PRINCE: A Pruned AMP Integrated Deep CNN Method for Efficient Channel Estimation of Millimeter-Wave and Terahertz Ultra-Massive MIMO Systems

Zhengdong Hu, Yuhang Chen, and Chong Han, Member, IEEE

Abstract—Millimeter-wave (mmWave) and Terahertz (THz)-band communications exploit the abundant bandwidth to fulfill the increasing data rate demands of 6G wireless communications. To compensate for the high propagation loss with reduced hardware costs, ultra-massive multiple-input multiple-output (UM-MIMO) with a hybrid beamforming structure is a promising technology in the mmWave and THz bands. However, channel estimation (CE) is challenging for hybrid UM-MIMO systems, which requires recovering the high-dimensional channels from severely few channel observations. In this paper, a Pruned Approximate Message Passing (AMP) Integrated Deep Convolutional-neural-network (DCNN) CE (PRINCE) method is firstly proposed, which enhances the estimation accuracy of the AMP method by appending a DCNN network. Moreover, by truncating the insignificant feature maps in the convolutional layers of the DCNN network, a pruning method including training with regularization, pruning and refining procedures is developed to reduce the network scale. Simulation results show that the PRINCE achieves a good trade-off between the CE accuracy and significantly low complexity, with normalized-mean-square-error (NMSE) of $-10$ dB at signal-to-noise-ratio (SNR) as 10 dB after eliminating 80% feature maps.

Index Terms—Millimeter-wave and terahertz communications, ultra-massive MIMO, channel estimation, pruned deep convolutional network.

I. INTRODUCTION

Millimeter-wave (mmWave) and Terahertz (THz)-band communications explore the potential of ultra-broad bandwidth to meet the demands of high data rates, which are considered as key technologies for 6G wireless communications [1], [2]. Nevertheless, the high spreading loss and molecular absorption in mmWave and THz-band severely restrict the communication distance [3]. By focusing the transmitted signals to achieve high beamforming gain, ultra-massive multiple-input multiple-output (UM-MIMO) is a promising technology to combat the distance limitation [4], [5], [6], [7], [8]. Moreover, UM-MIMO can exploit the spatially uncorrelated channels for spatial multiplexing and increased spectral efficiency [4], [5], [6], [7], [8]. In the UM-MIMO systems, hybrid architectures are widely adopted to reduce power consumption, which control a large number of antennas with a small number of RF chains [8]. To enable the beamforming design of the hybrid UM-MIMO wireless systems, the acquisition of accurate channel state information (CSI) is critical, which strongly depends on reliable and efficient channel estimation (CE).

However, CE for the mmWave and THz hybrid UM-MIMO systems is a challenging problem. First, due to the high dimension of antennas with few RF chains in the hybrid architectures, CE requires recovery of the high-dimensional channels based on the received signals severely compressed to the dimension of the number of RF-chains. Second, traditional CE methods cannot achieve satisfactory estimation accuracy without prior knowledge of channel statistics, which is usually unknown and hard to be accurately estimated. Moreover, with the unprecedented large number of antennas in the mmWave and THz UM-MIMO systems, the recently rising deep learning (DL) methods suffer from high computational complexity.

To this end, the efficient CE scheme with high estimation accuracy for mmWave and THz hybrid UM-MIMO systems is highly in demand.

