Near-Field Beam Training Based on Deep Learning for Extremely Large-Scale MIMO

Guoli Jiang, *Graduate Student Member, IEEE*, and Chenhao Qi, *Senior Member, IEEE*

***Abstract*— Extremely large-scale multiple-input multiple-output (XL-MIMO) is considered as a key technology for future wireless communications. For near-field beam training in XL-MIMO, the distance information is another important factor aside of the angular information. The most recent work trains the deep neural network (DNN) by performing beam training based on far-field codebook that only contains angular information, where the optimal angle and distance are independently predicted. In this letter, the DNN is trained by performing beam training based on near-field codebook that contains both angular and distance information, where the optimal angle and distance are jointly predicted with improved performance. We first propose a deep learning-based near-field beam training (DNBT) scheme. To further improve the beam training performance, a DNBT with supplementary codewords (DNBT-SC) scheme is proposed, where the supplementary codewords are selected to perform beam training based on the probability vector acquired in the DNBT scheme. Simulation results show that under the same training overhead, the proposed schemes can achieve better performance than the existing schemes.**

***Index Terms*— Beam training, deep learning, near field, extremely large-scale MIMO (XL-MIMO).**

I. Introduction

FOR the sixth generation (6G) wireless communications, the extremely large-scale multiple-input multiple-output (XL-MIMO) with much more antennas than massive MIMO in the fifth generation (5G) wireless communications has been considered as a key enabling technology [1], [2]. The spectral efficiency can be significantly improved through MIMO beamforming based on the XL antenna array equipped at the base station (BS).

For 5G wireless communications, codebook-based beam training with its advantage of fast generating beams is widely adopted. Since traversing a codebook for massive MIMO is time-consuming, hierarchical codebook-based beam training is then proposed, where narrow beams are iteratively identified based on the trained wide beams to reduce the training overhead [3], [4], [5]. To exploit the channel coherence, some beam training schemes based on deep learning are proposed. In [6], the optimal beam is predicted by a deep neural network (DNN) based on the sampled narrow beams. Since the narrow beams can not cover the entire angle of interest, a DNN-based wide beam training scheme is proposed [7].

Manuscript received 23 May 2023; revised 17 June 2023; accepted 22 June 2023. Date of publication 26 June 2023; date of current version 12 August 2023. This work was supported in part by National Natural Science Foundation of China under Grant 62071116. The associate editor coordinating the review of this letter and approving it for publication was C. Gong. *(Corresponding author: Chenhao Qi.)*

The authors are with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China (e-mail: qch@seu.edu.cn).

Digital Object Identifier 10.1109/LCOMM.2023.3289513

With a significant increase of antenna aperture in XL-MIMO, the users are more likely to be located in the near-field region in XL-MIMO than in massive MIMO. Therefore, the beam training schemes for massive MIMO based on the far-field channel may not be applicable for XL-MIMO. In [8], a polar-domain representation of the near-filed channel with the corresponding codebook considering both the distance and angle is proposed. Compared with the angular-domain codebook of the far-field channel, the polar-domain codebook of the near-field channel contains much more candidate beams, which will lead to unacceptable overhead for beam training. In [9], a fast beam training scheme is proposed for the near-field channel, where the candidate angles are first determined by angular-domain beam sweeping based on a conventional far-field codebook and then the best effective distance is determined using a polar-domain codebook based on the obtained candidate angles. To reduce the overhead of angular-domain exhaustive search in beam sweeping, the far-field wide beams that only contain angular information are utilized to predict the optimal near-field beam based on deep learning [10]. However, generating wide beam codebook by powering off some antennas will cause power loss. Besides, predicting the optimal angle and distance independently may not achieve satisfactory performance compared to predicting them jointly. As a result, the near-field beam training for XL-MIMO is still a challenging topic.

Different from the existing works, in this letter, we train the DNN by performing beam training based on the near-field codebook that contains both angular and distance information, where the optimal angle and distance are jointly predicted instead of independently predicted. We first propose a deep learning-based near-field beam training (DNBT) scheme. To further improve the beam training performance, a DNBT with supplementary codewords (DNBT-SC) scheme is proposed, where the supplementary codewords are selected to perform beam training based on the output of the DNN in the DNBT scheme.

