Enhanced Channel Tracking in THz Beamspace Massive MIMO: A Deep CNN Approach

Navjot Kaur, Seyyed Saleh Hosseini, and Benoit Champagne
Dept. of Electrical and Computer Eng., McGill University, Montreal, Canada
E-mails: {navjot.kaur, seyyed.hosseini}@mail.mcgill.ca, benoit.champagne@mcgill.ca

***Abstract*—In this paper, we propose a novel model-driven deep learning approach to improve the performance of the channel tracking process in terahertz (THz) massive MIMO (m-MIMO) systems. Specifically, a recently introduced *a priori* aided (PA) channel tracking scheme which exploits the kinematics of the mobile users, is first used to obtain preliminary estimates of the THz m-MIMO channel. Then, a deep convolutional neural network (DCNN) based on the visual geometry group network architecture (VGGNet) is employed to refine these estimates, where the DCNN is trained offline to learn strong features of the non-linear map between PA-based channel estimates and the true channels. Simulation results demonstrate that the proposed DCNN-based approach significantly outperforms its traditional counterpart in terms of normalized mean square error. The resulting gains in accuracy can be traded to reduce the pilot overhead or required signal-to-noise ratio (SNR).**

***Index Terms*—Beamspace channel tracking, massive MIMO, DCNN, THz communications.**

I. Introduction

Massive multiple-input multiple-output (m-MIMO), one of the key enabling technologies for the fifth generation (5G) and beyond communications systems, offers unprecedented data transmission rates over the terahertz (THz) frequency band [1]–[4]. Similar to the mmWave m-MIMO systems, the key challenge lies in providing one radio frequency (RF) chain per antenna, which leads to high implementation costs and power consumption for a large-scale antenna system [5]. This burden can be efficiently alleviated by applying recently introduced techniques such as hybrid analog-digital processing [6] and beamspace m-MIMO [7].

The implementation of beamspace m-MIMO requires an accurate knowledge of the beamspace channels which can be obtained through a one-step (i.e., block) estimation method or a two-step approach consisting of an initial estimation followed by recursive tracking. Applying a one-step method to the fast time-varying THz beamspace channels necessitates a large pilot overhead to guarantee an effective beam selection. Accordingly, incorporating recursive tracking into the estimation process can significantly reduce the pilot overhead.

Recently, an *a priori* aided (PA) channel tracking scheme was proposed in [8] for THz beamspace m-MIMO systems. By exploiting kinematics of the mobile users along with temporal correlation of radio channels at THz frequencies, the PA scheme astutely utilizes the beamspace channel estimates from initial time slots to track the state of the channel in subsequent time slots. Afterwards, references [9] and [10] integrated a base-station (BS)-based cooperative scheme and a hierarchical multi-resolution codebook, respectively, along with the PA scheme so as to improve the accuracy of channel tracking. However, the improved accuracy is achieved at the additional cost of inter-BS coordination or exponentially increasing overhead of the codebook search.

Artificial intelligence (AI) methods have been recently applied to various wireless communication problems in order to enhance system performance and functionality [11], [12]. For instance, deep learning (DL) methods have been successfully applied to key problems of channel estimation and equalization [13]–[15]. The achieved improvements with DL-based approaches are largely due to their inherent capability to learn and model non-linear relationships in the observed data.

In this paper, we propose a novel model-driven DL-based approach to improve the performance of on-line channel tracking for THz beamspace m-MIMO systems. Specifically, the low-complexity PA channel tracking scheme is first used to obtain preliminary estimates of the m-MIMO channel. A deep convolutional neural network (DCNN) is then emploued to refine these estimates, where the DCNN is trained offline to learn strong features of the non-linear map between the PA-based channel estimates and the true channels. Simulation results show that the proposed DCNN-based approach can significantly outperform the original PA channel tracking method in terms of normalized mean square error (NMSE). The resulting gains in accuracy can be traded to reduce the pilot overhead or required signal-to-noise ratio (SNR).

II. System Model and Background

In this section, we first introduce the THz beamspace m-MIMO system model and then, briefly review the PA channel tracking scheme in [8].

A. System and Channel Model

Fig. 1 illustrates the use of a THz m-MIMO system with conventional (i.e., fully digital) structure for downlink transmission in a multi-user scenario. The base station (BS) employs N antennas and the same number of RF chains to simultaneously serve K single-antenna users. The received signal vector $\mathbf{y} \in \mathbb{C}^{K\times 1}$ by all K users can be written as,

$$\mathbf{y} = \mathbf{H}^H\mathbf{P}\mathbf{s} + \mathbf{n}, \tag{1}$$

where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K] \in \mathbb{C}^{N\times K}$ is the MIMO channel matrix, $\mathbf{h}_k \in \mathbb{C}^{N\times 1}$ is the channel vector between the BS and the kth user with $k \in \{1, \ldots, K\}$, $\mathbf{P} \in \mathbb{C}^{N\times K}$ is the precoding matrix satisfying $\text{tr}(\mathbf{P}\mathbf{P}^H) \leq \rho$ where ρ is the total transmit power, $\mathbf{s} \in \mathbb{C}^{K\times 1}$ represents the original signal vector for all K users with zero-mean and normalized covariance matrix $\mathbb{E}(\mathbf{s}\mathbf{s}^H) = \mathbf{I}_K$ and $\mathbf{n} \in \mathbb{C}^{K\times 1}$ is an additive white

