are serrated. Moreover, the power distribution of \mathbf{V} is a symmetrical image.

As shown in Fig. 6, with the reduction of feedback codewords L_ε , the reconstructed $\widehat{\mathbf{V}}$ by the BS will lose more information. This loss is mainly reflected in the sawtooth

energy at the edge of eigenmatrix, while the loss of power principal component (diagonal distribution) is limited. When the length of codewords comes to $L_{\varepsilon} = 26$, i.e. $\beta_h = 256$ and $\beta_{emev} = 4096$, the reconstructed $\hat{\mathbf{V}}$ losses almost all edge information. It can be seen from the figure that when $L_{\varepsilon} \leq 26$, the edge power of $\hat{\mathbf{V}}$ cannot be reconstructed. However, we can ensure that the distribution of $\hat{\mathbf{V}}$ remains unchanged and the diagonal power is hardly affected in any case. As for $\hat{\mathbf{S}}$ at the BS, it can be well reconstructed under almost any L_{ε} . The detailed numerical results of \mathbb{N}_{sp}^A , i.e. $NMSE, \rho$, will be shown and discussed in **Section IV-E**. From the exhibition and analysis of this subsection, we vividly proved the feasibility of EMEVNet by visualizing some experimental results.

C. Superiority Analysis

In this subsection, we hope to prove the superior performance of the proposed architecture. As is mentioned in the previous section, we assign a channel identification module before encoding \mathbf{V} and \mathbf{S} . Therefore, this part compares the performance of specific networks \mathbb{N}_{sp}^* trained with large datasets \mathbb{D}_{sp}^* and general network \mathbb{N}_{mix} trained with mixed datasets \mathbb{D}_{mix} . In order to completely verify the superiority of the proposed architecture, we have checked five CDL channels defined by 3GPP. The testing objects are $\{\mathbb{N}_{sp}^A, \mathbb{N}_{sp}^B, \mathbb{N}_{sp}^C, \mathbb{N}_{sp}^D, \mathbb{N}_{sp}^E\}$, and the baseline is set as $\{\mathbb{N}_{mix}\}$. For each experiment, we give four evaluation indexes defined in **Section IV-A5**, including $NMSE(\mathbf{V}, \hat{\mathbf{V}})$, $NMSE(\mathbf{S}, \hat{\mathbf{S}})$, $\rho(\mathbf{V}, \hat{\mathbf{V}})$ and $\rho(\mathbf{S}, \hat{\mathbf{S}})$.

The experimental results can be seen in Fig. 7. Each channel type has been tested and verified by 7 different compression ratios mentioned in **Section IV-A4**. The horizontally adjacent figures respectively show the reconstruction performance curves of $\hat{\mathbf{V}}$ and $\hat{\mathbf{S}}$ under the same channel, e.g. Fig. 7a and Fig. 7b show the performance under CDL-A channel environment. In addition, the vertically adjacent figures show the performance of the same feedback information with different channel types, e.g. Fig. 7a and Fig. 7c show the performance with different channels. Each figure has two y-axes of different scales, corresponding to $NMSE(dB)$ and ρ respectively. In addition, there are four performance curves in each figure, among of which the solid blue lines show $NMSE$ performance, the dotted red lines represent ρ performance, the circle marked lines represent specific network \mathbb{N}_{sp}^* , and the triangle marked lines represent general network \mathbb{N}_{mix} .

As can be seen from Fig. 7, the two blue solid lines of each figure show an upward trend, while the two red dotted lines show a downward trend. This result is the same as the analysis in **Section IV-A5**. The performance of reconstruction is getting worse with the decrease of L_{ε} , i.e., $NMSE(dB)$ tends to be larger and ρ deviates from 1. What's more, all the blue lines with circle mark are lower than the triangle mark, and all the red lines with circle mark are higher than the triangle mark. These results mean that the performance of specific network \mathbb{N}_{sp}^* is better than general network \mathbb{N}_{mix} . It can also be found that two lines with the same color are almost parallel, which shows that the performance improvement of \mathbb{N}_{sp}^* is relatively stable compared with \mathbb{N}_{mix} , and is less affected by L_{ε} . In