A. Related Work

In the literature, CE schemes for MIMO systems mainly include the conventional methods [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35] and the emerging DL methods [36], [37], [38], [39], [40], [41], [42]. On one hand, the conventional methods can be further classified into on-grid and off-grid methods. The on-grid methods treat the angles of arrival and departure as taken from fixed grids, which include the
compressive sensing (CS) methods such as the orthogonal matching pursuit (OMP) algorithm\cite{9} and the approximate message passing (AMP) algorithm\cite{10}. Both OMP and AMP are able to exploit the sparsity of the channel with less pilot training overhead, especially for mmWave and THz channels. Moreover, the authors in\cite{11} propose a distributed grid matching pursuit (DGMP) algorithm by exploiting the angle-domain structured sparsity of mmWave channel. The authors in\cite{12} analyze a simultaneous weighted OMP (SW-OMP) algorithm in the frequency-selective mmWave system. The authors in\cite{13} further propose a polar-domain simultaneous OMP (P-SOMP) algorithm to efficiently estimate the near-field channel. In\cite{14}, a sparse message passing (SMP) algorithm is developed to detect the nonzero entries in sparse channel matrix iteratively. However, the performance of these solutions degrades considerably due to the grid-mismatch problem. The grid-mismatch problem is induced by the fact that the parameters in the physical field such as the angles of the propagation path, are continuous and do not fall into the predefined grid precisely. By comparison, the off-grid methods refine the grid resolution or discard the on-grid assumption to improve the accuracy. Particularly, the authors in\cite{15} and\cite{16} iteratively refine the grid to increase the grid resolution. The beamspace 2-D multiple signal classification (MUSIC)\cite{17} and dynamic array-of-subarrays (DAoSA)-MUSIC\cite{18} methods exploit the subspac by performing eigenvalue decomposition. In\cite{19}, the direction-of-arrival (DOA) information is estimated by using the 2-D discrete Fourier transform (2D-DFT), and the angle rotation technique is applied to enhance the estimation accuracy. However, these schemes achieve better estimation accuracy than the on-grid methods with the price of high computation complexity.

On the other hand, with the rapid development of DL tools for wireless communications, many DL-based methods are proposed for CE, which extract the inherent characteristics of the physical channel and can improve the performance of CE\cite{36},\cite{37},\cite{38},\cite{39},\cite{40},\cite{41},\cite{42}. The DL-based CE schemes can be further divided into two categories, namely, model-driven\cite{36},\cite{37},\cite{38} and data-driven\cite{39},\cite{40},\cite{41},\cite{42} methods. The model-driven methods construct the networks with the domain knowledge, usually through the process of unfolding the conventional iterative algorithms. The DL networks only need to learn the parameters required by the iterative algorithm, which are fast and efficient. The learned AMP (LAMP) is proposed in\cite{36} to unfold the AMP iterations into corresponding deep network layers, where the parameters of AMP can be learned and optimized through training with a large dataset. In\cite{37}, the learned denoising-based approximate message passing (LDAMP) substitutes the original shrinkage function in AMP by a denoising convolutional neural network (DnCNN). The complex-valued Gaussian mixture LAMP (GM-LAMP) based beamspace CE scheme is presented in\cite{38} to integrate the LAMP\cite{36} with a derived Gaussian mixture shrinkage function, which can fully utilizes the prior information of the beamspace channel. The AMP algorithm is chosen in all the aforementioned methods\cite{36},\cite{37},\cite{38} for its powerful capacity in solving the sparse recovery problem and low complexity, while the DL network largely enhances the estimation performance of the AMP algorithm. However, the performance of these model-driven methods is bounded by the underlying AMP algorithm.

By contrast, the data-driven methods\cite{39},\cite{40},\cite{41},\cite{42} train the black-box based networks with a large number of data. The authors in\cite{39} propose a deep neural network (DNN) trained as a black box without relying on the knowledge of channels statics for the doubly selective channels. The estimator in\cite{40} employs a customized DNN design. It is based on the deep image prior (DIP) network, which first denoises the received signal, and then applies the least-squares (LS) estimation. In\cite{41}, the DCNN network is trained with a large number of received signals to directly output the important parameters involved in the reconstruction of channels. A spatial-frequency convolutional neural network (SF-CNN) based CE is proposed in\cite{42} to exploit spatial and frequency correlations, by inputting the channel matrices of adjacent sub-carriers into CNN simultaneously. Compared to the model-driven methods, the data-driven networks are not bounded by the conventional algorithms and can achieve high accuracy with proper network design and datasets.

Nevertheless, current design of data-driven DL methods keeps pursuing the high estimation accuracy without considering the applicability in practical deployment. The size of all of the aforementioned networks can easily boom for converging to the best possible performance, which is inefficient and impractical. In addition, the DL network can fall into the trap of over-fitting problem with the increment of network scale, leading that the DL networks lack adaptability to different environments apart from the trained one. As a result, it is essential to explore an efficient DL network with small scale and wide applicability.

