

## Research Article

# A Next-Generation Codebook Evolution Strategy for Massive Arrays Using Deep Neural Networks

**Minwoo Choi and Wonjin Sung** 

*Department of Electronic Engineering, Sogang University, Seoul 04107, Republic of Korea*

Correspondence should be addressed to Wonjin Sung; [wsung@sogang.ac.kr](mailto:wsung@sogang.ac.kr)

Received 3 June 2022; Revised 14 November 2022; Accepted 19 November 2022; Published 25 November 2022

Academic Editor: Rajkishor Kumar

Copyright © 2022 Minwoo Choi and Wonjin Sung. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The discrete Fourier transform (DFT)-based codebook is currently among the mostly commonly adopted codebooks for beamforming using arrays of different shapes and sizes, including the large-scale two-dimensional uniform planar array (UPA). DFT-based codevectors can be easily generated in arbitrary angle resolutions and apply well to millimeter-wave (mmWave) channels due to their directive nature of resulting beams. However, a fixed set of codevectors is applied regardless of the user distributions and the propagation environment, which may exhibit limited beamforming performance under certain transmission scenarios. In this paper, we propose a new way of generating a set of beamforming vectors for multiple-input multiple-output (MIMO) transmission using massive arrays under the limited feedback of the channel state information (CSI). Precoder matrix indicator (PMI) and channel quality indicator (CQI) reports from the users have become the sources for the generation of a new set of codevectors, which are autonomously determined by the deep learning (DL) module at the base station (BS). The process is operated in an iterative fashion to produce updated versions of the codebook with the reduced return of the loss function at the deep neural network (DNN). The time-varying codebook for each BS automatically reflects the characteristics of a given wireless environment to adapt to its channel and traffic conditions. The reference signal (RS) at the BS is periodically transmitted in the form of beamformed CSI-RS, thus the operation is transparent to the users of the system and no significant specification changes are necessary. A simple plug-and-play type of BS installation suffices to achieve the potential gain of the proposal, which is demonstrated by the implementation details of the DL engine and the corresponding performance simulation results.

## 1. Introduction

With the development of advanced mobile devices and related multimedia applications such as ultra-high-definition (UHD) and virtual reality (VR), an ever-increasing throughput demand has rapidly made the sub-6 GHz band crowded [1, 2]. Upper frequency bands including the millimeter-wave (mmWave) with rich spectrum availability are successfully supplementing the existing wireless spectrum [3, 4]. Despite the severe attenuation of the mmWave signals, well-designed beamforming strategies using large-scale antenna arrays render reliable multiple-input multiple-output (MIMO) transmission possible [5–7]. Since the massive MIMO transmission of mmWave signals is enabled by highly directional beams, the accuracy of beam alignment

has a significant influence on the link quality and reliability [8]. To acquire the knowledge of the channel state information (CSI) for beamforming, the base station (BS) periodically transmits the reference signal (RS) using the CSI-RS in preallocated time and frequency resources [9]. Upon the observation of the received RS, the user equipment (UE) determines which codevector is best suited for beamforming at what level of the preference degree. The notification of this determination is reported back to the BS by sending the precoding matrix indicator (PMI) and the channel quality indicator (CQI) [10].

For MIMO transmission under the constraint of the limited CSI feedback, beamforming performance heavily depends on the specific codebook in use. Different design criteria can be applied for the codebook construction, such

as Grassmannian line packing [11], the generalized Lloyd algorithm [12], and the random vector quantization (RVQ) [13]. Performance comparison of these codebooks together with the discrete Fourier transform (DFT)-based codebook has been reported in [14]. The DFT-based codebook is known to approximately match the optimal beamforming vectors [15, 16] and is well suited to the mmWave channel with the sparsity nature. Despite the wide adoption of the DFT-based codebook including the 3GPP specifications [9, 10], it does not reflect the geographical characteristics of the cell or adapt to time-varying UE distributions. Some of the codevectors within the codebook may be rarely utilized, thereby diminishing the beamforming efficiency per each feedback bit.

Deep learning (DL) techniques possess a large potential to enhance and optimize the beamforming codevectors. Motivated by successes in classification and image processing applications [17], the deep neural network (DNN) with its ability to learn hidden characteristics of data has been applied to many areas of physical layer wireless communications [18]. By extracting high-dimensional features of physical layer aspects, the performance very close to that of conventional algorithms is achievable with a fraction of computational complexity [19]. In [20], a convolutional neural network (CNN) technique is applied to allocate beams to users by modeling their positions as a gray-scale image. A beam training scheme that matches the combination of beams to the strongest channel paths was proposed in [21], producing larger beam gains and improved signal coverage. However, all aforementioned algorithms perform supervised learning which requires labels and thus are not inherently environment adaptive. Upon the change of surroundings or channel conditions, a substantial amount of offline data may be needed to reproduce the labels for learning. Such supervised nature of learning is also found in [22], where a codebook update scheme is proposed by using the neural network which adapts to the channel. The network takes the channel matrix as the input to produce the beamforming vectors as weights of the DNN. The codebook generation scheme is further enhanced in [23], by eliminating the need to use the full channel information as the DNN input and employing the self-supervised learning process for the channel. Authors in [24] apply DL for the system sum-rate maximization to obtain the beamformer power allocation parameters with the channel matrix as the DNN input. A beam design technique without using the CSI knowledge is proposed in [25] by utilizing the Bayesian statistical learning, which significantly reduces the search time compared to exhaustive beam training.

In this paper, we propose an autonomous codebook generation strategy using DL which improves aforementioned schemes by eliminating many of their limitations. The generated codebook is automatically updated by using the feedback data from the UEs as the input to the neural network software at the BS. The input consisting of the PMI and CQI is a part of conventional uplink report data in existing wireless systems and therefore required specification changes for this proposed operation so that it can be kept at a minimal level. The codebook is continuously

updated via unsupervised learning using the system report data without additional offline training. While conventional codebook generation methods such as the VQ require and the UE location information determine codevectors, the DNN continuously produces the updated codebook which reflects the UE mobility and geographical characteristics of the cell. The real-time interaction between the neural network and the system signaling data produces the environment-aware codebook which is autonomously customized to the transmission condition at given time and location. To demonstrate the effectiveness of the proposed method, we evaluate the transmission performance including the signal correlation and the sum-rate. The evaluation result is compared to performance of existing codebooks, and the amount of gain is quantified.

Key advantages of the proposed method which differentiate it from earlier related works include the following: (1) The codebook generation is based on unsupervised learning, enabling an adaptive operation for changing mobile traffic conditions as well as environmental variations. The BS can be simply installed in a conventional manner with expected beamforming performance enhancement regardless of the time and space of operation. (2) The DNN uses a small set of parameters as its input and output, without the need to handle a huge amount of complex data such as the full CSI which can significantly slow down the learning process. (3) No specific hardware constraints including the existence of phase shifters are imposed in utilizing the DL method which autonomously produces the codebook with progressively improving accuracy. (4) Perhaps most importantly, the proposal is 3 GPP standard compliant and hence can be put into use without changing the control signaling. It is also user transparent in the sense that the UE can report back the conventional PMI and CQI reports for the received beamformed CSI-RS. The rest of this paper is organized as follows: In Section 2, the system model is described and the DL-based codebook generation strategy is explained including the details of the neural network parameters. In Section 3, performance evaluation results as well as specific simulation scenarios are presented. Conclusions are given in Section 4. Symbol  $\mathbb{C}^{A \times B}$  denotes the space of complex matrices of  $A$  rows and  $B$  columns, and  $\mathcal{CN}(\mu, \sigma^2)$  represents the complex Gaussian random variable with mean  $\mu$  and variance  $\sigma^2$ . Also,  $\|\bullet\|$  denotes the Frobenius norm and  $\mathbb{E}[\bullet]$  is the expectation operator.