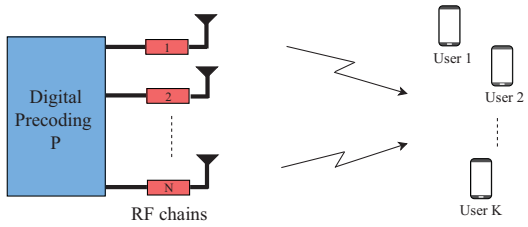

Fig. 1. Downlink conventional multiuser massive MIMO system.

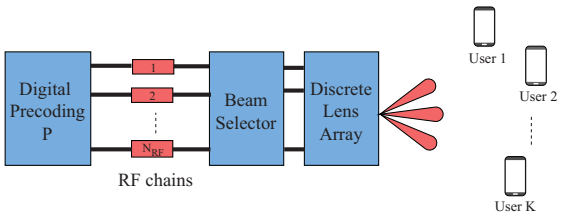

Fig. 2. Downlink multiuser beamspace massive MIMO system.

Gaussian noise vector with zero mean and covariance matrix $\sigma^2\mathbf{I}_K$. It is seen that for this system, the number of required RF chains is equal to the number of BS antennas, which for a massive setting (i.e., $N \gg 1$) entails high implementation costs as well as power dissipation.

We assume that the system operates in a time-division duplexing (TDD) mode, so that the reciprocity between uplink and downlink channels holds true. We consider the Saleh-Valenzuela channel model and first express the channel vector between the BS and the kth user as,

$$\mathbf{h}_k = \beta_k^{(0)}\mathbf{a}(\psi_k^{(0)}) + \sum_{i=1}^{L}\beta_k^{(i)}\mathbf{a}(\psi_k^{(i)}), \quad k = 1, \ldots, K \tag{2}$$

where $\beta_k^{(0)}\mathbf{a}(\psi_k^{(0)})$ is the line of sight (LOS) component, $\beta_k^{(i)}\mathbf{a}(\psi_k^{(i)})$ is the ith non-LOS (NLOS) component, and L is the total number of NLOS components. For the ith path, $\beta_k^{(i)}$ denotes the complex gain and $\mathbf{a}(\psi_k^{(i)}) \in \mathbb{C}^{N\times 1}$ is the array steering vector corresponding to the normalized spatial direction $\psi_k^{(i)}$. For a uniform linear array (ULA) configuration, the mth entry of $\mathbf{a}(\psi_k^{(i)})$ can be expressed as,

$$[\mathbf{a}(\psi_k^{(i)})]_m = \frac{1}{\sqrt{N}} e^{-j2\pi\psi_k^{(i)}(m-(N-1)/2)}, \quad m = 0, \ldots, N-1 \tag{3}$$

where $\psi_k^{(i)} = \frac{d}{\lambda}\sin\theta_k^{(i)}$, $\theta_k^{(i)}$ is the physical direction of departure of the ith path, λ is the wavelength at the carrier frequency, and d is the inter-element spacing in the ULA. In a THz channel, the NLOS attenuation is approximately 20dB stronger than that induced on the LOS path [16]. Hence, the LOS is the dominant component and proceeding as in [8]–[10], the NLOS components can be ignored. In the sequel, we remove the NLOS components from the model and drop the superscript (0) of the LOS component.

By employing a discrete lens array (DLA) at the BS, the spatial-domain channel of the K users in (2) can be transformed into sparse beamspace channels. In mathematical terms, the beamspace channel matrix resulting from this transformation can be written as [7],

$$\overline{\mathbf{H}} = [\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \ldots, \bar{\mathbf{h}}_K] \in \mathbb{C}^{N\times K}, \tag{4}$$

where $\bar{\mathbf{h}}_k = \mathbf{U}\mathbf{h}_k$ is the beamspace channel vector of the k-th user and $\mathbf{U} \in \mathbb{C}^{N\times N}$ is the spatial discrete Fourier transform (DFT) matrix. The latter is defined as

$$\mathbf{U} = [\mathbf{a}(\phi_0), \mathbf{a}(\phi_1), \ldots, \mathbf{a}(\phi_{N-1})]^H, \tag{5}$$