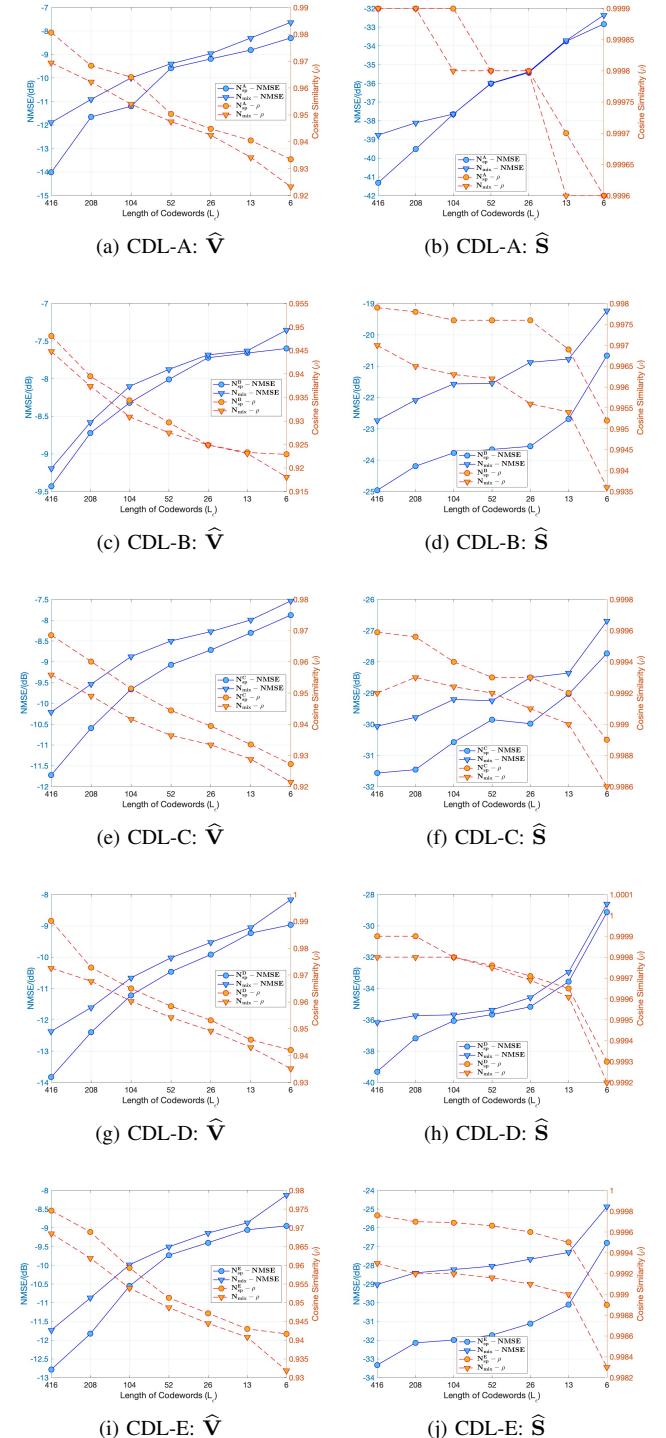


Fig. 7: Illustration of superiority experimental results. The x-axis represents different lengths of codewords. The blue and red y-axes denote $NMSE$ (dB) and cosine similarity (ρ).

summary, we can conclude that the channel identification module is necessary, and the performance of feedback and reconstruction can be improved by selecting a specific network \mathbb{N}_{sp}^* .