## 2. Materials and Methods

*2.1. System Setting.* We consider multiuser (MU)-MIMO downlink transmission from the BS with  $M_t$  antenna elements to  $K$  UEs, each equipped with  $M_r$  antenna elements. The received signal vector for the  $k$ -th UE is expressed as

$$\mathbf{y}_k = \sqrt{\rho_k} \mathbf{h}_k \mathbf{w}_k s_k + \sum_{j \neq k}^K \sqrt{\rho_k} \mathbf{h}_k \mathbf{w}_j s_j + \mathbf{n}_k, \quad (1)$$

where  $\rho_k$  is the average received power,  $\mathbf{h}_k \in \mathbb{C}^{M_r \times M_t}$  is the channel matrix such that

$\mathbb{E}[\|\mathbf{h}_k\|^2] = M_r M_t$ , and  $\mathbf{w}_k \in \mathbb{C}^{M_t \times 1}$  is the beamforming vector for data symbol  $s_k$  with power constraint  $\|s_k\|^2 = 1/K$ . Noise vector  $\mathbf{n}_k$  includes independent and identically distributed (i.i.d.) complex Gaussian  $\mathcal{CN}(0, \sigma^2)$ . Using the combining vector  $\mathbf{u}_k \in \mathbb{C}^{M_r \times 1}$ , the UE combines the signal as  $r_k = \mathbf{u}_k^H \mathbf{y}_k$ . The corresponding signal-to-interference-plus-noise ratio (SINR) is

$$\gamma_k = \rho_k / k \|\mathbf{u}_k^H \mathbf{h}_k \mathbf{w}_k\|^2 / \sum_{j \neq k} \frac{\rho_j}{K} \|\mathbf{u}_k^H \mathbf{h}_k \mathbf{w}_j\|^2 + \sigma^2. \quad (2)$$

The achievable sum-rate for all UEs becomes

$$R = \sum_{K=1}^K \log_2(1 + \gamma_k). \quad (3)$$

Using the extended Saleh–Valenzuela model for multipath wireless transmission, channel matrix  $\mathbf{h}_k$  can be written as [26, 27]

$$\mathbf{h}_k = \sum_{l=1}^{I_k} \alpha_{k,l} \mathbf{a}_r(\phi_{k,l}^r, \theta_{k,l}^r) \mathbf{a}_t^H(\phi_{k,l}^t, \theta_{k,l}^t), \quad (4)$$

where  $I_k$  is the number of propagation paths,  $\alpha_{k,l}$  is the complex gain of the  $l$ -th path,  $\phi_{k,l}^r(\theta_{k,l}^r)$  and  $\phi_{k,l}^t(\theta_{k,l}^t)$  are its azimuth (zenith) angle of arrival and departure, respectively. The complex gain follows the following distribution:

$$\alpha_{k,l} \sim \mathcal{CN}(0, \Lambda_r(\phi_{k,l}^r, \theta_{k,l}^r) \Lambda_t(\phi_{k,l}^t, \theta_{k,l}^t) \sigma_{k,l}^2), \quad (5)$$

where  $\Lambda_r(\phi_{k,l}^r, \theta_{k,l}^r)$  and  $\Lambda_t(\phi_{k,l}^t, \theta_{k,l}^t)$  represent the directional gains of antenna elements at corresponding arrival and departure angles, respectively, and  $\sigma_{k,l}^2$  denotes the average power of the  $l$ -th path. Vectors  $\mathbf{a}_r(\bullet)$  and  $\mathbf{a}_t(\bullet)$  are receive and transmit array response vectors, respectively. For a uniform linear array (ULA) with  $M$  horizontal elements, the array response vector without the zenith angle argument is given by

$$\mathbf{a}_{\text{ULA}}(\phi) = [1, e^{j\kappa d \sin \phi}, \dots, e^{j\kappa d (M-1) \sin \phi}]^T, \quad (6)$$

where  $\kappa = 2\pi/\lambda$  is the wave number,  $\lambda$  is the carrier wavelength, and  $d$  is the interelement spacing. In the case of a uniform planar array (UPA) with  $M_h$  horizontal elements and  $M_v$  vertical elements, the array response vector is expressed as [28]

$$\mathbf{a}_{\text{UPA}}(\phi, \theta) = [1, \dots, e^{j\kappa d (h \sin \phi \sin \theta + v \cos \theta)}, \dots, e^{j\kappa d ((M_h-1) \sin \phi \sin \theta + (M_v-1) \cos \theta)}]^T. \quad (7)$$

For  $0 < h < M_h - 1$  and  $0 < v < M_v - 1$ . The total number of antenna elements in the UPA is  $M = M_h M_v$ .

**2.2. DFT Codebook.** A codebook of size  $N$  is a set of beamforming codevectors

$$\mathcal{C} = \{\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{N-1}\}, \quad (8)$$

where each vector can be chosen as the preferred codevector by the UE. The index of the preferred codevector for the  $k$ -th UE is fed back to the BS in the form of the PMI report  $p_k \in \{0, 1, \dots, N-1\}$ . For the  $B$ -bit feedback, the codebook size becomes  $N = 2^B$ . Upon the reception of the PMI and CQI reports from the UEs, the BS schedules a set of UEs for simultaneous multiuser transmission at each transmission time instance and determines the precoding matrix. The preferred codevector can either be directly applied as the component vector in the precoding matrix or used as a CSI estimate to compute the matrix as in the case of zero-forcing (ZF) precoding. The CQI provides information on the quality of the transmission link and is determined by channel measurement means such as the SINR estimate.

For the case of the DFT-based codebook, the codevectors for the ULA of horizontal length  $M_h$  are expressed as

$$\mathbf{c}_m^{(h)} = \frac{1}{\sqrt{M_h}} \left[ 1, e^{j\frac{2\pi}{N_h} m}, e^{j\frac{2\pi}{N_h} 2m}, \dots, e^{j\frac{2\pi}{N_h} (M_h-1)m} \right]^T. \quad (9)$$

For  $m = 0, 1, \dots, N_h - 1$ ,  $N_h$  is the horizontal codebook size. Similarly, the DFT codevectors for the ULA of vertical length  $M_v$  are given as

$$\mathbf{c}_n^{(v)} = \frac{1}{\sqrt{M_v}} \left[ 1, e^{j\frac{2\pi}{N_v} n}, e^{j\frac{2\pi}{N_v} 2n}, \dots, e^{j\frac{2\pi}{N_v} (M_v-1)n} \right]^T. \quad (10)$$

For  $n = 0, 1, \dots, N_v - 1$ ,  $N_v$  is the vertical codebook size. Combining (9) and (10), the two-dimensional (2D) DFT codevectors for both horizontal and vertical beam tilting are determined as

$$\mathbf{c}_{m,n} = \mathbf{c}_m^{(h)} \otimes \mathbf{c}_n^{(v)}, \quad (11)$$

where  $\otimes$  denotes the Kronecker product. For example,  $N_h = N_v = 8$  results in the 2D codebook of size  $N = N_h N_v = 64$ , requiring  $B = 6$  bit feedback for the PMI report. Since the beam directions for the codevectors in (11) are distributed over the entire horizontal and vertical directions, some of them may be more heavily used than others while some may rarely be chosen as the preferred vector. In an illustrative transmission scenario shown in Figure 1, most of the UE traffic is generated by vehicles on the road and pedestrians on the roadside, potentially making a good portion of the codevectors less utilized.