where $\phi_c = \frac{1}{N}(c - \frac{N-1}{2})$ for $c = 0, 1, \ldots, N-1$, are the spatial directions predefined by the DLA to cover the entire angular domain of interest. By multiplying matrices $\mathbf{U}$ and $\mathbf{H}$, the system model for THz beamspace m-MIMO transmissions can be expressed as,

$$\bar{\mathbf{y}} = \mathbf{H}^H\mathbf{U}^H\mathbf{P}\mathbf{s} + \mathbf{n} = \overline{\mathbf{H}}^H\mathbf{P}\mathbf{s} + \mathbf{n}, \tag{6}$$

where $\bar{\mathbf{y}} \in \mathbb{C}^{K\times 1}$ is the received signal vector in the beamspace m-MIMO system, and $\overline{\mathbf{H}} \triangleq \mathbf{U}\mathbf{H} = [\bar{\mathbf{h}}_1, \bar{\mathbf{h}}_2, \ldots, \bar{\mathbf{h}}_K] \in \mathbb{C}^{N\times K}$ is the THz beamspace channel matrix, with $\bar{\mathbf{h}}_k \in \mathbb{C}^{N\times 1}$ representing the beamspace channel vector between the BS and kth user. The downlink operation of the m-MIMO system is illustrated in Fig. 2.

Since the LOS is the predominant mode of propagation, the THz channel enjoys a sparse structure in the angular domain. Hence, a subset of $V \ll N$ beams with large magnitude can be selected from sparse beamspace channel vectors so as to decrease the dimension of the m-MIMO system [8]. Doing so, we can reformulate (6) as,

$$\bar{\mathbf{y}} = \overline{\mathbf{H}}_r^H\mathbf{P}_r\mathbf{s} + \mathbf{n}, \tag{7}$$

where $\overline{\mathbf{H}}_r^H \triangleq [\overline{\mathbf{H}}(r_1,:)^H\ \overline{\mathbf{H}}(r_2,:)^H\ \ldots\ \overline{\mathbf{H}}(r_V,:)^H]$, $r_i \in \{0, 1, \ldots, N-1\}$ is the index of the ith selected beam, and $\mathbf{P}_r \in \mathbb{C}^{V\times K}$ is the corresponding digital precoding matrix with reduced dimensions. Since the number of rows of $\mathbf{P}_r$ is much smaller than that of $\mathbf{P}$ in (1), the number of required RF chains can be reduced significantly. To guarantee the spatial multiplexing gain of the K users, the following condition should be satisfied, i.e., $K \le V \le N_{\text{RF}}$.

B. PA Channel Tracking

The PA channel tracking scheme in [6] exploits the underlying kinematics of mobile users to simplify the channel estimation and tracking. This is achieved in two phases: 1) conventional channel estimation within the initial time slots, followed by 2) channel tracking during subsequent time slots. In essence, the estimated beamspace channel vectors in the first phase, together with the equations describing the mobile user trajectories, are used to predict the *support*, i.e., the set of indices with non-zero entries, of the beamspace channel vectors in the second phase.

In the PA approach, the users are assumed to move according to a uniform linear motion model, i.e. with constant speed v_k along a straight line with direction φ_k as depicted in Fig. 3. The motion state vector of the kth user at the nth time slot, where the index $n \in \{0, 1, \ldots\}$, is then defined as $\mathbf{m}_k(n) = [\theta_k(n), \omega_k(n), \varphi_k]^T$ where $\theta_k(n)$ is the LOS direction of the user as seen from the BS, $\omega_k(n) = v_k/r_k(n)$ is related to the angular speed, and $r_k(n)$ is the distance between

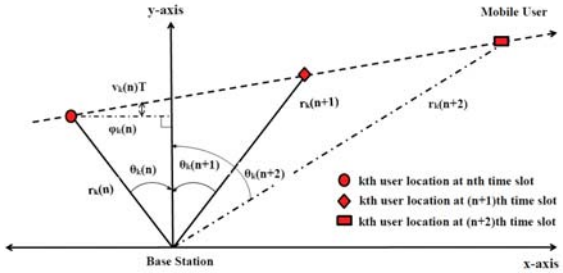

Fig. 3. Geometrical relationship between the BS and the kth user in the case of uniform linear motion.

the user and the BS. The following recursive relation satisfied by the motion state vector at two consecutive time slots can be derived by exploiting the uniform linear motion model:

$$\mathbf{m}_k(n+1) = F(\mathbf{m}_k(n)), \tag{8}$$

where $F(\mathbf{m}_k(n))$ is a mapping from $\mathbb{R}^3 \rightarrow \mathbb{R}^3$ such that for any state vector $\mathbf{m}_k(n) \triangleq [\theta_k(n), \omega_k(n), \varphi_k]^T \in \mathbb{R}^3$,

$$F(\mathbf{m}_k(n)) = \begin{bmatrix} f_1(n) \\ f_2(n) \\ \varphi_k \end{bmatrix}, \tag{9}$$

$$f_1(n) = \arctan\left(\frac{\sin\theta_k(n) + T\omega_k(n)\cos\varphi_k}{\cos\theta_k(n) + T\omega_k(n)\sin\varphi_k}\right), \tag{10}$$

$$f_2(n) = \frac{\omega_k(n)}{\sqrt{1 + 2T\omega_k(n)\sin(\theta_k(n)+\varphi_k) + T^2\omega_k^2(n)}}, \tag{11}$$