D. Robustness Analysis

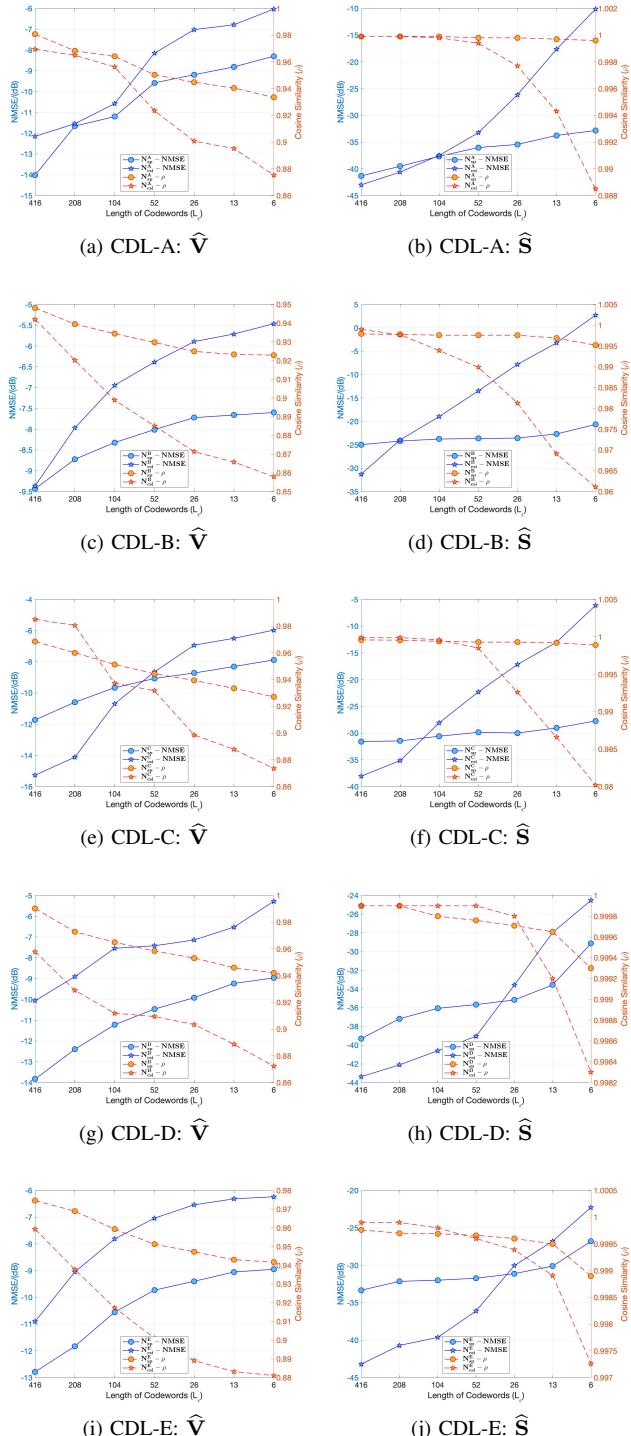


Fig. 8: Illustration of robustness experimental results. The x-axis represents different lengths of codewords. The blue and red y-axes are NMSE (dB) and cosine similarity (ρ).

This subsection proves the robustness of proposed architecture by comparing EMEVNet with the classical CSI feedback scheme CsiNet [11]. In order to narrate conveniently, \mathbb{N}_{csi}^* is defined as the classical CsiNet framework. We distinguish different channels by superscripts. e.g. \mathbb{N}_{csi}^A represents CsiNet based on CDL-A channel. For fairness, specific datasets with 50,000 samples are used in this subsection, and \mathbb{N}_{sp}^* and \mathbb{N}_{csi}^* are trained respectively. Similarly, the experiments in this part also verify 5 channel types and 7 different compression ratios set as **Section IV-A4**. It should be noted that the comparison experiment is based on the same length of codewords, that is, the performance is compared with the same feedback overhead. Considering that CsiNet encodes and feedbacks the channel matrix \mathbf{H} , we keep the same feedback overhead and carry out SVD transform on the decoded $\tilde{\mathbf{H}}$ at the BS. The testing objects are $\{\mathbb{N}_{sp}^A, \mathbb{N}_{sp}^B, \mathbb{N}_{sp}^C, \mathbb{N}_{sp}^D, \mathbb{N}_{sp}^E\}$, and the baselines are set as $\{\mathbb{N}_{csi}^A, \mathbb{N}_{csi}^B, \mathbb{N}_{csi}^C, \mathbb{N}_{csi}^D, \mathbb{N}_{csi}^E\}$.