**2.3. Codebook Generation Using Deep Learning.** We intend to eliminate this problem of inefficient codebook utilization by introducing a new type of the codebook generation method based on deep learning. The proposed method



FIGURE 1: MU-MIMO transmission scenario in an urban intersection.

provides the real-time codebook updates by taking the collected statistics of the PMI and CQI from the field operation. Using the interaction between the DNN and the system statistics, the output codebook transforms into a more efficient form as the generation cycles iterate. The system operator can customize the efficiency metric if desired and apply it to the DNN loss function. The traffic condition may change depending on specific time of the day, e.g., more data users are connected inside the office building during the work hours and certain vehicular routes become massive sources of mobile connection during commuting hours. For all these circumstances, the generated codebook adapts itself accordingly to enhance the data transmission efficiency. As this adaptation takes place in an autonomous fashion, no additional efforts are needed for the hardware or software optimization at the time of BS installation.

The codebook update procedure starts with initializing the input to the DNN. The DNN input is composed of the initial codebook  $\mathcal{C}_0$  and the PMI counts  $\mathcal{P}_0$ . We assume that the conventional uniform DFT-based codebook is used for the initial input, although other codebooks can also be applied as the input including random codebooks. To limit the size of the input, only the phase information for the codebook is applied as the input to the DNN instead of the full complex vectors. Initial PMI counts  $\mathcal{P}_0 = \{i_0, \dots, i_{N-1}\}$  is a set of  $N$  random nonnegative integers. The codebook update procedure is described by the flowchart shown in Figure 2. While the codebook update status is turned on by the system, the DNN is trained online via unsupervised learning in an iterative fashion. The input matrix is composed of the codebook generated from the previous iteration and the PMI counts obtained from the UE feedback data. An updated codebook at current iteration is produced as the DNN output, which is applied for the DL transmission from the BS to the UEs in the form of beamformed CSI-RS. Unlike conventional nonprecoded CSI-RS transmitted from each antenna port, beamformed CSI-RS uses multiple beamforming codevectors in the spatial domain, and the UE chooses the best beam and feeds back its index [29]. Beamformed CSI-RS provides many benefits over

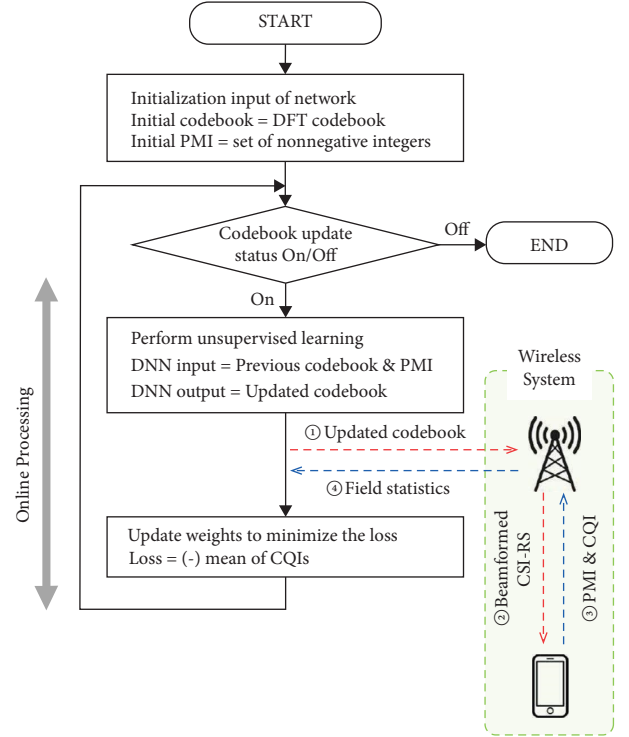


FIGURE 2: Algorithm flowchart for the autonomous codebook generation.

nonprecoded CSI-RS transmission, including less feedback and pilot resource overheads as well as the increased signal-to-noise ratio (SNR) for RS [30]. The PMI for the  $k$ -th UE is chosen to maximize the correlation power between the UE channel and the selected codevector in the codebook of size  $N$ , as given by [31].

$$p_k = \arg \max_{n \in \{0, \dots, N-1\}} \|\mathbf{h}_k \mathbf{c}_n\|^2. \quad (12)$$

The CQI is determined as the corresponding maximum correlation power

$$q_k = \max_n \|\mathbf{h}_k \mathbf{c}_n\|^2 = \|\mathbf{h}_k \mathbf{c}_{p_k}\|^2. \quad (13)$$

The PMI and CQI reports from the UEs are collected at the BS to form environmental statistics over a preset time duration, which is applied for the codebook generation at the next iteration. Let  $\mathcal{P}_i$  denote the set of  $N$  integer values which are the counts of PMI reports for  $N$  codevectors at the  $i$ -th iteration. Also, let  $\mathcal{Q}_i$  denote the collected set of CQI reports at the  $i$ -th iteration. The CQI values averaged over all elements in  $\mathcal{Q}_i$  is referred to as the average correlation power and is denoted by  $\mu_i$ . The loss function at the  $i$ -th iteration of unsupervised learning is defined to be the negative of the average correlation power

$$L_i = -\mu_i = -\mathbb{E}[q_k]. \quad (14)$$

Thus, set  $\mathcal{Q}_i$  is used for computation of the loss function of the DNN, so that the neural network weights are updated to maximize the average CQI. Set  $\mathcal{P}_i$  is applied as the input

to the DNN at the next iteration, together with the updated codebook produced as the DNN output at the current iteration. This iterative learning process involving the interaction between the DNN and the field statistics provides a progressively improving selection of beamforming codevectors while the codebook update status is on. If the codebook update is no longer necessary, the system turns off the codebook update status and the iterative learning process ends.

Figure 3 shows the structure of the proposed DNN, consisting of the input layer, multiple hidden layers, and the output layer. The  $N \times 3$  input matrix includes the set of  $N$  integer values for the PMI counts, and  $N$  pairs the azimuth and zenith angles for  $N$  codevectors. Utilization of angular parameters instead of full complex codevectors of length  $M$  decreases the dimension of the input matrix, which reduces the computational complexity of the DNN. Each of  $J_1$  hidden layers includes  $J_2$  nodes which are fully-connected and utilizes the rectifier linear unit (ReLU) activation function. The fully-connected output layer using the linear activation function determines the  $N$  updated codevectors by producing their angular parameters. The input to the DNN for unsupervised learning is the set of codebook parameters and the PMI counts obtained from the previous iteration, as indicated by the solid lines in the figure. The output is the updated codebook  $\mathcal{C}_i$ , which is subsequently applied for beamformed CSI-RS transmission for the UE CSI estimation. The collected statistics  $\mathcal{Q}_i$  is used to compute the loss function  $L_i$  in (14), and  $\mathcal{P}_i$  is fed back to the DNN as the input for the next iteration.