The notations in this letter are defined as follows. The symbols a and $\mathcal{A}$ define a scalar and a set, respectively. For a vector $\boldsymbol{a}$, $[\boldsymbol{a}]_i$ denotes the ith entry of $\boldsymbol{a}$. For a matrix $\boldsymbol{A}$, $[\boldsymbol{A}]_{m,n}$ denotes the (m,n)th entry of $\boldsymbol{A}$. $\boldsymbol{\mathcal{A}} = (\boldsymbol{a}_{m,n})_{M\times N}$ denotes a matrix of size $M \times N$ with each entry being a vector, where $\boldsymbol{a}_{m,n}$ denotes the (m,n)th entry of $\boldsymbol{\mathcal{A}}$. $\boldsymbol{I}_K$ denotes the identity matrix of size K. $(\cdot)^{\mathrm{T}}, (\cdot)^{\mathrm{H}}, |\cdot|$ denote the transpose, conjugate transpose and absolute value, respectively. $\lfloor\cdot\rfloor$ denotes the floor function. $\mathcal{CN}(0,\sigma^2)$ denotes the complex Gaussian distribution with zero mean and the variance being σ^2. $\mathcal{U}(a,b)$ denotes the uniform distribution between a and b. $\mathbb{R}$ and $\mathbb{C}$ define the sets of real number and

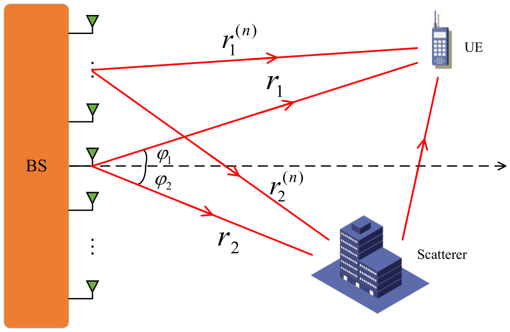

Fig. 1. Illustration of near-field channel between the BS and the UE.

complex number, respectively. $\emptyset$ defines the empty set. $\cup$ and $\cap$ denote the union and intersection of sets, respectively.

II. System Model

In this letter, we consider the downlink beam training for XL-MIMO. The base station (BS) is equipped with a N-antenna uniform linear array (ULA) and the antenna spacing is $d \triangleq \lambda_c/2$, where λ_c is the carrier wavelength. For simplicity, only one radio frequency (RF) chain is considered at the BS, which implies that we only need analog beamformer to perform the downlink beam training, and a single-antenna user equipment (UE) is considered. The received signal by the UE can be expressed as

$$y = \sqrt{P}\boldsymbol{h}\boldsymbol{v}x + \eta, \quad (1)$$

where $\boldsymbol{v} \in \mathbb{C}^N$ denotes the analog beamformer, $\eta \sim \mathcal{CN}(0, \sigma_n^2)$ denotes the channel noise, P denotes the transmit power of the BS, and x denotes the transmitted signal which is typically normalized in power, i.e., $|x|^2 = 1$. $\boldsymbol{h} \in \mathbb{C}^{1\times N}$ denotes the near-field downlink channel between the BS and the UE, and can be expressed as [8]

$$\boldsymbol{h} = \sqrt{\frac{N}{L}} \sum_{l=1}^{L} g_l e^{-jk_c r_l} \boldsymbol{b}^{\mathrm{H}}(\phi_l, r_l), \quad (2)$$

where $k_c \triangleq 2\pi/\lambda_c$ is the wavenumber corresponding to λ_c. L denotes the number of multipaths and $g_l \sim \mathcal{CN}(0, \sigma_l^2)$ denotes the channel gain of the lth path. $\phi_l \triangleq \sin\varphi_l$ denotes the spatial angle, where $\varphi_l \in [-\pi/2, \pi/2]$ denotes the physical angle. The near-field channel steering vector can be expressed as

$$\boldsymbol{b}(\phi_l, r_l) = \frac{1}{\sqrt{N}}[e^{-jk_c(r_l^{(1)}-r_l)}, \cdots, e^{-jk_c(r_l^{(N)}-r_l)}]^{\mathrm{T}}, \quad (3)$$

where r_l denotes the distance between the BS and the scatterer or the UE, and $r_l^{(n)}$ denotes the distance between the nth antenna of the BS and the scatterer or the UE for $n = 1, 2, \ldots, N$, as shown in Fig. 1.