The experimental results can be seen in Fig. 8. The arrangement of different subfigures is the same as that in Fig. 7. Each figure has two y-axes of different scales, corresponding to $NMSE(dB)$ and ρ respectively. In addition, there are four performance curves in each figure, and the solid blue lines show $NMSE$ performance, the dotted red lines represent ρ performance, the circle marked lines represent EMEVNet \mathbb{N}_{sp}^* , and the pentagram marked lines represent the baseline, i.e. CsiNet \mathbb{N}_{csi}^* . The visualization results show that the lines marked by the pentagram have larger gradients. This phenomenon is more obvious in the performance curve of reconstructed $\hat{\mathbf{S}}$ at the BS. This can be explained as the baseline \mathbb{N}_{csi}^* is greatly affected by the length of codewords L_e . As for eigenmatrix $\hat{\mathbf{V}}$, except Fig. 8e shows that the baseline is better than EMEVNet with long L_e , the other results all show the performance of EMEVNet is better with all L_e . As for eigenvector $\hat{\mathbf{S}}$, we can find baseline has better performance with long L_e in all channels. For NLOS channels (CDL-A, CDL-B and CDL-C), when the length of codewords decreases to $L_e = 104$, the performance of EMEVNet becomes better than that of baseline. While discussing LOS channels (CDL-D and CDL-E), this threshold is relaxed to $L_e = 26$. After analysis, we believe that this is because there exists line of sight fading path with strong energy in LOS channels, which cause improvement of baseline. At the same time, we find that the performance of baseline attenuates seriously in the case of limited L_e . However, the performance of EMEVNet is relatively stable. Especially in the case of limit L_e , EMEVNet shows much better performance than baseline. To sum up, the proposed EMEVNet has good robustness and adaptability. It can be applied to larger system compression rate, that is, it occupies smaller feedback codewords, which can effectively reduce the feedback overhead and improve the spectrum utilization.

E. Numerical Results

This subsection shows the numerical results of all simulation experiments in **Sections IV-B**, **IV-C** and **IV-D**. All numerical results can be found in Tab. V. The \mathbb{N}_{sp}^* and \mathbb{N}_{csi}^* in the table correspond to the special EMEVNet

TABLE V: The numerical results of simulation experiments carried out and discussed in section IV.