The operating process of the proposed codebook update algorithm can be described as follows: During the forward propagation step, output  $\mathbf{o}^{(l)}$  of  $l$ -th hidden layer represents the extracted features and is determined as

$$\mathbf{o}^{(l)} = \sigma(\mathbf{W}^{(l)} \mathbf{o}^{(l-1)}). \quad (15)$$

For  $l = 1, \dots, J_1$ ,  $\sigma(\cdot)$  is the ReLU activation function and  $\mathbf{W}^{(l)}$  is the weight matrix connecting the  $(l-1)$ -th and  $l$ -th hidden layers. Input to the first hidden layer  $\mathbf{o}^{(0)}$  is the  $N \times 3$  input matrix  $\mathbf{x}$  consisting of the PMI counts and the directional angles for  $N$  codevectors. The prediction function of the DNN with  $J_1$  hidden layers is given by

$$\mathbf{F}(\mathbf{x}) = \mathbf{W}^{(J_1+1)} \mathbf{o}^{(J_1)}, \quad (16)$$

which produces the  $N \times 2$  output matrix describing the directional angle values for the updated codevectors. The weight matrices are updated at the  $i$ -th iteration using the online gradient descent (OGD) described in [32] as

$$\mathbf{W}_i^{(l)} \leftarrow \mathbf{W}_i^{(l)} - \eta \nabla_{\mathbf{W}_i^{(l)}} L_i. \quad (17)$$

For  $i = 0, 1, 2, \dots, \eta$  is the learning rate,  $\nabla_{\mathbf{W}_i^{(l)}}$  is the gradient, and  $L_i$  is the loss function in (14) determined by using the updated codevectors in (16). The subscript  $i$  in each symbol denotes the update at the  $i$ -th iteration.

One of the key features of the proposed learning algorithm is the real-time interaction of the DNN with the field operational data, which provides the practicality and

adaptive capability to the algorithm. By properly choosing the time duration for the system feedback collection, the proposed scheme can provide timely adaptation to environmental and traffic condition changes as discussed in the next section.

### 3. Results and Discussion

The simulation environment for performance evaluation is shown in Figure 4, which models a typical urban intersection with the BS near the crossroad. The BS is located at  $(-25 \text{ m}, +25 \text{ m})$  from the intersection center in the figure, with height  $h_{\text{BS}} = 15 \text{ m}$  from the ground. The UPA of size  $M_t = M_h \times M_v = 8 \times 8$  is used for MU-MIMO transmission, with the downtilt angle of  $12^\circ$  and the array bearing angle facing the intersection center. The lengths of both horizontal and vertical lanes over which traffic data is collected are set to be 100 m, using 6 lanes each. The width of each lane is 3.5 m. The channel coefficients are generated based on the Saleh-Valenzuela model in (4), with the azimuth and zenith angles of multipath components following the Laplacian distribution. The mean of the distribution corresponds to the line-of-sight (LoS) direction of the target UE, and the standard deviation is determined by the spread angle specified in [33]. Unless otherwise stated, spread angle of 6 degrees is applied to channel. UEs with a single-antenna with isotropic radiation pattern are assumed for performance evaluation in this section.

The mobile user distribution is generated by the simulation of urban mobility (SUMO) package, which is widely adopted open-source traffic simulation software providing useful features including the import capability from map databases [33, 34]. Support tools perform various tasks such as the configuration of the traffic light system, vehicle speed, and lane change settings [35]. By integrating SUMO with the DNN module, the evaluation of learning performance based on traffic data very close to actual mobile movements can be achieved. Figure 5 illustrates the traffic simulation implementation using SUMO. A snap-shot example of the mobile UE generation is depicted in Figure 5(a), and the set of user locations loaded from the SUMO simulator over one iteration period is shown in Figure 5(b). The average number of mobile UEs at a given time instance is set to 15, each moving at the speed in the range of 0 to 60 km/h. The interval between two feedback instances by UEs range from 5 to 80 m-sec [36]. Among the PMI and CQI report values received by the BS, the DNN selectively collects one data pair at every second for locational diversity of feedback data. Then, approximately 3000 PMI and CQI data pairs are accumulated over the period of 200 sec, which is a preset time duration for one training iteration. Thus, the codebook update reflecting the changing UE traffic conditions can be performed in a few-minute time frame.

The DL algorithm is performed on Intel i5-9400F CPU with NVIDIA GeForce RTX 2060 GPU and 32 GB RAM using TensorFlow 2.0.0 and Keras 2.3.1 integrated with Python 3.7.6. The Glorot normal initializer is used for weight initialization, and the Adam optimizer with learning rate of

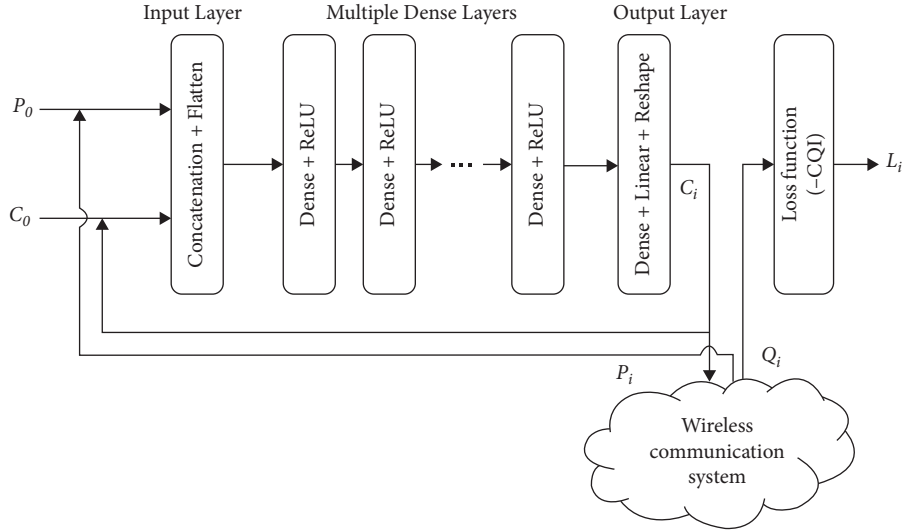


FIGURE 3: DNN structure and system feedback information flow for the unsupervised learning procedure.

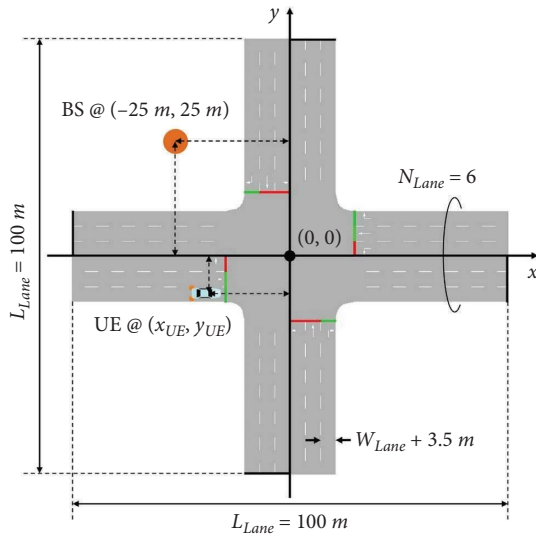


FIGURE 4: Urban intersection simulation environment with road parameters.

0.001 is applied. The performance of the proposed algorithm is evaluated over different hyperparameters of the DNN as well as operation parameters as shown in Figure 6. In Figure 6(a), the average correlation power  $\mu = \mathbb{E}[q_k]$  in (14) is shown for different values of  $J_1$  and  $J_2$ . As can be observed from the evaluation result, using  $J_1 = 3$  hidden layers with  $J_2 = 5$  nodes in each layer provides satisfactory performance while maintaining the computational complexity at a reasonable level. Therefore, these hyperparameter values are adopted for the performance evaluation in the remainder of the discussion. Figure 6(b) indicates the performance variations for different numbers of transmit antennas  $M_t$  and codebook size  $N$ . It is to be noted that the average correlation power tends to decrease as  $M_t$  increases. This is because the correlation power in (13) is normalized to have the maximum value of 1 and two vectors with more elements

results in smaller correlation in general. The evaluation result in the figure confirms the loss function convergence for antenna arrays with various sizes. Figure 6(b) also quantifies the amount of correlation power differences when the codebook size changes.