The near-field channel steering vectors are sampled by some curves, which are termed as the distance rings [8]. The spatial angle and distance sampling can be expressed as

$$\phi_n = \frac{2n - N - 1}{N}, \quad n = 1, 2, \cdots, N, \quad (4)$$

$$r_{n,s} = \frac{N^2 d^2}{2s\beta_\Delta^2 \lambda_c}(1 - \phi_n^2), \quad s = 1, 2, \cdots, S, \quad (5)$$

respectively. S is the number of distance rings and β_Δ denotes the correlation of the near-field channel steering vectors. Based on the angle and distance sampling, the near-field codeword can be expressed as

$$\boldsymbol{w}_{n,s} = \boldsymbol{b}(\phi_n, r_{n,s}), \quad n = 1, 2, \cdots, N, s = 1, 2, \cdots, S. \quad (6)$$

Then we define a near-field codebook as $\boldsymbol{\mathcal{W}} \triangleq (\boldsymbol{w}_{n,s})_{N\times S}$, where $\boldsymbol{w}_{n,s}$ is the (n, s)th entry of $\boldsymbol{\mathcal{W}}$. The near-field codebook which considers both angular and distance information is two-dimensional (2D) and contains totally $Q \triangleq NS$ codewords. Note that different from the codebook generating method for wide beams by powering off some antennas [10], in this letter we generate the codebook with all the antennas powered on, which will not cause any power loss.

According to (1), we can obtain the achievable rate of the UE by

$$R = \log_2\left(1 + \frac{P|\boldsymbol{h}\boldsymbol{v}|^2}{\sigma_n^2}\right). \quad (7)$$

The beam training aims to identify the optimal codeword from $\boldsymbol{\mathcal{W}}$ to match the near-field channel with the largest achievable rate. In fact, the beam training can be formulated as

$$\boldsymbol{w}_{\widetilde{n},\widetilde{s}} = \arg \max_{\boldsymbol{w}_{n,s}\in\boldsymbol{\mathcal{W}}} \log_2\left(1 + \frac{P\left|\boldsymbol{h}\boldsymbol{w}_{n,s}\right|^2}{\sigma_n^2}\right), \quad (8)$$

where $\widetilde{n}$ and $\widetilde{s}$ denote the angle index and distance index of the optimal codeword, respectively.

The most intuitive method to identify the optimal codeword is beam sweeping which needs to traverse the entire codebook. However, the significant increase in the scale of antenna array in XL-MIMO makes the overhead of beam sweeping unacceptable. In addition, the distance information of the near-field channel leading to greater non-linearity also increases the difficulty of beam training. Since the DNN is good at dealing with nonlinear problem, in the next section, the DNN is used to preform beam training in the near-field domain to reduce the overhead.

III. Near-Field Beam Training Based on Deep Learning

A. Initial Codeword Selection

We initially select a small number of codewords from $\boldsymbol{\mathcal{W}}$ for beam training. For each distance ring, the codewords are selected uniformly with interval D. Therefore, the total number of initial codewords from $\boldsymbol{\mathcal{W}}$ is

$$E = TS, \quad (9)$$

where $T \triangleq \lfloor N/D \rfloor$ is the number of initial codewords for each s. The angle index of the selected codewords can be expressed as

$$n^{(s)} = n_0^{(s)} + kD \quad (10)$$

for $k = 0, 1, \cdots, T - 1$ and $s = 1, 2, \cdots, S$, where $n_0^{(s)} \in \{1, 2, \cdots, D\}$ denotes the initial angle index with respect to s.