			CDL-A		CDL-B		CDL-C		CDL-D		CDL-E	
			NMSE (dB)	ρ								
$L_\varepsilon = 416$ $\beta_h = 16$ $\beta_{emev} = 256$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-14.018	0.9807	-9.432	0.9481	-11.721	0.9686	-13.830	0.9902	-12.784	0.9746
		$\hat{\mathbf{S}}$	-41.318	0.9999	-24.964	0.9979	-31.566	0.9996	-39.307	0.9999	-33.332	0.9998
		$\hat{\mathbf{V}}$	-12.148	0.9695	-9.362	0.9421	-15.264	0.9851	-10.064	0.9579	-10.902	0.9593
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-42.999	0.9999	-31.292	0.9990	-38.057	0.9999	-43.368	0.9999	-43.211	0.9999
		$\hat{\mathbf{V}}$	-11.895	0.9694	-9.196	0.9448	-10.206	0.9558	-12.367	0.9726	-11.733	0.9685
		$\hat{\mathbf{S}}$	-38.764	0.9999	-22.736	0.9970	-30.065	0.9993	-36.161	0.9998	-29.025	0.9993
$L_\varepsilon = 208$ $\beta_h = 32$ $\beta_{emev} = 512$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-11.657	0.9683	-8.722	0.9395	-10.592	0.9601	-12.398	0.9728	-11.828	0.9689
		$\hat{\mathbf{S}}$	-39.512	0.9999	-24.193	0.9978	-31.456	0.9996	-37.178	0.9999	-32.143	0.9997
		$\hat{\mathbf{V}}$	-11.545	0.9649	-7.970	0.9202	-14.113	0.9806	-8.905	0.9290	-9.048	0.9377
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-40.598	0.9999	-23.947	0.9977	-35.133	0.9999	-42.110	0.9999	-40.704	0.9999
		$\hat{\mathbf{V}}$	-10.909	0.9623	-8.580	0.9374	-9.536	0.9491	-11.607	0.9677	-10.870	0.9619
		$\hat{\mathbf{S}}$	-38.111	0.9999	-22.082	0.9965	-29.778	0.9992	-35.734	0.9998	-28.401	0.9992
$L_\varepsilon = 104$ $\beta_h = 64$ $\beta_{emev} = 1024$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-11.202	0.9641	-8.321	0.9344	-9.660	0.9514	-11.217	0.9650	-10.552	0.9593
		$\hat{\mathbf{S}}$	-37.645	0.9999	-23.765	0.9976	-30.574	0.9994	-36.067	0.9998	-31.993	0.9997
		$\hat{\mathbf{V}}$	-10.58	0.9562	-6.943	0.8989	-10.688	0.9373	-7.5421	0.9119	-7.816	0.9173
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-37.476	0.9998	-18.961	0.9939	-28.065	0.9996	-40.592	0.9999	-39.626	0.9998
		$\hat{\mathbf{V}}$	-9.998	0.9540	-8.101	0.9309	-8.874	0.9416	-10.661	0.9603	-9.983	0.9539
		$\hat{\mathbf{S}}$	-37.643	0.9998	-21.568	0.9963	-29.206	0.9992	-35.680	0.9998	-28.225	0.9992
$L_\varepsilon = 52$ $\beta_h = 128$ $\beta_{emev} = 2048$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-9.581	0.9504	-8.011	0.9297	-9.070	0.9444	-10.465	0.9584	-9.7297	0.9513
		$\hat{\mathbf{S}}$	-36.016	0.9998	-23.657	0.9976	-29.855	0.9993	-35.663	0.9998	-31.729	0.9997
		$\hat{\mathbf{V}}$	-8.152	0.9235	-6.386	0.8851	-8.646	0.9317	-7.422	0.9094	-7.038	0.9011
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-33.241	0.9994	-13.486	0.9899	-22.293	0.9985	-39.054	0.9999	-36.104	0.9996
		$\hat{\mathbf{V}}$	-9.394	0.9475	-7.877	0.9275	-8.499	0.9364	-10.021	0.9542	-9.507	0.9487
		$\hat{\mathbf{S}}$	-36.001	0.9998	-21.547	0.9962	-29.249	0.9992	-35.369	0.9997	-28.046	0.9992
$L_\varepsilon = 26$ $\beta_h = 256$ $\beta_{emev} = 4096$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-9.187	0.9448	-7.721	0.9249	-8.717	0.9394	-9.917	0.9532	-9.397	0.9472
		$\hat{\mathbf{S}}$	-35.434	0.9997	-23.559	0.9976	-29.982	0.9993	-35.167	0.9997	-31.112	0.9996
		$\hat{\mathbf{V}}$	-7.020	0.9007	-5.893	0.8713	-6.931	0.8986	-7.143	0.9034	-6.539	0.8891
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-26.173	0.9977	-7.831	0.9813	-17.173	0.9926	-33.582	0.9998	-30.038	0.9994
		$\hat{\mathbf{V}}$	-8.963	0.9425	-7.683	0.9249	-8.274	0.9334	-9.530	0.9492	-9.139	0.9445
		$\hat{\mathbf{S}}$	-35.393	0.9998	-20.877	0.9956	-28.506	0.9991	-34.578	0.9997	-27.672	0.9991
$L_\varepsilon = 13$ $\beta_h = 512$ $\beta_{emev} = 8192$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-8.812	0.9405	-7.659	0.9233	-8.303	0.9335	-9.232	0.9459	-9.051	0.9430
		$\hat{\mathbf{S}}$	-33.754	0.9997	-22.682	0.9969	-29.037	0.9992	-33.561	0.9996	-30.105	0.9995
		$\hat{\mathbf{V}}$	-6.786	0.8952	-5.711	0.8658	-6.492	0.8879	-6.524	0.8887	-6.315	0.8832
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-17.651	0.9943	-3.233	0.9691	-13.018	0.9866	-27.956	0.9992	-26.774	0.9989
		$\hat{\mathbf{V}}$	-8.356	0.9342	-7.628	0.9231	-7.994	0.9288	-9.055	0.9431	-8.866	0.9408
		$\hat{\mathbf{S}}$	-33.717	0.9996	-20.771	0.9954	-28.358	0.9990	-32.944	0.9996	-27.311	0.9990
$L_\varepsilon = 6$ $\beta_h = 1024$ $\beta_{emev} = 16384$	\mathbb{N}_{sp}^* (proposed) \mathbb{N}_{csi}^*	$\hat{\mathbf{V}}$	-8.298	0.9335	-7.599	0.9229	-7.879	0.9272	-8.969	0.9421	-8.948	0.9417
		$\hat{\mathbf{S}}$	-32.833	0.9996	-20.660	0.9952	-27.729	0.9989	-29.132	0.9993	-26.799	0.9989
		$\hat{\mathbf{V}}$	-6.033	0.8754	-5.465	0.8579	-5.971	0.8735	-5.283	0.8723	-6.238	0.8811
	\mathbb{N}_{mix}^*	$\hat{\mathbf{S}}$	-10.133	0.9885	2.702	0.9611	-6.120	0.9802	-24.523	0.9983	-22.261	0.9973
		$\hat{\mathbf{V}}$	-7.633	0.9233	-7.355	0.9181	-7.538	0.9215	-8.165	0.9352	-8.125	0.9319
		$\hat{\mathbf{S}}$	-32.357	0.9997	-19.231	0.9936	-26.690	0.9986	-28.593	0.9992	-24.871	0.9983