The quantities $\omega_k(n)$ and φ_k in $\mathbf{m}_k(n)$ can be reformulated in terms of the LOS directions at previous times as follows:

$$\omega_k(n) = \frac{\sin(\theta_k(n) - \theta_k(n-2))}{2T\cos(\theta_k(n-2)+\varphi_k)}, \tag{12}$$

$$\varphi_k = \frac{2a_k(n)\cos\theta_k(n) - b_k(n)\cos\theta_k(n-1)}{2a_k(n)\sin\theta_k(n) - b_k(n)\sin\theta_k(n-1)}, \tag{13}$$

where $a_k(n) \triangleq \sin(\theta_k(n-1) - \theta_k(n-2))$ and $b_k(n) \triangleq \sin(\theta_k(n) - \theta_k(n-2))$. Based on (8)-(13), it can be observed that once the LOS directions in time slots n, $(n-1)$ and $(n-2)$ have been determined, then the LOS directions in the next time slot $(n+1)$ can be predicted using (8) without channel estimation.

Based on the above relationships, the two phases of the PA channel tracking approach can be summarized as follows:

Phase 1 ($0 \leq n \leq 2$):

- The beamspace channels $\bar{\mathbf{h}}_k(n)$ are estimated by means of the orthogonal matching pursuit (OMP) method [17]. To this end, the users must send send a total of Q_{OMP} known pilots per time slot to the BS.
- The position $c_k^*(n)$ of the element of $\bar{\mathbf{h}}_k(n)$ with largest magnitude is found and used to approximate the physical direction $\theta_k(n)$:

$$\theta_k(n) \approx \sin^{-1}\frac{\lambda}{Nd}\left(c_k^*(n) - \frac{(N-1)}{2}\right). \tag{14}$$

Phase 2 ($n \geq 3$):

- $\theta_k(n)$ is predicted by using the values of $\theta_k(n-i)$ for $i = 1, 2, 3$ in the recursive relationships satisfied by the state vector $\mathbf{m}_k(n)$ for the case of uniform linear motion, i.e., (8)-(13).
- Using the prediction of $\theta_k(n)$, the support of $\bar{\mathbf{h}}_k(n)$ is obtained without channel estimation as follows:

$$c_k^*(n) = \underset{0\leq c\leq N-1}{\arg\min} \left|\phi_c - \frac{d}{\lambda}\sin\theta_k(n)\right|, \tag{15}$$

$$\mathcal{S}\{\bar{\mathbf{h}}_k(n)\} = \left\{c_k^*(n) - \tfrac{V}{2}, \ldots, c_k^*(n) + \tfrac{V}{2} - 1\right\}, \tag{16}$$

  where $\mathcal{S}\{\bar{\mathbf{h}}_k(n)\}$ with cardinality V (assumed even) denotes the support of $\bar{\mathbf{h}}_k(n)$.
- The non-zero elements of $\bar{\mathbf{h}}_k(n)$ are estimated by a least-squares approach. To this end, the kth user transmits V/K pilots to the BS and for each pilot transmission, the DLA of the BS selects K beams for estimation within the detected support. Hence, the total number of pilots per time slot during this phase is $Q_{\text{PA}} = (V/K)K = V$.
- The physical direction $\theta_k(n)$ is refined by finding the location $c_k^*(n)$ of the largest element of $\bar{\mathbf{h}}_k(n)$ and applying formula (14).

III. DCNN-Based Channel Tracking Approach

In this section, we describe the architecture and training mechanism of the proposed DCNN, which will be utilized to improve the performance of PA channel tracking in THz beamspace m-MIMO systems.

A. The DCNN Architecture

The DCNN refers to a well-known family of DL architectures which have achieved a high level of performance in several AI applications [18]. In this work, we employ the visual geometry group network (VGGNet) [24] as our main DCNN architecture. The latter comprises one input layer data, J hidden processing layers, and one output data layer (Fig. 4). In the training phase, the input to the DCNN consists of N_s data *samples*, each composed of preliminary estimates and true values of beamspace channel matrices. Specifically, the ith sample consists of a time sequence of channel matrix estimates obtained with the PA method, denoted as $\{\overline{\mathbf{G}}_i(n)\}_{n=0}^{T-1}$ where T is the number of time slots, along with the corresponding sequence of true channel matrices that the DCNN seeks as target, i.e., $\{\overline{\mathbf{H}}_i(n)\}_{n=0}^{T-1}$. In the online phase, the received signals at the BS are first input to a preliminary estimation module which tracks the beamspace channel vectors based on the PA method. Then, the estimated beamspace channel vectors are passed through the trained DCNN to enhance the accuracy of the estimated beamspace channel vectors.