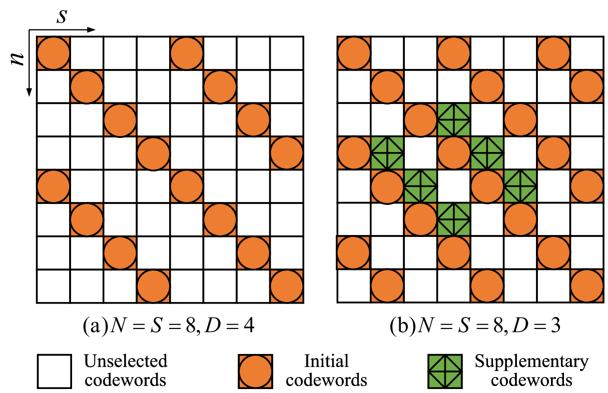

Fig. 2. The selected codewords in the near-field codebook $\boldsymbol{\mathcal{W}}$.

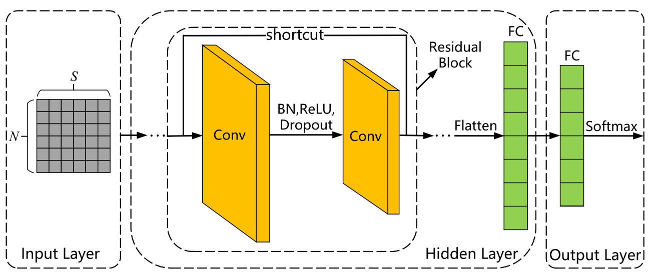

Fig. 3. The structure of the DNN for near-field beam training.

To maximize the coverage of selected codewords in the whole angular domain, there should be a certain offset between the initial angle indices of adjacent distance rings, i.e., $n_0^{(s+1)} = n_0^{(s)} + \Delta n$. For example, the initial codewords are shown in Fig. 2(a), where $N = S = 8$, $D = 4$ and $\Delta n = 1$. It can be seen that the initial codewords are evenly distributed in the entire near-field codebook.

B. The Structure of DNN

The structure of the DNN for beam training is illustrated in Fig. 3, where the DNN consists of an input layer, hidden layers and an output layer.

1) Input Layer: The codewords initially selected in the previous subsection are used for beam training and the absolute value of the received signal matrix fed from the UE to the BS is defined as $\boldsymbol{Y} \in \mathbb{R}^{N\times S}$, where the (n, s)th entry of $\boldsymbol{Y}$ can be expressed as

$$[\boldsymbol{Y}]_{n,s} = \begin{cases} |\sqrt{P}\boldsymbol{h}\boldsymbol{w}_{n,s}x + \eta|, & n = n^{(s)} \\ 0, & n \neq n^{(s)} \end{cases} \tag{11}$$

for $n = 1, 2, \cdots, N$ and $s = 1, 2, \cdots, S$.

To reduce the dispersion of data and the learning difficulty of the DNN, $\boldsymbol{Y}$ is normalized and then fed into the hidden layers.

2) Hidden Layers: For the highly nonlinear problem of near-field beam training, we increase the depth of the hidden layers to achieve better performance. In addition, the residual network, which introduces a shortcut connection, is adopted to solve the degradation problem in deeper network. The hidden layers consist of several residual blocks and a fully connected (FC) layer, where each residual block is composed of several convolutional (Conv) layers followed by a batch normalization (BN) layer, a ReLU activation layer and a dropout layer. The Conv layers and FC layer mainly act as classifiers. The role of the ReLU activation layer is to improve the nonlinear mapping capability of the DNN. The BN layers and dropout layers are used to speed up the convergence of the DNN and avoid overfitting. The multidimensional output of the residual block is flattened into one dimensional (1D) data to facilitate the processing of the FC layer.

3) Output Layer: We can obtain the output of the DNN by the $softmax$ function as

$$[\boldsymbol{p}]_k = \frac{e^{[\boldsymbol{\Gamma}]_k}}{\sum_{q=1}^{Q} e^{[\boldsymbol{\Gamma}]_q}} \tag{12}$$

for $k = 1, 2, \cdots, Q$, where $\boldsymbol{p} \in \mathbb{R}^Q$ denotes a probability vector and $\boldsymbol{\Gamma} \in \mathbb{R}^Q$ denotes the output of the hidden layers.

We can reshape $\boldsymbol{p}$ into a matrix $\boldsymbol{P} \in \mathbb{R}^{N\times S}$ by columns, where $[\boldsymbol{P}]_{n,s}$ denotes the probability that $\boldsymbol{w}_{n,s}$ in $\boldsymbol{\mathcal{W}}$ is the predicted optimal codeword.