To improve the efficiency of the data-driven DL network, a network slimming method is developed recently\cite{43}, by pruning the insignificant output feature maps in convolutional (CV) layers. The network slimming method\cite{43} can substantially reduce the model size, run-time memory and computational operations with low network training overhead. Particularly, the feature maps in the CV layers encode the presence or absence, and degree of presence of the features they detect. However, not every feature map can detect the critical features of the network input, leading that there exists feature map redundancy. On one hand, the network slimming method in\cite{43} is useful in pruning the less important feature maps, so that the compact pruned network owns with smaller size. The computational complexity of the network is also reduced by the pruning method. On the other hand, since the significant feature maps are maintained after pruning, the attention of DL network is not distracted to unimportant features. This avoids the potential over-fitting problem and improves the robustness of the network. To the best of our knowledge, the pruning strategy, which is promising to solve the network redundancy problem, has not been used in the CE DL methods before.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
B. Contributions

In this paper, we first propose an AMP integrated DCNN network (AMP-DCNN), which involves with the DCNN network after the AMP algorithm. The CV layers in AMP-DCNN are used to excavate the features of the channel structure encoded in the feature maps, to achieve remarkable performance increment in estimation accuracy. Furthermore, a pruned AMP integrated DCNN CE (PRINCE) method is developed based on AMP-DCNN. By truncating the insignificant feature maps in CV layers, the proposed PRINCE method largely reduces the size of the DCNN network to achieve low complexity, with negligible CE performance degradation. The contributions of this paper are summarized as follows.

- **We propose an AMP-DCNN method for CE of the mmWave and THz UM-MIMO systems**, which exploits the benefits of traditional AMP CE method and the DL tool. In AMP-DCNN, AMP is first applied to obtain the coarse CE result, which can exploit the sparsity of channel with less pilot training overhead. Then, DCNN refines the results from AMP by further enhancing the estimation accuracy with its powerful learning ability.

- **We propose a PRINCE method by pruning the DCNN network in AMP-DCNN**, to significantly reduce the computational complexity. Through training with regularization, pruning and refining, the PRINCE prunes the insignificant feature maps in the CV layers of AMP-DCNN with flexible ratios. The network size after pruning is thereby remarkably decreased in PRINCE, compared to AMP-DCNN, which enables lower computational complexity, while the estimation accuracy maintains.

- **We exploit the two-level sparsity in the mmWave and Terahertz channel**, to improve the robustness and efficiency of the proposed PRINCE method. On the one hand, the channel estimation problem is formulated as a sparse recovery problem, which utilizes the sparsity of the channel matrix. On the other hand, the feature maps in the DCNN network, encoding the features of the channel matrix, are also sparse with redundancy, which are saved in the PRINCE method.

The rest of this paper is organized as follows. The system model of the mmWave and THz UM-MIMO systems and the CE problem are investigated in Sec. II. The AMP-DCNN method is elaborated in Sec. III, and the PRINCE method is illustrated in Sec. IV. Sec. V evaluates the performance of the proposed methods. Finally, the conclusion is drawn in Sec. VI.

**Notation:** $a$ is a scalar. $\mathbf{a}$ denotes a vector. $\mathbf{A}$ represents a matrix. $(\cdot)^T$ defines the transpose. $(\cdot)^*$ refers to the conjugate transpose. $E\{\cdot\}$ describes the expectation. $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_F$ represent the L1 norm, L2 norm and Frobenius norm, respectively. $\vec{\cdot}$ represents the vectorization of the matrix. $|\cdot|$ denotes the absolute value. $*$ represents the convolution operation. $\mathbf{I}_N$ defines an $N$ dimensional identity matrix. $\otimes$ refers to the Kronecker product.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first introduce the system model of the mmWave and THz UM-MIMO systems. Then, we formulate the CE problem.