The jth hidden processing layer of the DCNN, $1 \leq j \leq J$, consists of three parts: i) convolutional layer; ii) batch normalization (BN) module; and iii) activation function. We explain the operations performed by each module as follows:

- The convolutional layer extracts the local features from the data by applying η_{K} different kernels. Each kernel is a matrix of size $l \times l$ which slides over the input

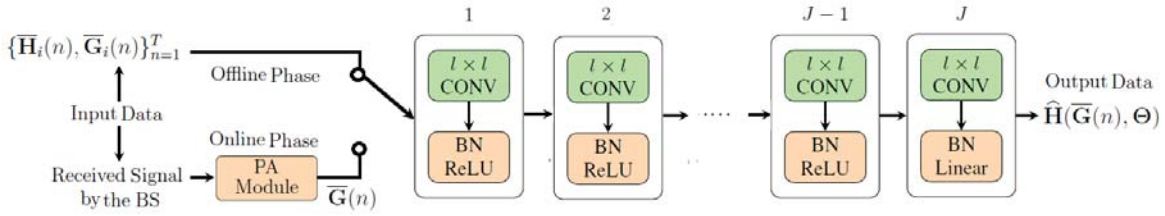

Fig. 4. Proposed DCNN architecture based on VGGNet for improving accuracy pf PA scheme.

data to perform the convolution operation. The kernel is swept over the entire data block (represented as an image) by sliding it in regular steps of a given size, called stride. Zero-padding is used to allow the application of the kernel over the boundary elements of the data block. The result of the convolution with each kernel is referred to as a feature map. All η_{K} feature maps are stacked along the third (depth) dimension and form the output of the convolutional layer.

- The BN is used to reduce the internal covariate shift of the feature maps [19], a common problem which affects the learning rate of the network and slows down the DCNN convergence [20]. Specifically, this module improves the performance and the convergence speed of the network by normalizing the mean and variance of each scalar feature to zero and one, respectively.
- The activation function is chosen as the rectified linear unit (ReLU) for layers $1 \leq j \leq J-1$ and as the linear function for layer $j = J$. The ReLU, which is defined as $\max(0, x)$ where x is the value of a scalar feature, introduces non-linearities in the DCNN structure to assist in the modeling and learning of more complex input/output relationships [21]. The linear function, simply defined as x, is utilized in the last hidden layer since the channel estimation is a regression task.

B. Training Mechanism of DCNN

The main objective of DCNN training is to find the best possible set of weights for the kernels in all hidden processing layers, as represented below by global parameter vector $\mathbf{\Theta}$. This can be achived in an offline fashion by iteratively minimizing the NMSE of the estimated beamspace channel vectors. For a set of N_s training samples, the NMSE metric can be expressed as

$$\mathcal{N}(\mathbf{\Theta}) = \frac{1}{TN_s} \sum_{i=1}^{N_s} \sum_{n=0}^{T-1} \frac{\|\overline{\mathbf{H}}_i(n) - \widehat{\mathbf{H}}(\overline{\mathbf{G}}_i(n), \mathbf{\Theta})\|_{\mathrm{F}}^2}{\|\overline{\mathbf{H}}_i(n)\|_{\mathrm{F}}^2}, \quad (17)$$

where $\widehat{\mathbf{H}}(\overline{\mathbf{G}}_i(n), \mathbf{\Theta}) \in \mathbb{C}^{N \times K}$ is the DCNN-based refined estimate of the true channel $\overline{\mathbf{H}}_i(n)$ and $\|.\|_{\mathrm{F}}$ stands for the Frobenius norm. In order to find $\mathbf{\Theta}$, an optimization problem with the objective function given in (17) must be solved. To this end, Adam optimizer is used with a learning rate which is adjusted adaptively based on the history of gradient descent updates [22]. The Adam optimizer uses the mini-batch gradient descent algorithm in which the N_s samples are first divided into *batches* with smaller number of samples and the average gradient of each batch is sequentially used to update the value of $\mathbf{\Theta}$ at each inner iteration. Once all inner iterations have been performed, an *epoch* is completed; hence, the total number of iterations is equal to the number of epochs times the number of batches.

The training process of a DCNN can be enhanced by adding and/or adjusting several training hyperparameters such as; the number of hidden layers, the size of the kernels, the length of a stride, the size of zero-padding, the type of optimizer, the learning rate, the batch size, and the number of epochs. A proper selection of hyperparameters can significantly improve the DCNN performance. For example, a kernel with small l extracts the fine-grained information and reduces the number of parameters to be learnt [23].