C. Offline Training of DNN

Based on the structure of the DNN in Fig. 3, we adopt the supervised learning method to perform beam training. To obtain the training dataset and the labels, we randomly generate M near-field channel vectors $\boldsymbol{h}_m$, $m = 1, 2, \cdots, M$, according to (2) for offline training of the DNN, while $\boldsymbol{Y}_m$ as the input of the DNN can be obtained by (11). According to the simplified form of (8), the optimal angle index and distance index corresponding to $\boldsymbol{Y}_m$ can be identified as

$$(\widetilde{n}_m, \widetilde{s}_m) = \arg \max_{\substack{n\in\{1,2,\cdots,N\} \\ s\in\{1,2,\cdots,S\}}} |\boldsymbol{h}_m\boldsymbol{w}_{n,s}|, \tag{13}$$

which can be obtained by beam sweeping. Then, we convert $(\widetilde{n}_m, \widetilde{s}_m)$ into an 1D index as

$$\widetilde{q}_m \triangleq (\widetilde{s}_m - 1) N + \widetilde{n}_m. \tag{14}$$

The probability vector $\widetilde{\boldsymbol{p}}_m$ obtained by one-hot encoding of $\widetilde{q}_m$ is the output label corresponding to $\boldsymbol{Y}_m$, and can be expressed as

$$[\widetilde{\boldsymbol{p}}_m]_q = \begin{cases} 1, & q = \widetilde{q}_m, \\ 0, & q \neq \widetilde{q}_m. \end{cases} \tag{15}$$

Therefore, the near-field beam training problem is transformed into a multi-category problem, where each category corresponds to a codeword in $\boldsymbol{\mathcal{W}}$. The *cross-entropy* is adopted as the loss function of the DNN, and can be expressed as

$$f_{\text{LOSS}}(\boldsymbol{p}_m, \widetilde{\boldsymbol{p}}_m) = -\sum_{q=1}^{Q} [\boldsymbol{p}_m]_q \log_{10}[\widetilde{\boldsymbol{p}}_m]_q. \tag{16}$$

D. Beam Training Scheme Based on Deep Learning

Based on the DNN trained offline, we propose two schemes, including DNBT and DNBT-SC.

Algorithm 1 Deep Learning-Based Near-Field Beam Training (DNBT)

Input: $\mathcal{W}$.
Select initial codewords to obtain $\boldsymbol{Y}$ via (11).
Obtain $\boldsymbol{p}$ by feeding $\boldsymbol{Y}$ into the trained DNN.
Obtain $\widehat{q}$ via (17).
Determine $(\widehat{n}, \widehat{s})$ based on $\widehat{q}$ via (14).
Output: $\boldsymbol{w}_{\widehat{n},\widehat{s}}$.

1) DNBT: Straightforwardly, the DNBT scheme predicts the optimal codeword based on $\boldsymbol{p}$ in (12). The index of the largest entry of $\boldsymbol{p}$ can be expressed as

$$\widehat{q} = \arg \max_{q \in \{1,2\cdots,Q\}} [\boldsymbol{p}]_q. \quad (17)$$

Then we determine $(\widehat{n}, \widehat{s})$ based on $\widehat{q}$ according to (14). The predicted optimal codeword $\boldsymbol{w}_{\widehat{n},\widehat{s}}$ is the $(\widehat{n}, \widehat{s})$th entry of $\mathcal{W}$. In **Algorithm** 1, we present the detailed process of the DNBT scheme.

2) DNBT-SC: To further improve the accuracy of codeword prediction, beam training based on some supplementary codewords is considered in the DNBT-SC scheme.

Firstly, we define the set composed of 2D indices of initial codewords in (10) as

$$\mathcal{I}_{\text{init}} \triangleq \{(n_e, s_e), \quad e = 1, 2, \cdots, E\}, \quad (18)$$

where E is defined in (9). Correspondingly, the set composed of absolute value of the received signal corresponding to $\mathcal{I}_{\text{init}}$ can be expressed as

$$\mathcal{Y}_{\text{init}} \triangleq \{|y_e|, \quad e = 1, 2, \cdots, E\}, \quad (19)$$

where $y_e \triangleq \sqrt{P}\boldsymbol{h}\boldsymbol{w}_{n_e,s_e} x + \eta$.