A. System Model

We consider the wideband multi-carrier mmWave and THz UM-MIMO communication systems deploying hybrid precoding and combining architectures at the transmitter (Tx) and receiver (Rx) sides [12], respectively, as shown in Fig. 1. There are totally $K$ subcarriers, with $N_t$ transmit and $N_r$ receive antennas. $L_s$ and $L_c$ RF chains are deployed at Tx and Rx, respectively. The systems deploy $N_s$ data streams, satisfying $N_s \leq L_s < N_r$ and $N_s \leq L_c < N_r$ for the considered mmWave and THz hybrid UM-MIMO systems [12]. At Tx, the transmitted symbol vector $\mathbf{s}[k] \in \mathbb{C}^{N_s \times 1}$ for the $k^{th}$ subcarrier is precoded to obtain the transmitted signal $\mathbf{x}[k] \in \mathbb{C}^{N_r \times 1}$ as

$$\mathbf{x}[k] = \mathbf{F}_{RF}\mathbf{F}_{BB}[k]\mathbf{s}[k],$$

(1)

where $k = 0, \ldots, K - 1$ indexes the subcarriers. Moreover, $\mathbf{F}_{RF} \in \mathbb{C}^{N_r \times L_s}$ represents the analog precoder, which is implemented by the phase shifters. Therefore, each element in $\mathbf{F}_{RF}$ satisfies the constant module constraint, which can be illustrated as $F_{RF}[i, j] = \frac{1}{\sqrt{N_s}}e^{j\phi_{i,j}}$, where $f_{i,j} \in [0, 2\pi]$ refers to the phase shift value, $i, j$ denote the position of the element in $\mathbf{F}_{RF}$. It is worth noting that the analog precoder is frequency independent [44], and the beam squint effect can be neglected in the planar array and the considered spectral window [28], [45]. In addition, $\mathbf{F}_{BB}[k] \in \mathbb{C}^{L_c \times N_s}$ represents the baseband digital precoder, which varies for different subcarriers.

After passing through the channel, the analog and digital combiners $\mathbf{W}_{RF} \in \mathbb{C}^{N_r \times L_s}$ and $\mathbf{W}_{BB} \in \mathbb{C}^{L_c \times N_s}$ are applied to the received signal at Rx, to obtain the baseband received signal $\mathbf{y}[k] \in \mathbb{C}^{N_r \times 1}$ as

$$\mathbf{y}[k] = \mathbf{W}_{BB}[k]^*\mathbf{W}_{RF}\mathbf{H}[k]\mathbf{x}[k] + \mathbf{W}_{BB}[k]^*\mathbf{W}_{RF}\mathbf{n}[k],$$

(2)

where $\mathbf{H}[k] \in \mathbb{C}^{N_r \times N_s}$ denotes the frequency-domain channel matrix for the $k^{th}$ subcarrier, $\mathbf{n}[k] \sim \mathcal{CN}(0, \sigma^2\mathbf{I}_{N_s})$ represents the circularly symmetric complex Gaussian distributed additive noise vector, with noise power equals $\sigma^2$. Since the analog combining is implemented by phase shifters, each element in $\mathbf{W}_{RF}$ satisfies the constant module constraint. Furthermore, the digital combiner $\mathbf{W}_{BB}$ is also subcarrier dependent.

We consider a frequency-selective wideband channel model for the mmWave and THz UM-MIMO systems, with a delay tap length $N_c$ in the time domain. Specifically, the channel matrix of the $k^{th}$ subcarrier is described as

$$\mathbf{H}[k] = \sum_{d=0}^{N_c-1} \mathbf{H}_d e^{-j \frac{2\pi d \nu_k}{N_c}},$$

(3)

where $d = 0, \ldots, N_c - 1$ indexes the delay tap. The channel matrix for the $d^{th}$ delay tap $\mathbf{H}_d \in \mathbb{C}^{N_r \times N_t}$ is represented as

$$\mathbf{H}_d = \sqrt{\frac{N_t N_c}{L}} \sum_{l=1}^{L} \alpha_{l,rc}(dT_s - \tau_l) \mathbf{a}_R(\theta_{R,l}, \phi_{R,l})\mathbf{a}_T^*(\theta_{T,l}, \phi_{T,l}),$$

(4)
and $a$ denotes the complex gain of the $\tau$ sampling period, $\tau$ lowpass filters evaluated at the filters including the effects of pulse-shaping and other $L$.

Fig. 1. Hybrid analog-digital precoding and combining for mmWave and THz UM-MIMO systems.

Fig. 2. We consider that channel coherence time is much longer than the symbol duration, i.e., a static channel is summarized in Table I.

