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

Abstract—The potential advantages of intelligent wireless communications with millimeter wave (mmWave) 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) 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 (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 transmit the compressed codewords to the BS. And the BS obtains the accurate downlink CSI after decoding the codewords. J. Guo et al. [12] 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. [17] 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. [13], [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)\(^\dagger\) 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,

\[
y = Hx + n
\]

where \(H \in \mathbb{C}^{N_{RB} \times N_r \times N_t}\) is the downlink CSI between the BS and the UE, \(n\) 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,

\[
H = \sum_{n=1}^{N} \sum_{m=1}^{M} \sqrt{P_{n,m}} \cdot e^{j2\pi \nu(n,m)} \alpha(\theta(n,m))
\]

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, \(\alpha(\theta(n,m))\) 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, \(\nu(n,m)\) stands for the speed and can be understood as Doppler shift parameter. Then, the the steering vector \(\alpha(\theta(n,m)) \in \mathbb{C}^{N_t \times 1}\) can be formulated as,

\[
\alpha(\theta) = \left[1, e^{-j2\pi \frac{d}{\lambda} \sin(\theta)}, \ldots, e^{-j2\pi \frac{d(N_r-1)d}{\lambda} \sin(\theta)}\right]
\]

where \(d\) and \(\lambda\) 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 \(\Pr_{LOS} = 1\), else the \(\Pr_{LOS}\) can be calculated via

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

where the \(C(h_{UT})\) can be found in [3], and the \(h_{UT}\) denotes the antenna height for the UE.

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

\(^\dagger\) We adopt the ULA model here for simpler illustration, nevertheless, the proposed approach does not restrict to the specific 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 $H \in \mathbb{C}^{N \times N}$. For simplicity of description, we discuss only one RB here\footnote{This assumption is only for the brief illustration of SVD, and this solution is also applicable to OFDM systems.}, i.e. $RB = 1$ and $H \in \mathbb{C}^{N_r \times N_t}$. First, the CSI matrix $H$ should carry on SVD transformation as

$$H = U \cdot \Sigma \cdot V^\ast$$

where $U \in \mathbb{C}^{N_r \times N_r}$ and $V \in \mathbb{C}^{N_t \times N_t}$ are the left-singular and the right-singular matrices respectively. What is more, $UU^\ast = I_{N_r}$, $VV^\ast = I_{N_t}$\footnote{Both $U$ and $V$ will be called as eigenmatrix in the follows.}. Note that $\Sigma = (\Lambda, 0)$ and $\Lambda$ can be expressed as follows:

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

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

$$y = Hx + n$$

where $y$ is the received data flow and $n$ denotes the noise vector. The channel matrix $H$ can be expressed by $[13]$, and we can obtain

$$y = U\Sigma V^\ast V \cdot x + n$$

Finally, the receiver will use $U^\ast$ for receiver combining, which can be expressed as

$$U^\ast y = U^\ast (U\Sigma V^\ast V \cdot x + n) = \Sigma x + U^\ast n$$

The noise component in $[10]$ will be filtered out by the receiver. Therefore, the receiver can recover the data flow $x$ by $\Sigma$.

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

C. Problem Formulation

As analyzed above, the CSI matrix $H$ 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 $V$ (right-singular matrix) and eigenvector $s$. What’s more, the eigenmatrix $V$ is unitary matrix which is symmetric and easily compressible, and the eigenvector $s$ is a simple real-value vector. Therefore, this paper proposes to jointly compress the unitary matrix $V$ and the corresponding eigenvector $s$, 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 $V \in \mathbb{C}^{N_r \times N_t}$, and the eigenvector $s \in \mathbb{R}^{N_r \times N_t}$. The UE needs to deploy an encoder to jointly encode $V$ and $s$, which can be formulated as,