Then, we define a matrix $\boldsymbol{P}_0$, which is the same as $\boldsymbol{P}$ except that the entries of $\boldsymbol{P}_0$ corresponding to $\mathcal{I}_{\text{init}}$ are all zero, as

$$[\boldsymbol{P}_0]_{n_e,s_e} = 0, \ e = 1, 2, \cdots, E. \quad (20)$$

The purpose of (20) is to ensure that the initial codewords will no longer be used for the additional beam training. The set composed of the 2D indices corresponding to the K largest entries of $\boldsymbol{P}_0$ can be defined as

$$\mathcal{I}_{\text{sup}} \triangleq \{(n_k, s_k), \ k = 1, 2, \cdots, K\}, \quad (21)$$

satisfying $\mathcal{I}_{\text{init}} \cap \mathcal{I}_{\text{sup}} = \emptyset$. We perform additional beam training using the supplementary codewords corresponding to $\mathcal{I}_{\text{sup}}$. For example, the supplementary codewords are illustrated in Fig. 2(b), where $K = 6$ and $\mathcal{I}_{\text{sup}} = \{(4, 2), (5, 3), (6, 4), (3, 4), (4, 5), (5, 6)\}$.

Similar to (19), the set composed of absolute value of the received signal corresponding to $\mathcal{I}_{\text{sup}}$ can be defined as

$$\mathcal{Y}_{\text{sup}} \triangleq \{|y_k|, \ k = 1, 2, \cdots, K\}. \quad (22)$$

We make a union for $\mathcal{I}_{\text{init}}$ and $\mathcal{I}_{\text{sup}}$, and have

$$\mathcal{I}_{\text{tot}} \triangleq \mathcal{I}_{\text{init}} \cup \mathcal{I}_{\text{sup}} = \{(n_g, s_g), \ g = 1, 2, \cdots, G\}, \quad (23)$$

Algorithm 2 Deep Learning-Based Near-Field Beam Training With Supplementary Codewords (DNBT-SC)

Input: $\mathcal{W}$, K.
Select initial codewords to obtain $\boldsymbol{Y}$ via (11).
Obtain $\boldsymbol{p}$ by feeding $\boldsymbol{Y}$ into the trained DNN.
Obtain $\boldsymbol{P}$ by reshaping $\boldsymbol{p}$ and determine $\boldsymbol{P}_0$ via (20).
Obtain $\mathcal{I}_{\text{init}}$ and $\mathcal{Y}_{\text{init}}$ via (18) and (19), respectively.
Obtain $\mathcal{I}_{\text{sup}}$ and $\mathcal{Y}_{\text{sup}}$ via (21) and (22), respectively.
Obtain $\mathcal{I}_{\text{tot}}$ and $\mathcal{Y}_{\text{tot}}$ via (23) and (24), respectively.
Determine $\widehat{g}$ via (25).
Output: $\boldsymbol{w}_{n_{\widehat{g}},s_{\widehat{g}}}$.

where $G \triangleq E + K$ denotes the total number of selected codewords and (n_g, s_g) denotes the 2D index of a selected codeword. We also make a union for $\mathcal{Y}_{\text{init}}$ and $\mathcal{Y}_{\text{sup}}$, and have

$$\mathcal{Y}_{\text{tot}} \triangleq \mathcal{Y}_{\text{init}} \cup \mathcal{Y}_{\text{sup}} = \{|y_g|, \ g = 1, 2, \cdots, G\}, \quad (24)$$

where y_g denotes the received signal by the UE corresponding to $\boldsymbol{w}_{n_g,s_g}$.

Finally, the index of the largest entry in $\mathcal{Y}_{\text{tot}}$ can be obtained by

$$\widehat{g} = \arg \max_{g \in \{1,2\cdots,G\}} |y_g|. \quad (25)$$

Therefore, the predicted optimal codeword is $\boldsymbol{w}_{n_{\widehat{g}},s_{\widehat{g}}}$. In **Algorithm** 2, we present the detailed process of the DNBT-SC scheme.