We assume that three bits are assigned for the PMI feedback for both horizontal and vertical codevectors, and the total of  $B = 6$  bits are reported to the BS for the 2D codebook of size  $N = 64$ . The saturation of the average CQI is shown in Figure 7, which compares the correlation power of the iteratively generated codebook with those of existing codebooks. The VQ codebook has been determined using the generalized Lloyd algorithm, based on sufficient UE location samples obtained from SUMO. The proposed algorithm not only outperforms the conventional codebooks as the iteration number increases but it possesses the capability of adaptively producing new codebooks even when traffic and environment conditions vary. The performance and convergence speed of the proposed algorithm does not depend heavily on the initial codebook types, as indicated by the figure for the cases of random and DFT initial codebooks. The figure also shows that near 100% correlation to the channel is achieved using the algorithm for the case of the single-path channel indicated by  $I_k = 1$ . The average correlation power tends to decrease for all codebooks as the number of propagation paths increases to  $I_k = 5$ . However, a noticeable amount of correlation gain is achieved by using the proposed codebook.

The computational complexity of the proposed algorithm can be compared with that of the conventional VQ codebook generation. The input vector to the DNN of dimension  $N \times 3$  is connected to the  $J_1$  hidden layers of  $J_2$  nodes, which are then connected to the output layer of dimension  $N \times 2$ . The overall number of weight multiplications required to produce an updated codebook is proportional to  $(N \times 3) \times J_2 + J_2^2 \times (J_1 - 1) + J_2 \times (N \times 2)$ , where the first two terms correspond to the computations at  $J_1$  hidden layers and the third term is for the output layer.

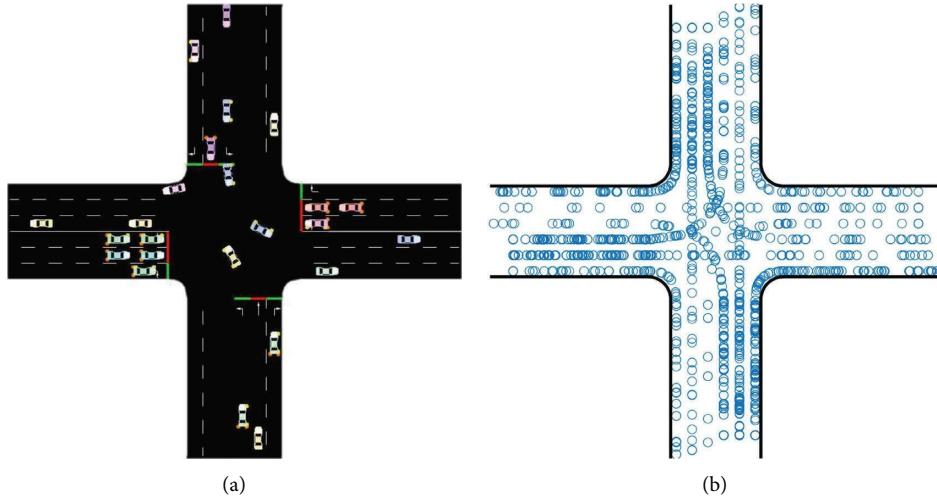


FIGURE 5: SUMO traffic generation: (a) mobile user distribution snapshot and (b) user locations for the PMI and CQI feedback over one iteration period.

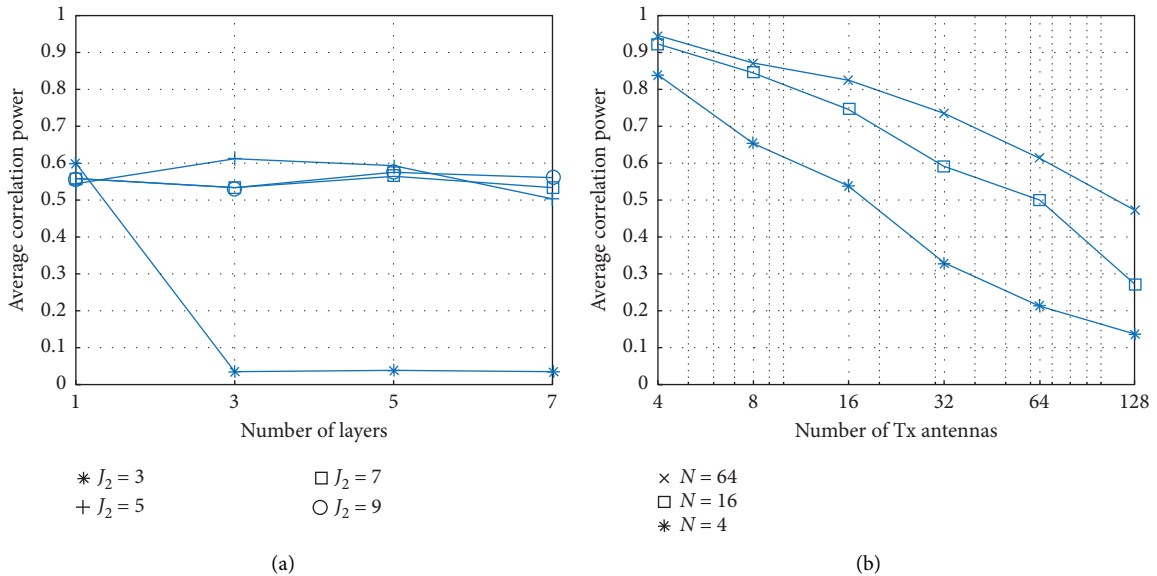


FIGURE 6: Average correlation power performance of the proposed DNN structure using (a) different numbers of layers and nodes and (b) different sizes of the codebook and transmit antenna array.

Thus, the computation of order  $\mathcal{O}(5NJ_2 + J_2^2(J_1 - 1))$  is needed to produce the output codevectors in each iteration. In our case of  $J_1 = 3$  hidden layers with  $J_2 = 5$  nodes, the computational order becomes  $\mathcal{O}(10N + 36)$ . For a sufficiently large codebook size satisfying  $N \gg \max(J_1, J_2)$ , the algorithm complexity can be approximated to  $\mathcal{O}(N)$ . On the other hand, the VQ codebook generation performs the comparison of the beam vector corresponding to each PMI sample in the dataset of size  $D$  to all existing  $N$  codevectors to form new clusters in each iteration, requiring the computation of  $\mathcal{O}(DN)$  order. In addition, the determination of updated centroids for new clusters performs the averaging of  $D$  beam vectors of dimension  $M_t$ , which requires the complex additions of order  $\mathcal{O}(DM_t)$ . The overall complexity order  $\mathcal{O}(DN) + \mathcal{O}(DM_t)$  for the VQ codebook generation

becomes significantly larger than that of the proposed algorithm, especially when a sufficient dataset of size  $D$  and a large array size  $M_t$  are in use. The proposed algorithm utilizes the input vector consisting of the accumulated PMI counts in integer numbers and the directional angle information instead of the beam vectors of dimension  $M_t$ ; therefore, the computational requirement mainly depends on the codebook size only.

Figure 8 visualizes the changing beam locations for the updated codebooks as the iteration number increases. Four subplots, respectively, show the beamforming directions projected on the ground after 0, 1, 10, and 100 iterations. The beam locations indicated by the cross marks in Figure 8(a) with no iterations is the initial DFT-based codebook with uniform spacing over both horizontal and vertical directions.