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

where $\varepsilon$ represents the codewords encoded by the UE, $\Theta_{en}$ 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 $V$ and $S$ respectively, and match a suitable mapping function to convert them into codewords. When received the codewords $\varepsilon$, the BS needs to switch the corresponding decoder to interpret and obtain the required $V$ and $S$. The decoder can be expressed as,

$$\hat{V}, \hat{S} = f_{de}(f_{en}(V, S, \Theta_{en}), \Theta_{de})$$

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

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

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

where $\mathbb{E}(\cdot)$ stands for mathematical expectation, $\| \cdot \|_2$ 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,

$$\left( \Theta_{en}^\ast, \Theta_{de}^\ast \right) = \operatorname{arg} \min_{\Theta_{en}, \Theta_{de}} \mathbb{E} \left[ \Gamma \left( \| V - \hat{V} \|_2^2, \| S - \hat{S} \|_2^2 \right) \right]$$

where $\Theta_{en}^\ast$ and $\Theta_{de}^\ast$ 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 $V$.
and eigenvector \( \mathbf{S} \) in this paper. First, the overview of proposed EMEV feedback architecture is shown. Then, the NN desgned 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 \( \varepsilon \) and reconstructs eigenmatrix \( \mathbf{V} \) and eigenvector \( \mathbf{S} \). 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. Fig. 2 is the illustration of its framework.

![Algorithm 1](image_url)

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

```
Input: \( \mathbf{H} \in \mathbb{C}^{N_{RB} \times N_t \times N_t} \) ← CSI matrix;
Output: \( \hat{\mathbf{S}} \in \mathbb{R}^{N_{RB} \times N_t} \) ← Reconstructed eigenvector;
\( \hat{\mathbf{V}} \in \mathbb{C}^{N_{RB} \times N_t \times N_t} \) ← Reconstructed eigenmatrix; \( id \) ← Channel type;
1 Stage I: UE operations:
2 SVD transformation: Initialize \( \mathbf{U} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}, \mathbf{S} \in \mathbb{R}^{N_{RB} \times N_t}, \mathbf{V} \in \mathbb{C}^{N_{RB} \times N_t \times N_t}; \)
3 for \( i = 1, \cdots, N_{RB} \) do
4 \( \mathbf{U}_i, \mathbf{S}_i, \mathbf{V}_i = f_{svd}(\mathbf{H}(i,\cdot,\cdot)) \)
5 if \( \mathbf{U}_i \cdot \mathbf{S}_i \cdot \mathbf{V}_i = \mathbf{H}(i,\cdot,\cdot) \) then
6 \( \mathbf{U}(i,\cdot,\cdot) = \mathbf{U}_i; \mathbf{S}(i,\cdot) = \mathbf{S}_i; \mathbf{V}(i,\cdot,\cdot) = \mathbf{V}_i. \)
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: \( (\hat{\mathbf{V}}, \hat{\mathbf{S}}) = f_{de}(\varepsilon) \)
```

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 \( \mathbf{S} \in \mathbb{R}^{N_{RB} \times N_t} \). The high-dimensional feature maps after convolution layers will be compressed to
one dimensional feature $\xi_V$ and $\xi_S$ by fully-connected layers. Thus, the feature extraction module can be described as,

$$\begin{align}
(\xi_V, \xi_S) &= L_{fc} \left[ L_{conv} \left( V, S, \Omega_{conv}, \Omega_{fc} \right), \Omega_{fc} \right] \\
\text{(15)}
\end{align}$$

where $L_{fc}(\cdot), L_{conv}(\cdot)$ represent fully-connected layer and convolution layer respectively, and $\Omega_{fc}, \Omega_{conv}$ are their corresponding weight values. Then, $\xi_V$ and $\xi_S$ pass through the transcoding module and output codewords $\varepsilon$ at specified system compression ratio. The attention mechanism\(^5\) based attention residual block and fully-connected layer are combined into a transcoding module, which is formulated as,

$$\varepsilon = L_{fc} \left[ L_{att}^{(5)} (\xi_V, \xi_S, \Omega_{att}), \beta_{CR}, \Omega_{fc} \right]$$