IV. Simulation Results

In this section, we evaluate the performance of the DNBT and DNBT-SC schemes. The downlink beam training for XL-MIMO system is considered. The number of channel paths is set as $L = 3$. For line-of-sight (LoS) path, the complex channel gain is set as $g_1 \sim \mathcal{CN}(0, 1)$. And for NLoS paths, the channel gain is set as $g_l \sim \mathcal{CN}(0, 0.01), l = 2, 3$. The number of antennas equipped at the BS is $N = 512$ and the carrier frequency is $f_c = 60$GHz. The angles of the UE or scatterer are randomly generated by $\varphi_l \sim \mathcal{U}(-60, 60)$. And the distances between the BS and the UE or scatterer are randomly generated by $r_l \sim \mathcal{U}(10, 60)$. For simplicity, the BS transmits the signal $x = 1$ with the power of $P = 1$. The number of distance rings of the near-field codebook is $S = 8$, so the total number of codewords is $I \triangleq NS = 4096$. The correlation of near-field channel steering vectors is set as $\beta_\Delta = 1.2$.

For the DNN, the hidden layers contain a total of 10 residual blocks. Each residual block contains 2 Conv layers with 2 and 8 kernels respectively, while the size of the convolution kernel is 3×3. The size of dataset is 50000, of which 80% is used for training and 20% for testing. The learning rate of the DNN is $\alpha = 0.001$. The number of epochs for training the DNN and the batch size are set as 100 and 1000, respectively.

The performance of the DNBT scheme and the DNBT-SC scheme is evaluated by two metrics. One metric is the achievable rate R defined in (7). The other metric is normalized gain, which can be defined as $z \triangleq |\boldsymbol{h}\boldsymbol{w}_{\widehat{n},\widehat{s}}|^2 / |\boldsymbol{h}\boldsymbol{w}_{\widetilde{n},\widetilde{s}}|^2$. $\boldsymbol{w}_{\widetilde{n},\widetilde{s}}$ and $\boldsymbol{w}_{\widehat{n},\widehat{s}}$ are the optimal codeword and the predicted optimal codeword, respectively.

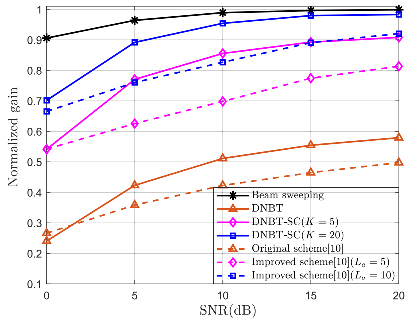

Fig. 4. Comparisons of the proposed schemes with the existing schemes in terms of normalized gain for different SNR.

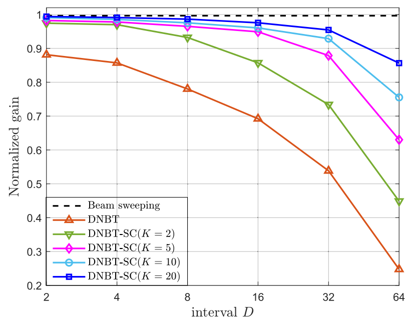

Fig. 5. Evaluation of the normalized gain for different intervals D.

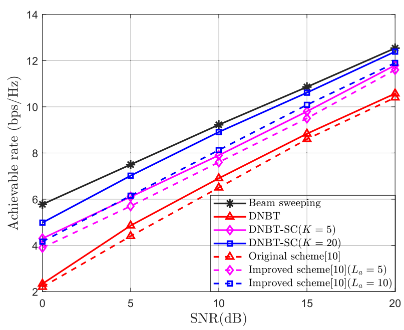

Fig. 6. Comparisons of the proposed schemes with the existing schemes in terms of achievable rate for different SNR.