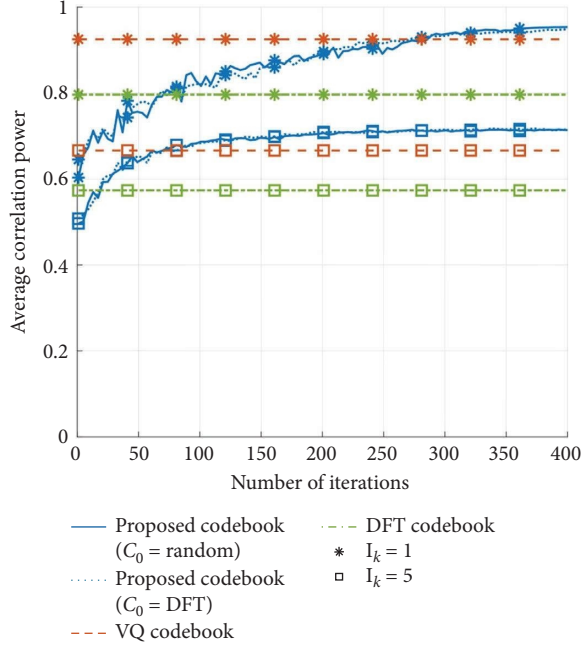


FIGURE 7: Average correlation power versus the number of iterations.

Codevectors are underutilized due to their concentration near the BS, and only a limited number of beams sparsely exist on the road surface which reduces the average correlation of codevectors to the UE channel. After a single iteration, a significant amount of beam location shifts occur to cover the busy traffic area on the road as indicated in Figure 8(b). The codevectors continue to evolve to increase the CQI experienced by the UEs as illustrated in Figure 8(c), and the codebook represented in Figure 8(d) outperforms the VQ codebook which requires a more complex generation procedure. To further quantify the angular gain of the proposed algorithm, let us define the angular distance  $\theta_d$  as the angle between the UE LoS vector and the closest codevector to it. The distribution of  $\theta_d$  is evaluated and shown in Figure 9, confirming the enhanced angular proximity of the proposed codebook. The average values of angular distance for the uniform DFT-based codebook, the VQ codebook, and the proposed codebook are, respectively,  $8.6^\circ$ ,  $4.6^\circ$ , and  $3.5^\circ$ . In terms of the chordal distance defined as  $\sin \theta_d$  [37], its average values for three codebooks are, respectively, 0.15, 0.08, and 0.06.

An important measure of transmission performance is the user sum-rate. To evaluate the sum-rate using the proposed codebook, we apply the ZF precoder for MU-MIMO transmission at the BS using the PMI reports  $\{p_1, \dots, p_K\}$  from  $K$  UEs. An estimated channel matrix of dimension  $K \times M$  obtained from these codevector indices can be written as

$$\tilde{\mathbf{H}} = [\mathbf{c}_{p_1} \mathbf{c}_{p_2} \dots \mathbf{c}_{p_K}]^H, \quad (18)$$

where  $\mathbf{c}_{p_k}$  is the codevector chosen by the  $k$ -th UE. The  $M \times K$  ZF precoder is then obtained as

$$\mathbf{W}_{\text{ZF}} = \tilde{\mathbf{H}}^H (\tilde{\mathbf{H}} \tilde{\mathbf{H}}^H)^{-1}. \quad (19)$$

The  $k$ -th column of ZF precoder matrix  $\mathbf{W}_{\text{ZF}}$  corresponds to beamforming vector  $\mathbf{w}_k$  in (1). Using ZF precoding, the sum-rate in (3) is evaluated for increasing values of  $K$  as shown in Figure 10. The sum-rate initially increases and then decreases as more UEs are scheduled for simultaneous transmission, since adding more user results in higher chances of producing an ill-conditioned channel matrix. The number of UEs maximizing the sum-rate approximately ranges from 5 to 10 depending on SNR  $\rho_k/(K\sigma^2)$  and codebook types. It can be observed from the figure that a substantial amount of gain in sum-rate can be achieved by the proposed codebook over the conventional ones for the entire range of UE numbers and SNR values. The channel includes  $I_k = 20$  paths for the evaluation results shown in Figures 7–10.

Performance comparison is extended to a wider class of channels in Figure 11, which shows the average correlation power for varying values of spread angle applied to different numbers of multipaths. The performance advantage of the proposed codebook becomes more obvious when the spread angle decreases, suggesting the scheme works particularly well for near-LoS channels such as the millimeter-wave propagation environments. However, the proposed codebook outperforms others over all ranges spread angles with substantial margins. Different numbers of multipaths are also tested and shown in the figure, demonstrating the usefulness of the proposal in a variety of channel conditions.

The performance comparison can be extended with some of the latest approaches. While the proposed algorithm generates the azimuth and zenith angles of codevectors, an alternative way of producing codevectors of the form

$$\mathbf{c} = \frac{1}{\sqrt{M_t}} [e^{j\theta_1}, e^{j\theta_2}, \dots, e^{j\theta_{M_t}}]^T, \quad (20)$$

is used to determine the phase values  $\theta_1, \theta_2, \dots, \theta_{M_t}$  as the DNN output as suggested in [23]. Such codevectors can be generated using the self-supervised learning algorithm and do not depend on specific array structures since arbitrary combinations of phase values can be produced. To obtain this type of “array-independent codebook,” the dimension of the DNN input vector increases to  $N \times (1 + M_t)$  to include the PMI counts and  $M_t$  phase values for each of  $N$  codevectors. Likewise, the output vector including the phase information of the updated codevectors has the increased dimension of  $N \times M_t$ . Another latest approach specified in 5G NR is the Type-II codebook [10]. The Type-II codebook enables the simultaneous transmission of multiple codevector beams for efficient coverage of multipath channels with a few significant reflecting clusters. The beamforming vectors for the Type-II codebook are determined by combining multiple PMI-reported codevectors with the highest CQI values.

In Figure 12, the average correlation power performance of the proposed codebook and other existing codebooks is evaluated and compared for channel conditions with varying