and CsiNet trained by large datasets, respectively. And \mathbb{N}_{mix} represents general EMEVNet obtained from mixed datasets. The horizontal comparison is to verify the performance of different channel environments, and the vertical comparison is to verify the impact of different length of codewords. Meanwhile, from the numerical analysis, the reconstruction performance of $\hat{\mathbf{V}}$ is worse than that of $\hat{\mathbf{S}}$, which is also consistent with the results in **Section IV-B**. From the analysis of numerical results, we can conclude that $\hat{\mathbf{S}}$ is almost perfectly reconstructed, while the performance of $\hat{\mathbf{V}}$ is worse with the decrease of L_ε .

V. CONCLUSION

In this paper, a novel channel feedback architecture for mmWave FDD systems was proposed. The key idea of our

architecture was to feed back useful channel information to the BS, instead of the complete CSI matrix. This paper discussed the beamforming technology based on SVD transformation, and the core design of the architecture was to feed back eigenmatrix and eigenvector. The major technical methods used in this paper include: using SVD transform to extract the effective information; utilizing attention mechanism to design a dual channel auto-encoder; deploying a channel identification NN at the UE to switch the appropriate specific EMEVNet. We considered and verified five common CDL channel environments and seven incremental system compression ratios. And all simulations were carried out in mmWave system. First, we demonstrated the feasibility of our proposed architecture by visualizing some experimental results. Then, we designed two comparison experiments to

prove the superiority and robustness of proposed architecture, respectively. Finally, we showed the numerical results of all simulation experiments to further prove our analysis. This paper provided a new solution to solve the problem that the BS hardly acquires downlink CSI in FDD wireless communication system. Through extracting and feeding back useful information for BS, the intelligent communication system is able to further improve the performance and reduce overhead.