In Fig. 4, we compare the DNBT and DNBT-SC schemes with the original and the improved schemes in [10] in terms of the normalized gain for different signal-to-noise ratio (SNR). The interval of codeword selection and the offset of initial angle index are set as $D = 32$ and $\Delta n = 4$, respectively. The number of supplementary codewords in the DNBT-SC scheme is set as $K = 5$ or 20. The simulation parameters of the existing schemes in [10] are set as $M = 128$ and $(L_a, L_d) = (5, 1)$ or $(10, 2)$, where M, L_a and L_d denote the overhead of initial training, additional training for angle and distance, respectively. From the figure, both DNBT and DNBT-SC can improve the normalized gain as K increases; and in particular, the latter is much better than the former because supplementary codewords are used by the latter. When the SNR is larger than 15dB, DNBT-SC with $K = 20$ can achieve over 95% of the normalized gain of beam sweeping while the training overhead is reduced by 96.4%. It is seen that the DNBT scheme outperforms the original scheme in [10], since the latter powers off some antennas to generate wide beams and causes power loss. Besides, DNBT-SC is better than the improved scheme in [10] when their training overhead is the same, because the latter predicts the optimal angle and distance independently but the former predicts them jointly. We can see that the DNBT-SC scheme with $K = 5$ can approach the performance of the improved scheme with $L_a = 10$, while the overhead of the former can be reduced by around $(148 - 133)/148 = 10\%$ compared to the latter.

In Fig. 5, we evaluate the normalized gain for different intervals D. It can be seen that the normalized gain will decrease with the increasing D, since less codewords are utilized for beam training. The DNBT-SC scheme outperforms the DNBT scheme, implying that the beam training using supplementary codewords can effectively improve the performance. When $D \leq 8$, the normalized gain of the DNBT-SC scheme has more than 12% performance improvement over the DNBT scheme.

In Fig. 6, we compare different schemes in terms of achievable rate for different SNR. The initial training overhead is set to be the same $E = 128$ for different schemes. It is seen that both DNBT and DNBT-SC scheme can achieve better performance than the schemes in [10]. As K increases, the achievable rate of the DNBT-SC scheme can approach that of beam sweeping.

V. Conclusion

In this letter, we have trained the DNN by performing the beam training based on near-field codebook that contains both angular and distance information, where the optimal angle and distance have been jointly predicted instead of independently predicted. We have proposed the DNBT and DNBT-SC schemes. Simulation results have shown that the proposed schemes can achieve better beam training performance than the existing schemes under the same overhead. In the future, we will continue our work with the focus on the multiuser scenario.

References

[1] M. Xiao et al., "Millimeter wave communications for future mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sep. 2017.

[2] T. S. Rappaport et al., "Wireless communications and applications above 100 GHz: Opportunities and challenges for 6G and beyond," *IEEE Access*, vol. 7, pp. 78729–78757, 2019.

[3] K. Chen, C. Qi, and G. Y. Li, "Two-step codeword design for millimeter wave massive MIMO systems with quantized phase shifters," *IEEE Trans. Signal Process.*, vol. 68, pp. 170–180, 2020.

[4] C. Qi, K. Chen, O. A. Dobre, and G. Y. Li, "Hierarchical codebook-based multiuser beam training for millimeter wave massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 8142–8152, Dec. 2020.

[5] Z. Xiao, T. He, P. Xia, and X. Xia, "Hierarchical codebook design for beamforming training in millimeter-wave communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[6] C. Qi, Y. Wang, and G. Y. Li, "Deep learning for beam training in millimeter wave massive MIMO systems," *IEEE Trans. Wireless Commun.*, early access, Sep. 22, 2020, doi: 10.1109/TWC.2020.3024279.

[7] K. Ma, D. He, H. Sun, Z. Wang, and S. Chen, "Deep learning assisted calibrated beam training for millimeter-wave communication systems," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6706–6721, Oct. 2021.

[8] M. Cui and L. Dai, "Channel estimation for extremely large-scale MIMO: Far-field or near-field?" *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2663–2677, Apr. 2022.

[9] Y. Zhang, X. Wu, and C. You, "Fast near-field beam training for extremely large-scale array," *IEEE Wireless Commun. Lett.*, vol. 11, no. 12, pp. 2625–2629, Dec. 2022.

[10] W. Liu, H. Ren, C. Pan, and J. Wang, "Deep learning based beam training for extremely large-scale massive MIMO in near-field domain," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 170–174, Jan. 2023.