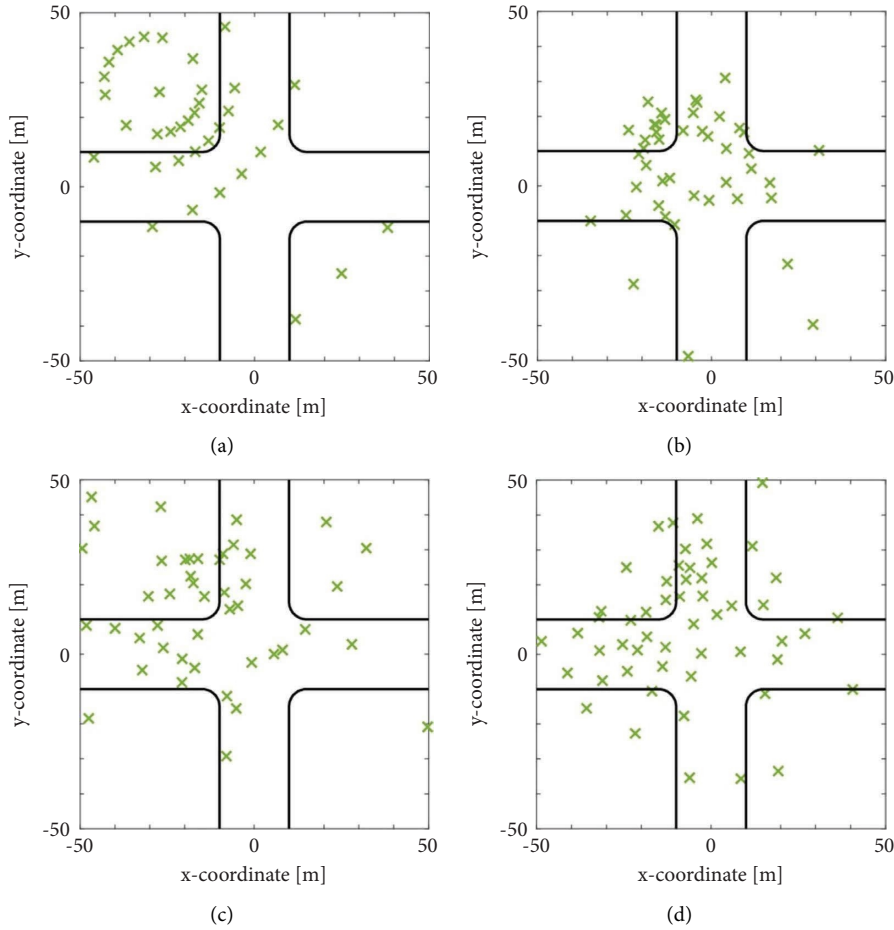


FIGURE 8: Beamforming directions of generated codevectors projected on the ground surface: (a) initial codebook, (b) after a single iteration, (c) after 10 iterations, and (d) after 100 iterations.

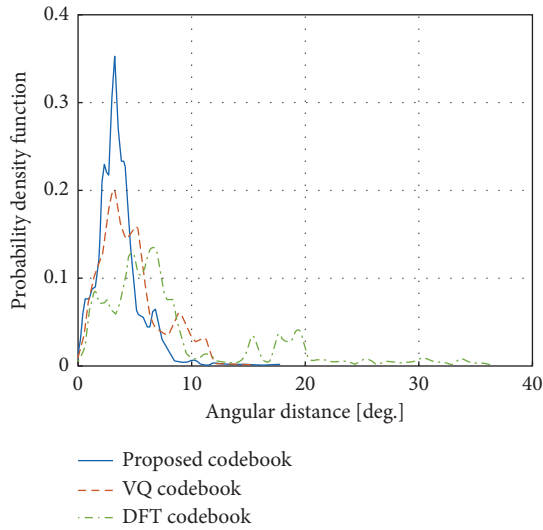


FIGURE 9: Probability density functions for the angular distance.

spread angles. The number of paths is assumed to be  $I_k = 10$ . As can be seen from the figure, the proposed codebook outperforms all existing codebooks in the entire range of channel conditions tested. The performance of the array-

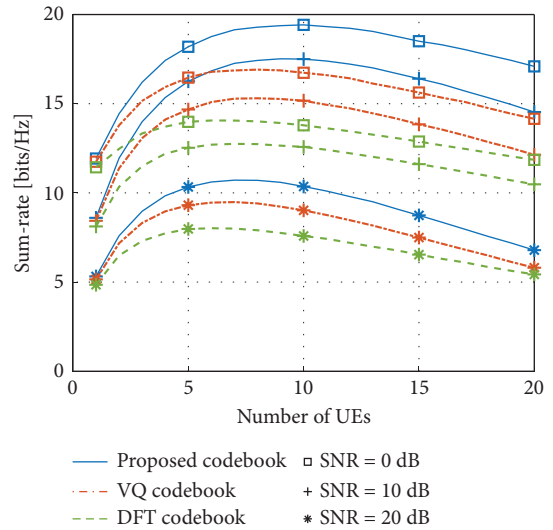


FIGURE 10: Sum-rate versus the transmission rank using ZF beamforming at different channel SNRs.

independent codebook is closed to that of the proposed codebook, especially for small values of the spread angle. Since the operational complexity of the array-independent

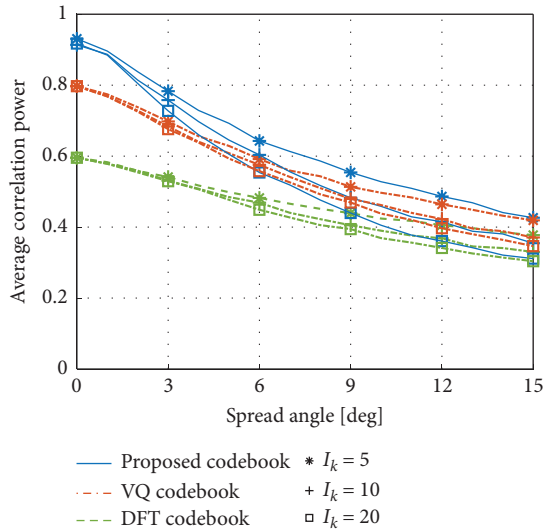


FIGURE 11: Performance comparison under channel parameter variations.

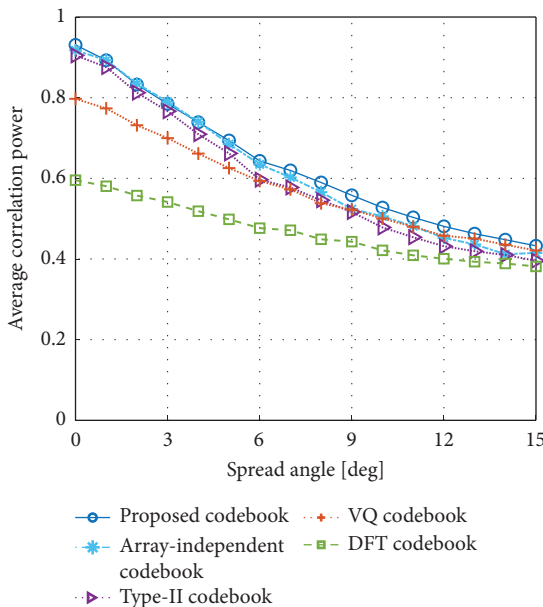


FIGURE 12: Correlation power performance comparison of the proposed and existing codebooks under channel conditions with varying spread angles.

codebook is proportional to  $O(NM_t)$ , the proposed codebook achieves the enhanced correlation performance at a reduced computational requirement. The Type-II codebook also exhibit similar performance to the proposed codebook. Nevertheless, there exists a clear performance gap between these two schemes which increases as the channel becomes more dispersive. Figure 13 shows the sum-rate performance comparison among the proposed and existing codebooks under the channel condition 10 multipaths and 10 dB SNR. As can be noted from the figure, latest MU-MIMO transmission strategies such as the array-independent codebook

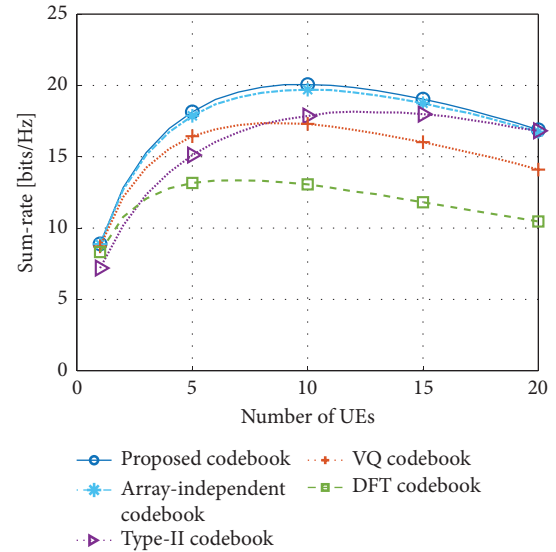


FIGURE 13: MU-MIMO sum-rate performance comparison of the proposed and existing codebooks for different numbers of users involved in simultaneous transmission.

and the Type-II codebook show improved throughput values over the VQ and DFT codebooks. However, the throughput evaluation result confirms that the proposed codebook outperforms all other codebooks regardless of the number of simultaneously transmitted users.