(16)

where $L_{att}^{(5)}$ indicates 5 loops of attention residual block, $\beta_{CR}$ is the input system compression ratio, and $\Omega_{att}$ is the corresponding weight value. Therefore, the length of $\varepsilon$ is determined by system $\beta_{CR}$, which can be defined as,

$$L_{\varepsilon} = L_{R(V)} + L_{R(\varepsilon)} + L_{id}$$

$$\text{(17)}$$

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

$$\begin{align}
(\hat{V}, \hat{S}) &= L_{conv} \left\{ L_{res}^{(3)} \left[ L_{fc} (\varepsilon, \Omega_{fc}), \Omega_{res} \right], \Omega_{conv} \right\} \\
\text{(18)}
\end{align}$$

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

\(^5\)The attention mechanism will be described separately when analyzing the transcoding module.

![Algorithm 2: The proposed EMEVNet algorithm for eigenmatrix and eigenvector feedback architecture.](image)

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

*Input:* $V \in \mathbb{C}^{N_{RB} \times N_f \times N_t}, S \in \mathbb{R}^{N_{RB} \times N_f \times N_t};$

1. Randomly initialize NN weight parameters $\Theta_{en}, \Theta_{de}$;
2. For $t = 1, \cdots, \tau$
   1. $\varepsilon = L_{fc} \left[ L_{conv} \left( V, S, \Omega_{conv}, \Omega_{fc} \right), \Omega_{fc} \right]$;
   2. $\varepsilon = L_{fc} \left[ L_{att}^{(5)} (\xi_V, \xi_S, \Omega_{att}), \beta_{CR}, \Omega_{fc} \right]$;
   3. $\hat{V}, \hat{S} = L_{conv} \left\{ L_{res}^{(3)} \left[ L_{fc} (\varepsilon, \Omega_{fc}), \Omega_{res} \right], \Omega_{conv} \right\}$;
3. **Training stage:**
   1. $\text{loss}_t = \mathbb{E} \left\{ \| V - \hat{V} \|^2_2, \| S - \hat{S} \|^2_2 \right\}$
   2. If $\text{loss}_t$ converges to $\text{loss}^*$ then
     1. Break;
   3. Else if $\text{loss}_t$ is not updated after 20 loops then
     1. $\eta = \eta \times 0.7$;
     2. $\Omega \leftarrow \text{Adam}(\Omega, \eta, \nabla \text{loss}_t)$;
   4. End
5. $[\Theta_{en}^*, \Theta_{de}^*] \leftarrow [\Omega_{fc}, \Omega_{att}, \Omega_{conv}, \Omega_{res}]$;
6. Save $f_{en}(\beta_{CR}, \Theta_{en}^*), f_{de}(\beta_{CR}, \Theta_{de}^*)$;
7. 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 × N_t × N_t × 2 | –             | –     |
| Input(S)   | –                | –          | N_RB × N_t × 1 | –             | –     |
| Conv2D_1   | Filter = 2,      | Leaky Relu | N_RB × N_t × N_t × 2 | 2 × 2 × 3^2   | (N_RB × N_t × N_t × 2) × (2 × 3^2) |
| Conv2D_2   | Kernel = 3,      | Leaky Relu | N_RB × N_t × N_t × 8 | 2 × 2 × 3^2   | (N_RB × N_t × N_t × 8) × (2 × 3^2) |
| Conv3D_1   | Filter = 8,      | Leaky Relu | N_RB × N_t × N_t × 8 | 8 × 2 × 3^2   | (N_RB × N_t × N_t × 8) × (2 × 3^2) |
| Conv3D_2   | Kernel = 3,      | Leaky Relu | N_RB × N_t × 8 | 8 × 2 × 3^2   | (N_RB × N_t × 8) × (2 × 3^2) |
| FCLayer_1(V) | Units = L_y  | Relu      | L_y x 1       | [N_RB × N_t × N_t × 8] x L_y | 2 x [N_RB x N_t x N_t x 8] x L_y |
| FCLayer_1(S) | Units = L_y  | Relu      | L_y x 1       | [N_RB × N_t × 8] x L_y | 2 x [N_RB x N_t x 8] x L_y |