REFERENCES

- [1] C. Qi, P. Dong, W. Ma, H. Zhang, Z. Zhang, and G. Y. Li, "Acquisition of channel state information for mmwave massive MIMO: Traditional and machine learning-based approaches," *Science China Information Sciences*, vol. 64, no. 8, pp. 1–16, June 2021.
- [2] W. Tong and G. Y. Li, "Nine challenges in artificial intelligence and wireless communications for 6G," Sept. 2021. [Online]. Available: <https://arxiv.org/abs/2109.11320>
- [3] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security, and intelligence," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126–132, Mar. 2020.
- [4] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI models for green communications towards 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 210–247, Mar. 2022.
- [5] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 53–87, Mar. 2022.
- [6] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, Aug. 2019.
- [7] C. Huang, S. Hu, G. C. Alexandropoulos, A. Zappone, C. Yuen, R. Zhang, M. D. Renzo, and M. Debbah, "Holographic MIMO surfaces for 6G wireless networks: Opportunities, challenges, and trends," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 118–125, Oct. 2020.
- [8] A. Froymog, G. Y. Li, and et al., "Ultra-low power wake-up radio for 5G IoT," *IEEE Communications Magazine*, vol. 57, no. 3, pp. 111–117, Mar. 2019.
- [9] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 93–99, Apr. 2019.
- [10] Z. Qin, G. Y. Li, and H. Ye, "Federated learning and wireless communications," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 134–140, Oct. 2021.
- [11] 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, Oct. 2018.
- [12] J. Guo, J. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Compression and acceleration of neural networks for communications," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 110–117, Aug. 2020.
- [13] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2827–2840, Apr. 2020.
- [14] 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, Apr. 2019.
- [15] Y. Sun, W. Xu, L. Fan, G. Y. Li, and G. K. Karagiannidis, "Ancinet: An efficient deep learning approach for feedback compression of estimated CSI in massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 9, no. 12, pp. 2192–2196, Dec. 2020.
- [16] Y. Sun, W. Xu, L. Liang, N. Wang, G. Y. Li, and X. You, "A lightweight deep network for efficient CSI feedback in massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1840–1844, Aug. 2021.
- [17] J. Zeng, J. Sun, G. Gui, B. Adeeb, T. Ohtsuki, H. Gacanin, and H. Sari, "Downlink CSI feedback algorithm with deep transfer learning for FDD massive MIMO systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 4, pp. 1253–1265, Dec. 2021.
- [18] X. Ma, Z. Gao, F. Gao, and M. Di Renzo, "Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid MIMO systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2388–2406, Aug. 2021.
- [19] M. Chen, J. Guo, C.-K. Wen, S. Jin, G. Y. Li, and A. Yang, "Deep learning-based implicit CSI feedback in massive MIMO," *IEEE Transactions on Communications*, Feb. 2021.
- [20] Z. Zhong, L. Fan, and S. Ge, "FDD massive MIMO uplink and downlink channel reciprocity properties: Full or partial reciprocity?" in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020, pp. 1–5.
- [21] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1994–1998, Nov. 2019.
- [22] Y. Yang, F. Gao, Z. Zhong, B. Ai, and A. Alkhateeb, "Deep transfer learning-based downlink channel prediction for FDD massive MIMO systems," *IEEE Transactions on Communications*, vol. 68, no. 12, pp. 7485–7497, Dec. 2020.
- [23] M. S. Safari, V. Pourahmadi, and S. Sodagari, "Deep UL2DL: Data-driven channel knowledge transfer from uplink to downlink," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 29–44, Dec. 2019.
- [24] Y. Zhang, B. Adeeb, H. Gacanin, and F. Adachi, "CV-3DCNN: Complex-valued deep learning for CSI prediction in FDD massive MIMO systems," *IEEE Wireless Communications Letters*, vol. 10, no. 2, pp. 266–270, Feb. 2021.
- [25] Y. Yang, F. Gao, C. Xing, J. An, and A. Alkhateeb, "Deep multimodal learning: Merging sensory data for massive MIMO channel prediction," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1885–1898, July 2021.
- [26] J. Wang, T. Ohtsuki, B. Adeeb, H. Gacanin, and H. Sari, "Compressive sampled CSI feedback method based on deep learning for FDD massive MIMO systems," *IEEE Transactions on Communications*, vol. 69, no. 9, pp. 5873–5885, Sept. 2021.
- [27] W. Liu, W. Tian, H. Xiao, S. Jin, X. Liu, and J. Shen, "EVCsiNet: Eigenvector-based CSI feedback under 3GPP link-level channels," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2688–2692, Dec. 2021.
- [28] F. Gao, B. Lin, C. Bian, T. Zhou, J. Qian, and H. Wang, "Fusionnet: Enhanced beam prediction for mmwave communications using sub-6 GHz channel and a few pilots," *IEEE Transactions on Communications*, vol. 69, no. 12, pp. 8488–8500, Dec. 2021.
- [29] J. Guo, C.-K. Wen, and S. Jin, "Deep learning-based CSI feedback for beamforming in single- and multi-cell massive MIMO systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 1872–1884, July 2021.
- [30] Z. Liu, Y. Yang, F. Gao, T. Zhou, and H. Ma, "Deep unsupervised learning for joint antenna selection and hybrid beamforming," *IEEE Transactions on Communications*, vol. 70, no. 3, pp. 1697–1710, Mar. 2022.
- [31] Z. Gao, M. Wu, C. Hu, F. Gao, G. Wen, D. Zheng, and J. Zhang, "Data-driven deep learning based hybrid beamforming for aerial massive MIMO-OFDM systems with implicit CSI," Feb. 2022. [Online]. Available: <https://arxiv.org/abs/2201.06778>
- [32] 3GPP, "Study 3D channel model for LTE (relase 12)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.873, 1 2018, version 14.1.0.
- [33] ———, "Study on channel model for frequencies from 0.5 to 100 GHz (relase 16)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.901, 12 2019, version 16.1.0.
- [34] Y. Zhang, J. Sun, H. Gacanin, and F. Adachi, "A novel channel identification architecture for mmwave systems based on eigen features," Apr. 2022. [Online]. Available: <https://arxiv.org/abs/2204.05052>
- [35] "NLOS and LOS of the 28 GHz bands millimeter-wave in 5G cellular networks."
- [36] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp. 5353–5360.
- [37] A. Vaswani and et al., "Attention is all you need," *Advances in neural information processing systems*, vol. 30, Dec. 2017.
- [38] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 6298–6306.
- [39] Y. Cui, A. Guo, and C. Song, "TransNet: Full attention network for CSI feedback in FDD massive MIMO system," *IEEE Wireless Communications Letters*, pp. 1–1, Feb. 2022.