TABLE I

| Parameter                                      | Symbol | Unit |
|------------------------------------------------|--------|------|
| Number of antennas at Tx                      | $N_t$  | 1    |
| Number of antennas at Rx                      | $N_r$  | 1    |
| Number of transmitted data streams            | $N_s$  | 1    |
| Number of subcarriers                         | $K$    | 1    |
| Number of RF chains at Tx                     | $L_t$  | 1    |
| Number of RF chains at Rx                     | $L_r$  | 1    |
| Transmitted symbol at Tx                      | $s$    | 1    |
| Transmitted signal at Tx                      | $x$    | 1    |
| Received signal at Rx                         | $y$    | 1    |
| Analog precoder at Tx                         | $P_{RF}$| 1    |
| Digital precoder at Tx                        | $P_{FB}$| 1    |
| Training precoder at Tx                       | $P_{rt}$| 1    |
| Analog combiner at Rx                         | $W_{RF}$| 1    |
| Digital combiner at Rx                        | $W_{FB}$| 1    |
| Training combiner at Rx                       | $W_{tr}$| 1    |
| Noise vector at Rx                            | $n_e$  | 1    |
| Channel matrix                                | $H$    | 1    |
| Length of the delay tap length                | $N_c$  | 1    |
| Number of paths                               | $L$    | 1    |
| Complex gain of the $l$th path                | $\alpha_l$| 1    |
| Delay of the $l$th path                       | $\tau_l$| s    |
| Angles pair of departure for the $l$th path   | $(\theta_{T,l}, \phi_{T,l})$| rad |
| Angles pair of arrival for the $l$th path     | $(\theta_{R,l}, \phi_{R,l})$| rad |
| Array steering vector                        | $a$    | 1    |
| Number of training frames                     | $M$    | 1    |
| Measurement matrix                            | $\Phi$ | 1    |
| Dictionary matrix                             | $\Psi$ | 1    |
| Vectorized channel vector                     | $h$    | 1    |

where $L$ stands for the number of propagation paths, $\alpha_l$ denotes the complex gain of the $l$th path, $\rho_{(l)}(\tau)$ represents the filters including the effects of pulse-shaping and other lowpass filters evaluated at $\tau$. Moreover, $T_s$ refers to the sampling period, $\tau_l$ denotes the delay of the $l$th path. The array steering vectors for Rx and Tx are $a_{R}(\theta_{R,l}, \phi_{R,l}) \in \mathbb{C}^{N_r \times 1}$ and $a_{T}(\theta_{T,l}, \phi_{T,l}) \in \mathbb{C}^{N_t \times 1}$, respectively, in which $(\theta_{R,l}, \phi_{R,l})$ and $(\theta_{T,l}, \phi_{T,l})$ represent the angle of arrival (AoA) and angle of departure (AOD) pairs.

We consider uniform planar arrays (UPAs) with $N_x \times N_z$ antennas in the $x$-$z$ plane. The array steering vector $a(\theta, \phi)$ holding for both $a_{R}(\theta_{R,l}, \phi_{R,l})$ and $a_{T}(\theta_{T,l}, \phi_{T,l})$ is represented as

$$a(\theta, \phi) = \frac{1}{\sqrt{N_x N_z}} e^{-j \pi \sin \theta \cos \phi} \otimes e^{-j \pi \sin \phi}$$

(5)

where $n_x = [0, 1, \cdots, N_x]^T$ and $n_z = [0, 1, \cdots, N_z]^T$, $N_x$ and $N_z$ denote the number of antennas on x- and z-axis, respectively. The channel matrix $H_d$ in (4) can be represented as a more compact form as

$$H_d = A_R \Delta_d A_T^t,$$

(6)

where $A_R = [a_R(\theta_{R,1}, \phi_{R,1}), \cdots, a_R(\theta_{R,L}, \phi_{R,L})] \in \mathbb{C}^{N_r \times L}$, and $A_T = [a_T(\theta_{T,1}, \phi_{T,1}), \cdots, a_T(\theta_{T,L}, \phi_{T,L})] \in \mathbb{C}^{N_t \times L}$, denote array manifold matrices, which contain the array steering vectors for Rx and Tx, respectively. Moreover, $\Delta_d = \text{diag}(\alpha_1, \cdots, \alpha_L) \in \mathbb{C}^{L \times L}$ is a diagonal matrix containing the path gains.