After offline training, the DCNN is deployed at the BS to track the THz channel in the online scenario with unseen data.

IV. Simulation Results

Herein, the performance of our proposed DCNN-based channel tracking approach for THz beamspace m-MIMO systems is evaluated via simulations and compared to the original PA scheme in [8]. We first explain the simulation methodology and then present selected results.

A. Methodology

As discussed earlier, the single-path Saleh-Valenzuela channel model describes adequately a practical THz m-MIMO channel. Hence, the training dataset is generated by simulations under the assumption that the channel model in (2) only consists of a LOS path. For the DCNN, the kernel size and the stride lengths are set to $l = 3$ and 1, respectively, while the number of feature maps is set to $\eta_{\mathrm{K}} = 64$. The batch size and the number of epochs are set to 32 and 200, respectively. The initial learning rate for the Adam optimizer is set to 10^{-3}.

The complete available dataset comprises $\sim 13 \times 10^4$ random samples for each one of the SNR dB value in the set $\{0, 5, 10, \ldots, 30\}$. This dataset is partitioned for three tasks, namely: i) training, ii) validation, and iii) testing. This division is essential for verifying and comparing the performance of a network with different parameters. Here, 80% of the samples are used for training, 10% for validation, and the remaining 10% for testing. We emphasize that the data used for testing is disjoint from the training set, as it is generated from unseen (independent) channel realizations.

A THz beamspace m-MIMO system is considered in which the BS is equipped with $N = 128$ antennas and $N_{\mathrm{RF}} = 4$ RF chains to serve simultaneously $K = 4$ users. We track the channel for $T = 50$ consecutive time slots. Similar to [8],

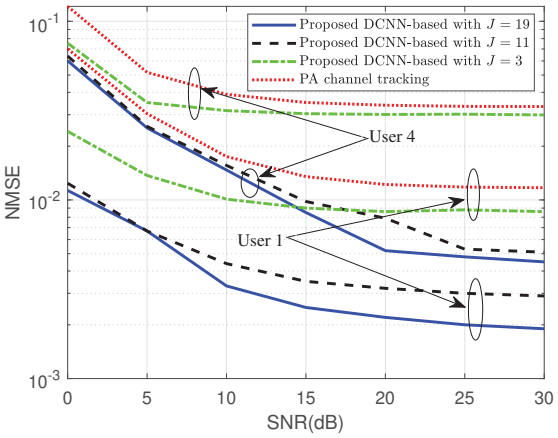

Fig. 5. NMSE versus SNR for the proposed DCNN-based channel tracking approach and the original PA scheme.

we select motion vectors of the first three users as $\mathbf{m}_1(1) = [\frac{\pi}{9}, 0.0154, \frac{3\pi}{4}]^T$, $\mathbf{m}_2(1) = [\frac{-\pi}{9}, 0.0071, \frac{\pi}{6}]^T$, and $\mathbf{m}_3(1) = [\frac{-2\pi}{9}, 0.0114, 0]^T$. For User 4, a nonlinear motion model is considered with $\mathbf{m}_4(1) = [0, 0.074, \frac{-3\pi}{4}]^T$ and $\mathbf{m}_4(16) = [\theta_4(16), \omega_4(16), 0]^T$. Except for Fig. 6, the number of pilots per time slot are set as $Q_{\text{OMP}} = 128$ and $Q_{\text{PA}} = V = 16$, respectively, where V is the length of the channel support.

B. Results and Discussion

Fig. 5 depicts the NMSE versus SNR for the proposed DCNN-based channel tracking approach and the original PA scheme in [8]. Here, we present the results related to User 1 and User 4 as benchmarks for performance comparison. Note that the results of the other two users are quite similar to that of User 1. As observed from the figure, the proposed approach significantly outperforms the PA scheme in terms of NMSE, especially when the number of processing layers J becomes larger. For User 1, the NMSE of the DCNN-based approach at SNR 30dB are 8.9×10^{-3}, 2.9×10^{-3}, and 1.9×10^{-3} for $J = 3$, $J = 11$ and $J = 19$, respectively, while the corresponding value of NMSE is 1.1×10^{-2} for the PA scheme. Similar observations also hold true for User 4, with nonlinear motion. Clearly, the performance of the DCNN-based channel tracking improves as the number of convolutional layers increases, since more complex and refined features can be learned from the training dataset.

It is worthwhile to mention that the main cost of the proposed method is imposed upon the training phase while the online phase entails negligible overhead. For the training phase, we compare the overhead components of the proposed DCNN, including training time, number of parameters, and the model size, versus the number of hidden processing layers J in Table I. Note that the model size refers to the size of the file which contains all the trained parameters after executing the training phase. As evidenced by Table I, the training overhead increases as expected with the number J of hidden layers.