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,

\[ Time\{Conv2D\} \sim O(S_M \cdot K^2 \cdot C_{in} \cdot C_{out}) \]  \hfill (19)
\[ Time\{Conv3D\} \sim O(V_M \cdot K^3 \cdot C_{in} \cdot C_{out}) \]  \hfill (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,

\[ Time\{fc\} \sim O(2 \cdot L_{in} \times L_{out}) \]  \hfill (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.

\[ Space\{Conv2D\} \sim O(K^2 \cdot C_{in} \times C_{out}) \]  \hfill (22)
\[ Space\{Conv3D\} \sim O(K^3 \cdot C_{in} \times C_{out}) \]  \hfill (23)
\[ Space\{fc\} \sim O(L_{in} \times L_{out}) \]  \hfill (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. [1] 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.

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

\[ s_i = f(q, k_i) = \begin{cases} q^T k_i \\ q^T W k_i \\ \frac{q^T k_i}{\sqrt{d}} \\ v \cdot \tanh(W q + U k_i) \end{cases} \]  \hfill (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,

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

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

\[ z = \text{Attention}(q, k, v) = \sum_i a_i \cdot v_i \]  \hfill (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)        | –                | –          | $L_{ξ_V} \times 1$ | –              | –      |
| Input(ξ_S)        | –                | –          | $L_{ξ_S} \times 1$ | –              | –      |
| Attention_res(V, S) | Head_num = 2, – | –          | $L_{ξ_V} \times 1$ | $2 \times (L_{ξ_V}^2 + L_{ξ_S}^2)$ | $8 \times (L_{ξ_V}^2 + L_{ξ_S}^2)$ |
| Attention_res(V, V) | Key_dim = 3, –  | –          | $L_{ξ_V} \times 1$ | $2 \times (L_{ξ_V}^2 + L_{ξ_V}^2)$ | $8 \times (L_{ξ_V}^2 + L_{ξ_V}^2)$ |
| FCLayer_codewords  | Units = $L_{c}$  | Linear     | $L_{c} \times 1$ | $(L_{c} \times 1) \times (L_{c} \times 1)$ | $2 \times (L_{c} \times 1) \times (L_{c} \times 1)$ |

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 |
|--------------|------------------|------------|--------------|----------------|-------|
| FCLayer_2(V) | Units = $[N_{RB} \cdot N_1 \cdot N_t \cdot 2]$ | Linear     | $[N_{RB} \cdot N_1 \cdot N_t \cdot 2]$ | $2 \times L_x \times [N_{RB} \cdot N_1 \cdot N_t \cdot 2]$ | $2 \times L_x \times [N_{RB} \cdot N_1 \cdot N_t \cdot 2]$ |
| FCLayer_2(S) | Units = $[N_{RB} \cdot N_t \cdot 1]$ | Linear     | $[N_{RB} \cdot N_t]$ | $L_x \times [N_{RB} \cdot N_t]$ | $2 \times L_x \times [N_{RB} \cdot N_t]$ |
| Conv3D_res   | Filter = [2,8,2], – | –          | $N_{RB} \times N_1 \times N_t \times 2$ | $(2 + 8 + 2) \times 2 \times 3^2$ | $(N_{RB} \times N_1 \times N_t \times 12) \times (12 \times 3^2)$ |
| Conv2D_res   | Kernel = 3, –    | –          | $N_{RB} \times N_1 \times N_t \times 2$ | $(1 + 8 + 2) \times 2 \times 3^2$ | $(N_{RB} \times N_1 \times N_t \times 11) \times (11 \times 3^2)$ |
| Conv3D_3     | Filter=2, Kernel=3, Tanh | Linear | $N_{RB} \times N_1 \times N_t \times 2$ | $2 \times 2 \times 3^2$ | $(N_{RB} \times N_1 \times N_t \times 2) \times (2 \times 3^2)$ |
| Conv2D_3     | Filter=2, Kernel=3 | Linear     | $N_{RB} \times N_t$ | $1 \times 2 \times 3^2$ | $(N_{RB} \times N_t \times 1) \times (2 \times 3^2)$ |