## 4. Conclusions

By combining the existing channel measurement reports with the DL algorithm, an autonomous codebook generation strategy at the BS is proposed and its performance advantage is verified. Unlike conventional codebooks, beamforming codevectors are adaptively relocated to enhance the coverage efficiency in a time-varying fashion, providing an improved sum-rate as well as increased received power. The proposed strategy can be extended to various channel conditions and environmental settings including the channels with blockage, multi-BS cooperation scenarios, and carrier frequency aggregations.

## Data Availability

The data used to support this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the National Research Foundation (NRF) of Korea, Ministry of Science, ICT and Future Planning (MSIP) under Grant NRF-2020R1A2C1004135, and in part by LG Electronics.

## References

- [1] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, 2016.
- [2] T. S. Xing, A. Kanhere, Ju Madanayake, Mandal Alkhateeb, Trichopoulos, and G. C. Trichopoulos, "Wireless communications and applications above 100 GHz: opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, June 2019.
- [3] S. H. Lim, S. Kim, B. Shim, and J. W. Choi, "Efficient beam training and sparse channel estimation for millimeter wave communications under mobility," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6583–6596, 2020.
- [4] A. N. Mahyuddin and N. M. Mahyuddin, "A comprehensive survey on millimeter wave communications for fifth-generation wireless networks: feasibility and challenges," *IEEE Access*, vol. 8, pp. 62367–62414, 2020.
- [5] R. C. Heath and R. W. Heath, "60 GHz wireless communications: emerging requirements and design recommendations," *IEEE Vehicular Technology Magazine*, vol. 2, no. 3, pp. 41–50, Sep. 2007.
- [6] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.
- [7] X. Li, "Hybrid precoding for massive mmWave MIMO systems," *IEEE Access*, vol. 7, pp. 33577–33586, 2019.
- [8] H. Yu, P. Guan, W. Qu, and Y. Zhao, "An improved beam training scheme under hierarchical codebook," *IEEE Access*, vol. 8, pp. 53627–53635, 2020.
- [9] 3GPP TS 38.213, "Physical Layer Procedures (Release 16). V16.4.0," Technical Specification Group Radio Access Network; NR, 2020.
- [10] 3GPP TS 38.214, V16.4.0, NR, "Physical Layer Procedures for Data (Release 16). V16.4.0," Technical Specification Group Radio Access Network; NR, 2020.
- [11] D. J. Love, R. W. Heath, and T. Strohmer, "Grassmannian beamforming for multiple-input multiple-output wireless systems," *IEEE Transactions on Information Theory*, vol. 49, pp. 2735–2747, 2003.
- [12] P. Xia and G. B. Giannakis, "Design and analysis of transmit beamforming based on limited-rate feedback," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1853–1863, 2006.
- [13] C. K. Au-Yeung and D. J. Love, "On the performance of random vector quantization limited feedback beamforming in a MISO system," *IEEE Transactions on Wireless Communications*, vol. 6, no. 2, pp. 458–462, 2007.
- [14] Y. T. Wu, Y. Y. Zhao, and F. Yu, "Comparison of codebooks for beamforming in limited feedback MIMO systems," in *Proceedings of the . Int. Conf. On Computer Science and Automation Engineering (CSAE)*, pp. 32–36, IEEE, Zhangjiajie, China, May 2012.
- [15] B. M. Hochwald, T. Marzetta, T. Richardson, W. Sweldens, and R. Urbanke, "Systematic design of unitary space-time constellations," *IEEE Transactions on Information Theory*, vol. 46, pp. 1962–1973, 2000.
- [16] D. Yang, L.-L. Yang, and L. Hanzo, "DFT-based beamforming weight vector codebook design for spatially correlated channels in the unitary precoding aided multiuser downlink," vol. 1, pp. 1–5, in *Proceedings of the IEEE Int. Conf. Communication*, vol. 1, IEEE, Cape Town, South Africa, May 2010.
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [18] H. Huang, "Deep learning for physical-layer 5G wireless techniques: opportunities, challenges and solutions," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 214–222, 2020.
- [19] F. Restuccia and T. Melodia, "Deep learning at the physical layer: system challenges and applications to 5G and beyond," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 58–64, 2020.
- [20] H. P. Tauqir and A. Habib, "Deep learning based beam allocation in switched-beam multiuser massive MIMO systems," in *Proceedings of the 2019 Second International Conference on Latest Trends in Electrical Engineering and Computing Technologies (INTELLECT)*, pp. 1–5, IEEE, Karachi, Pakistan, November 2019.
- [21] C. Qi, Y. Wang, and G. Y. Li, "Deep learning for beam training in millimeter wave massive MIMO systems," *IEEE Transactions on Wireless Communications*, 2020.
- [22] Y. Zhang, M. Alrabeiah, and A. Alkhateeb, "Learning beam codebooks with neural networks: towards environment-aware mmWave MIMO," in *Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, IEEE, Atlanta, GA, USA, February 2020.
- [23] M. Alrabeiah, Y. Zhang, and A. Alkhateeb, "Neural networks based beam codebooks: learning mmWave massive MIMO beams that adapt to deployment and hardware," *IEEE Transactions on Communications*, vol. 70, no. 6, pp. 3818–3833, 2022.
- [24] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1866–1880, 2020.
- [25] M. Chizhik, "Beamforming learning for mmWave communication: theory and experimental validation," Article ID 12406, 2019.
- [26] A. A. M. Saleh and R. A. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE Journal on Selected Areas in Communications*, vol. 5, no. 2, pp. 128–137, 1987.
- [27] O. E. Ayach, R. W. Heath, S. Abu-Surra, S. Rajagopal, and Z. Pi, "The capacity optimality of beam steering in large millimeter wave MIMO systems," in *Proceedings of the 2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC '19)*, pp. 100–104, IEEE, Cesme, Turkey, June 2012.
- [28] C. A. Balanis, *Antenna Theory: Analysis and Design*, Wiley, Hoboken, New Jersey, 2005.
- [29] Q. Nadeem, A. Kammoun, and M. Alouini, "Elevation beamforming with full dimension MIMO architectures in 5G systems: a tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3238–3273, 2019.
- [30] H. Ji, "Overview of full-dimension MIMO in LTE-advanced pro," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 176–184, 2017.
- [31] J. Sung and W. Sung, "Large-scale MIMO beamforming using successive channel state estimation and codebook extension," in *Proceedings of the 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 408–413, IEEE, London, UK, September 2013.
- [32] D. Sahoo, Q. Pham, J. Lu, and S. C. H. Hoi, "Online Deep Learning: Learning Deep Neural Networks on the Fly," 2017, <https://arxiv.org/abs/1711.03705>.

- [33] M. Contreras and E. Gamess, "Real-time counting of vehicles stopped at a traffic light using vehicular network technology," *IEEE Access*, vol. 8, pp. 135244–135263, 2020.
- [34] P. A. Lopez, "Microscopic traffic simulation using SUMO," in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582, IEEE, Maui, HI, November 2018.
- [35] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - simulation of urban MObility," *International Journal of Agile Systems and Management*, vol. 5, no. 3&4, pp. 128–138, 2012.
- [36] 3GPP TS 36.211, "Physical channels and modulation (release 16). V16.5.0," Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-TRA), 2021.
- [37] J. H. Conway, R. H. Hardin, and N. J. A. Sloane, "Packing lines, planes, etc.: packings in Grassmannian space," *Experimental Mathematics*, vol. 5, pp. 139–159, 1996.