Fig. 6 compares the NMSE performance versus the number of pilots per time slot for the proposed approach with $J = 19$ and the traditional PA channel tracking method, where the SNR is set to 10 dB. Clearly, increasing the number of pilots per time slot improves the NMSE performance of the THz channel tracking for the proposed DCNN-based and traditional PA methods alike. As seen from Fig. 6, to achieve the same accuracy in terms of the NMSE, the required number of pilots per time slot by the proposed DCNN-based approach is comparatively less than that of the PA scheme. For User 1, the PA scheme uses 128 pilots per slot to achieve the NMSE of 2×10^{-3} while the proposed DCNN-based scheme can achieve the same accuracy by using only 40 pilots per time slot. This advantage can be also observed for User 4.

TABLE I
TRAINING SPECIFICATIONS OF THE PROPOSED DCNN-BASED APPROACH

J	Training time (sec.)	No. of parameters	Model size (KB)
3	9.3×10^2	3.8×10^4	501
11	3.9×10^3	7.7×10^5	9260
19	6.6×10^3	1.3×10^6	16,343

Fig. 7 provides the NMSE versus SNR performance for the proposed DCNN-based channel tracking approach with different batch sizes and $J = 19$. As seen from the figure, the curves corresponding to the batch size of 32 provide the best NMSE performance from moderate to high SNR values. However, the DCNN-based approach exhibits a slightly lower NMSE with a batch size of 10 and 128 in low value of SNR. These results reveal that increasing the batch size does not necessarily lead to a better NMSE performance for the proposed approach. Moreover, it should be noted that the training time corresponding to the batch size of 10 is approximately twice that with the batch size of 32. Therefore, using batches with a moderate size such as 32 can be an efficient choice for the proposed DCNN-based channel tracking approach.

It is worth noting that the training dataset used for the DCNN-based approach was obtained with SNR dB values in the discrete set $\{0, 5, 10, \ldots, 30\}$. Nevertheless, our further results show that the proposed approach is capable of tracking the channel consistently at other SNR values, not included in the above set; hence, it is robust to the mismatch between the training and testing SNR values. This advantage can be interpreted as the strong power of the DCNN in learning the non-linear mapping between the preliminary channel estimates from the PA scheme and the true beamspace channels.

V. CONCLUSION

In this paper, we proposed a DCNN-based channel tracking approach for THz beamspace m-MIMO systems. The DCNN network is trained using a dataset comprising preliminary estimates of the beamspace channel matrix, as obtained from the PA channel tracking approach in [8] for different SNR levels. In application, the trained DCNN is used to refine the preliminary estimates from the PA method by extracting and modeling the strong features of the THz channel. Simulation results have shown that the proposed DCNN-based approach can significantly outperform the traditional PA approach in terms of NMSE performance. The improvement in estimation

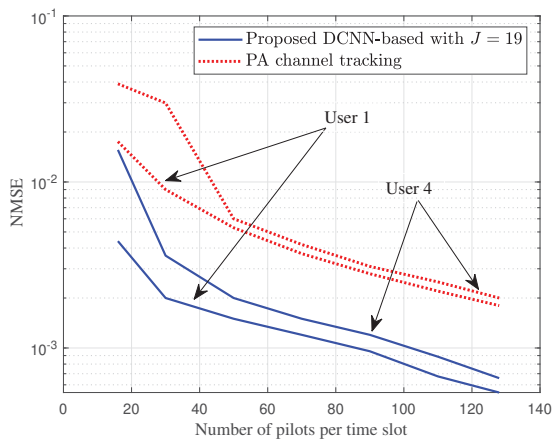

Fig. 6. NMSE versus number of pilots per time slot Q_{PA} for the proposed DCNN-based channel tracking approach and the PA scheme, with SNR= 10dB.

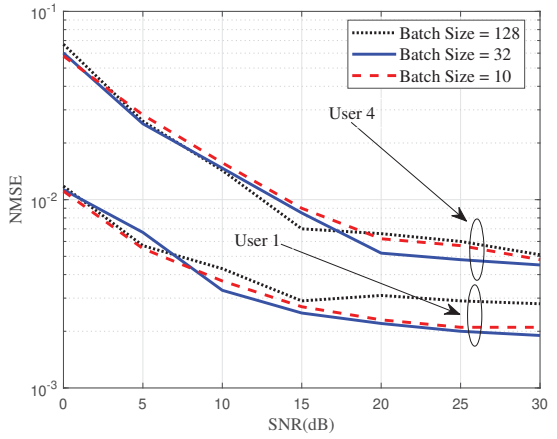

Fig. 7. NMSE versus SNR for the proposed DCNN-based channel tracking approach, with $J = 19$ and different batch sizes.

accuracy can be traded-off to reduce pilot overhead or required SNR in practice. An interesting avenue for future work will be to take the effects of NLOS multipath components into consideration when training and evaluating the proposed DCNN-based channel tracking approach for beamspace m-MIMO systems.