As is shown in Fig. 3, transcoding module is combined with two attention residual blocks and a fully-connected layer. The attention residual block receives two parallel input tensors $X$ and $X_{key}$. If $X = X_{key}$, then we call it self-attention residual block, which pays more attention to self hidden information. If $X \neq X_{key}$, the attention residual block will embed the $X_{key}$’s feature information in $X$, 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 multi-head attention layer can be respectively formulated as,

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

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

where $L_{in}$ is the length of input tensor, and $d_{in}$ denotes the value of hyper-parameter head_number. 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 $\hat{V}$ and eigenvectors $\hat{S}$. Fig. 5 shows the detailed decoder module and the convolution residual block applied in it. We design two different branches to reconstruct $\hat{V}$ and $\hat{S}$ respectively. Two different fully-connected layers are utilized to extract the high-dimensional features of $V$ and $S$ from codewords $e$. And then we design convolutional residual blocks to reconstruct $\hat{V}$ and $\hat{S}$. Since 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 Github.

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 | CDL-A | CDL-B | CDL-C | CDL-D | CDL-E |
|----------------------|------|-------|-------|-------|-------|-------|
| NRB                  | 13   | 50    | 50    | 50    | 50    | 50    |
| 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, 65 ns |       |       |       |       |       |
| 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 $D_{sp}$ to train specific network $N_{sp}$ for each channel. For comparison, we mix 10,000 samples of five CDL channels to obtain $D_{mix}$, and train a general network $N_{mix}$. The operations of datasets and neural networks can be respectively expressed as,

$$N_{sp}^* \leftarrow D_{sp} = \left\{ \mathbf{H}^{(50k)}_c \right\}$$ (30)

$$N_{mix} \leftarrow D_{mix} = \left\{ \mathbf{H}^{(10k)}_A, \mathbf{H}^{(10k)}_B, \cdots, \mathbf{H}^{(10k)}_E \right\}$$ (31)

where $\mathbf{H}^{(50k)}_c$ denotes the 50,000 samples of each channel environment represented by different subscripts, and $\mathbf{H}^{(10k)}_c$ 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 $\varepsilon$ to the BS. The length of codewords $L_{c}$ is related to the overhead of feedback, which will affect the spectral efficiency of communication system. And the $L_{c}$ is decided by compression ratio $\beta_{CR}$. 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 $\beta_{CR}$ 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 $L_{c}$ can be expressed as,

$$L_{c} = \frac{\mathbb{E}(\mathbf{H}) + \mathbb{E}(\mathbf{S})}{\beta_{h}} + 1$$

$$= \frac{N_{RB} \times N_r \times N_t \times 2}{\beta_{h}} + 1$$ (32)

where $\beta_{h}$ is the compression ratio of $\mathbf{H}$. In order to guarantee the effectiveness of the comparative simulation experiments, fixed $L_{c}$ is given in the follow discussion. To ensure the same $L_{c}$, the compression ratio of EMEVNet can be defined as,

$$\beta_{emev} \approx \frac{\mathbb{E}(\mathbf{V}) + \mathbb{E}(\mathbf{S})}{\mathbb{E}(\mathbf{H}) + \mathbb{E}(\mathbf{S})} \beta_{h}$$

$$= \frac{N_{RB} \times (N_r \times N_t \times 2 + N_s)}{N_{RB} \times N_r \times N_t \times 2} \beta_{h}$$ (33)

where $\beta_{emev}$ 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 epochs. 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:

- $L_{c} = \{416, 208, 104, 52, 26, 13, 6\}$ 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 ($\rho$) to measure the reconstruction accuracy, which can be respectively defined as,

$$NMSE \left( \mathbf{V}, \hat{\mathbf{V}} \right) = E \left\{ \| \mathbf{V} - \hat{\mathbf{V}} \|_2^2 / \| \mathbf{V} \|_2^2 \right\}$$

$$= 10 \log \left( E \left\{ \| \mathbf{V} - \hat{\mathbf{V}} \|_2^2 / \| \mathbf{V} \|_2^2 \right\} \right)$$ (34)

$$NMSE \left( \mathbf{S}, \hat{\mathbf{S}} \right) = E \left\{ \| \mathbf{S} - \hat{\mathbf{S}} \|_2^2 / \| \mathbf{S} \|_2^2 \right\}$$

$$= 10 \log \left( E \left\{ \| \mathbf{S} - \hat{\mathbf{S}} \|_2^2 / \| \mathbf{S} \|_2^2 \right\} \right)$$ (35)

$$\rho \left( \mathbf{V}, \hat{\mathbf{V}} \right) = E \left\{ \| \mathbf{V}^* \cdot \hat{\mathbf{V}} \|_2 / \| \mathbf{V} \|_2 \right\}$$ (36)

$$\rho \left( \mathbf{S}, \hat{\mathbf{S}} \right) = E \left\{ \| \mathbf{S}^* \cdot \hat{\mathbf{S}} \|_2 / \| \mathbf{S} \|_2 \right\}$$ (37)

where $(a, b)$ 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. $\rho$ 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...
50,000 samples of CDL-A channel, i.e. $D_{sp}^A = \{H_A^{(50k)}\}$, as simulation datasets. We set the training, validation and testing datasets obeying the ratio of 70 : 15 : 15. Before training EMEVNet, the datasets $D_{sp}^A$ need to carry out SVD transformation according to Algorithm 1. And the generated $V, S$ are the input of EMEVNet. After NN training stage according to Algorithm 2, we can obtain the trained $N_{sp}^A$, 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 $V \in \mathbb{C}^{N_{RB} \times N_t \times N_t}$ 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. $V(3) \in \mathbb{C}^{N_t \times N_t}$. Then, the Euclidean norm of $V(3)$ is applied for convenient plotting, which can be interpreted as the power distribution. Fig. 6a shows the initial $V$ sample at UE and Fig. 6b to Fig. 6h show the reconstructed $\hat{V}$ at BS with different compression ratio from $L_e = 416$ to $L_e = 6$. For eigenvector $S \in \mathbb{R}^{N_{RB} \times N_r}$, it is a 2D real-valued matrix. We directly plot the figure without any processing. Fig. 6i shows the initial $S$ at UE and Fig. 6j to Fig. 6p show the reconstructed eigenvector $\hat{S}$ with the same settings as mentioned above. After many experiments, the shape of $V$ of different RBs is almost the same. They all present diagonal power distribution, and the edges are serrated. Moreover, the power distribution of $V$ is a symmetrical image.

As shown in Fig. 6, with the reduction of feedback codewords $L_e$, the reconstructed $\hat{V}$ 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 \( L_c = 26 \), i.e. \( \beta_h = 256 \) and \( \beta_{emev} = 4096 \), the reconstructed \( \hat{V} \) losses almost all edge information. It can be seen from the figure that when \( L_c \leq 26 \), the edge power of \( \hat{V} \) cannot be reconstructed. However, we can ensure that the distribution of \( \hat{V} \) remains unchanged and the diagonal power is hardly affected in any case. As for \( \hat{S} \) at the BS, it can be well reconstructed under almost any \( L_c \).

The detailed numerical results of \( N_{mix}^s \), 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 \( \hat{V} \) and \( \hat{S} \). Therefore, this part compares the performance of specific networks \( N_{sp}^s \) trained with large datasets \( D_{sp}^s \) and general network \( N_{mix}^s \) trained with mixed datasets \( D_{mix} \). In order to completely verify the superiority of the proposed architecture, we have checked five CDL channels defined by 3GPP. The testing objects are \( \{N_{mix}^A, N_{mix}^B, N_{mix}^C, N_{mix}^D, N_{mix}^E\} \) and the baseline is set as \( \{N_{mix}^A\} \). For each experiment, we give four evaluation indexes defined in Section IV-A3 including \( NMSE(\hat{V}, \hat{V}), NMSE(\hat{S}, \hat{S}), \rho(\hat{V}, \hat{V}) \) and \( \rho(\hat{S}, \hat{S}) \).

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-A3. The horizontally adjacent figures respectively show the reconstruction performance curves of \( \hat{V} \) and \( \hat{S} \) 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 \( \rho \) 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 \( \rho \) performance, the circle marked lines represent specific network \( N_{sp}^s \), and the triangle marked lines represent general network \( N_{mix}^s \).

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-A3. The performance of reconstruction is getting worse with the decrease of \( L_c \), i.e., \( NMSE\) (dB) tends to be larger and \( \rho \) 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 \( N_{sp}^s \) is better than general network \( N_{mix}^s \). It can also be found that two lines with the same color are almost parallel, which shows that the performance improvement of \( N_{sp}^s \) is relatively stable compared with \( N_{mix}^s \), and is less affected by \( L_c \).
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 $N_{sp}$.

**D. Robustness Analysis**

This subsection proves the robustness of proposed architecture by comparing EMEVNet with the classical CSI feedback scheme CsiNet [11]. In order to narrate conveniently, $N_{csi}^*$ is defined as the classical CsiNet framework. We distinguish different channels by superscripts, e.g. $N_{A}^*$ represents CsiNet based on CDL-A channel. For fairness, specific datasets with 50,000 samples are used in this subsection, and $N_{sp}^*$ and $N_{csi}^*$ are trained respectively. Similarly, the experiments in this part also verify 5 channel types and 7 different compression ratios set as Section IV-A3. 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 $H$, we keep the same feedback overhead and carry out SVD transform on the decoded $H$ at the BS. The testing objects are $\{N_{sp}^A, N_{sp}^B, N_{sp}^C, N_{sp}^D, N_{sp}^E\}$, and the baselines are set as $\{N_{csi}^A, N_{csi}^B, N_{csi}^C, N_{csi}^D, N_{csi}^E\}$.

The experimental results can be seen in Fig. 8. The arrangement of different subfigures is the same as that in Fig. 4. Each figure has two y-axes of different scales, corresponding to $NMSE (dB)$ and $\rho$ respectively. In addition, there are four performance curves in each figure, and the solid blue lines show $NMSE$ performance, the dotted red lines represent $\rho$ performance, the circle marked lines represent EMEVNet $N_{sp}^*$, and the pentagram marked lines represent the baseline, i.e. CsiNet $N_{csi}^*$. 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 $\hat{S}$ at the BS. This can be explained as the baseline $N_{csi}^*$ is greatly affected by the length of codewords $L_c$. As for eigenmatrix $\hat{V}$, except Fig. 8e shows that the baseline is better than EMEVNet with long $L_c$, the other results all show the performance of EMEVNet is better with all $L_c$. As for eigenvector $\hat{S}$, we can find baseline has better performance with long $L_c$ in all channels. For NLOS channels (CDL-A, CDL-B and CDL-C), when the length of codewords decreases to $L_c = 104$, the performance of EMEVNet becomes better than that of baseline. While discussing LOS channels (CDL-D and CDL-E), this threshold is relaxed to $L_c = 26$. 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 $L_c$. However, the performance of EMEVNet is relatively stable. Especially in the case of limit $L_c$, 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 $N_{sp}^*$ and $N_{csi}^*$ in the table correspond to the special EMEVNet.
Meanwhile, from the numerical analysis, the reconstruction of mmWave FDD systems was proposed. The key idea of our approach is almost perfectly consistent with the results in Section IV-B. From the analysis of numerical results, we can conclude that $\hat{S}$ is almost perfectly reconstructed, while the performance of $\hat{V}$ is worse with the decrease of $L_e$. 

### 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, Jul. 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. 3–53, Mar. 2022.
[6] K. B. Letaisf, 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. Froytlog, 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. Adesibi, T. Ohtskuki, 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 Select Areas in Communications, vol. 39, no. 8, pp. 2388–2406, Aug. 2021.