REFERENCES

[1] I. Akyildiz, J. M. Jornet, and C. Han, "TeraNets: Ultra-broadband communication networks in the terahertz band," *IEEE Wireless Commun.*, vol. 21, no. 4, pp. 130-135, Aug. 2014.

[2] C. Han, A. O. Bicen, and I. F. Akyildiz, "Multi-ray channel modeling and wideband characterization for wireless communications in the terahertz band," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2402-2412, May 2015.

[3] A. Morsali, S. S. Hosseini, B. Champagne, and X. Chang, "Design criteria for omnidirectional STBC in massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1435-1439, Oct. 2019.

[4] S. S. Hosseini, J. Abouei, and M. Uysal, "Fast-decodable MIMO HARQ systems," *IEEE Trans. Wireless Commun.*, vol. 14, no. 5, pp. 2827-2840, May 2015.

[5] X. Gao, L. Dai, Z. Chen, Z. Wang, and Z. Zhang, "Near-optimal beam selection for beamspace mmWave massive MIMO systems," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1054-1057, May 2016.

[6] F. Sohrabi and W. Yu, "Hybrid digital and analog beamforming design for large-scale antenna arrays," *IEEE J. Sel. Areas Commun.*, vol. 10, no. 3, pp. 501-513, Apr. 2016.

[7] J. Brady, N. Behdad, and A. M. Sayeed, "Beamspace MIMO for millimeter-Wave communications: System architecture, modeling, analysis, and measurements," *IEEE Trans. Antennas Propag.*, vol. 61, no. 7, pp. 3814-3827, July 2013.

[8] X. Gao, L. Dai, Y. Zhang, T. Xie, X. Dai, and Z. Wang, "Fast channel tracking for terahertz beamspace massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 5689-5696, July 2017.

[9] G. Stratidakis, A. A. Boulogeorgos, and A. Alexiou, "A cooperative localization-aided tracking algorithm for THz wireless systems," in Proc. 2019 *IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1-7.

[10] G. Stratidakis, G. D. Ntouni, A. A. Boulogeorgos, D. Kritharidis, and A. Alexiou, "A low-overhead hierarchical beam-tracking algorithm for THz wireless systems" in Proc. *Eur. Conf. Netw. Commun. (EUCNC)*, Virtual, June 2020, pp. 1-5.

[11] C. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G. Larsson, "Artificial intelligence enabled wireless networking for 5G and beyond: Recent advances and future challenges," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 16-23, Feb. 2020.

[12] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang,"Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175-183, Oct. 2017.

[13] X. Cheng, D. Liu, C. Wang, S. Yan, and Z. Zhu, "DL-based channel estimation and equalization scheme for FBMC/OQAM systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 881-884, June 2019.

[14] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. on Veh. Technol.*, vol. 68, no. 10, pp. 10325-10329, Oct. 2019.

[15] H. Sarieddeen, M.-S. Alouini, and T. Y. Al-Naffouri, "An overview of signal processing techniques for terahertz communications," May 2020, [Online]. Available: arXiv:2005.13176.

[16] C. Lin and G. Y. Li, "Indoor terahertz communications: How many antenna arrays are needed?," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3097-3107, June 2015.

[17] R. Méndez-Rial, C. Rusu, A. Alkhateeb, N. González-Prelcic, and R. W. Heath, "Channel estimation and hybrid combining for mmWave: Phase shifters or switches?," in Proc. *2015 Inf. Theory and Appl. Workshop (ITA)*, San Diego, CA, USA, Feb. 2015, pp. 90-97.

[18] Y. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127-138, Jan. 2017.

[19] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in Proc. *Eur. Conf. Comput. Vision (ECCV)*, Sept. 2018, pp. 262-277.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167 [cs.LG]*, Mar. 2015, pp. 1-11.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in Proc. *27th Int. Conf. Mach. Learn. (ICML-10)*, Haifa, Israel, June 2010, pp. 807-814.

[22] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in Proc. *3rd Int. Conf. Learn. Representations (ICLR 2015)*, San Diego, CA, USA, May 2015, pp. 1-15.

[23] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *arXiv:1901.06032 [cs.CV]*, 2019, pp. 1-67.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. *3rd Int. Conf. Learn. Representations (ICLR)*, San Diego, CA, USA, May 2015, pp. 1-14.

[25] P. V. Amadori and C. Masouros, "Low RF-complexity millimeter-wave beamspace-MIMO systems by beam selection," *IEEE Trans. Commun.*, vol. 63, no. 6, pp. 2212-2223, June 2015.