The on-grid model approximates the channel using the extended virtual channel model \cite{46}. Specifically, by considering grid of size $G_{rx} \times G_{rz}$ for the AoA and grid of size $G_{tz} \times G_{gz}$ for the AoD, with $G_r = G_{rx} G_{rt}$ and $G_l = G_{tz} G_{gz}$, the channel matrix $H_d$ can be approximated as

$$H_d \approx \tilde{A}_R \Delta_v^u \tilde{A}_T^t,$$

(7)

where the dictionary matrices $\tilde{A}_R = [a_R(\tilde{\theta}_{R,1}, \tilde{\phi}_{R,1}), \cdots, a_R(\tilde{\theta}_{R,L}, \tilde{\phi}_{R,L})] \in \mathbb{C}^{N_r \times G_r}$ and $\tilde{A}_T = [a_T(\tilde{\theta}_{T,1}, \tilde{\phi}_{T,1}), \cdots, a_T(\tilde{\theta}_{T,L}, \tilde{\phi}_{T,L})] \in \mathbb{C}^{N_t \times G_t}$ contain the array steering vectors. Calculation of the grid points at Tx and Rx are similar. Particularly, at Tx, the grid points are obtained as $\sin \theta_{T,g_{rx}} \cos \phi_{R,g_{rz}} \in \{-1 + \frac{2}{G_{rx}}, \cdots, 1\}$, and $\sin \theta_{T,g_{rx}} \in \{-1 + \frac{2}{G_{rx}}, \cdots, 1\}$, where $G_{rx}$ and $G_{rz}$ denote the number of grid points of x- and z-axis at Rx, respectively, satisfying $G_r = G_{rx} G_{rt}$, $G_l = G_{tz} G_{gz}$. Moreover, $G_{tx_1} = 1, \cdots, G_{tx_L}$ and $g_{rz} = 1, \cdots, G_{rz}$. Additionally, $\Delta_v^u \in \mathbb{C}^{G_r \times G_t}$ forms a sparse matrix with the non-zero elements being the path gains of the quantized spatial frequencies. In this way, $H[k]$ in (3) can be expressed as

$$H[k] \approx \tilde{A}_R(\sum_{d=0}^{N_v-1} \Delta_v^u e^{-j \frac{2\pi kd}{N_v}}) \tilde{A}_T^t$$

(8a)

$$\approx \tilde{A}_R \Delta_v^u[k] \tilde{A}_T^t,$$

(8b)

where $\Delta_v^u[k] = \sum_{d=0}^{N_v-1} \Delta_v^u e^{-j \frac{2\pi kd}{N_v}} \in \mathbb{C}^{G_r \times G_t}$. The parameters of the mmWave and THz UM-MIMO systems are summarized in Table I.

B. CE Problem Formulation

The frame structure for channel estimation is depicted in Fig. 2. We consider that channel coherence time is much longer than the symbol duration, i.e., a static channel is maintained for multiple consecutive symbol durations over...
a time slot $T_{\text{slot}}$. This assumption is reasonable, since the symbol durations in the mmWave and THz frequency band are on the orders of microsecond [47] and picosecond [48], respectively. By contrast, the channel coherence time is on the order of millisecond in the THz band [49]. During the training process, the pilot signals containing $M$ consecutive symbols are transmitted from Tx, which are received to reconstruct the channel matrix. For the $m^{\text{th}}$ training symbol at the $k^{\text{th}}$ subcarrier, the received signal is given by

$$y^{(m)}[k] = W^{(m)}_{\text{tr}} H[k] F^{(m)}_{\text{ref}} s^{(m)}[k] + n^{(m)}_{\text{e}}[k],$$  \hspace{1cm} (9)$$

where $m = 1, \ldots, M$ indexes the training symbols. The training combiner $W^{(m)}_{\text{tr}} = W^{(m)}_{\text{RF}} W^{(m)}_{\text{BB}} \in \mathbb{C}^{L_r \times N_r}$ while $F^{(m)}_{\text{ref}} F^{(m)}_{\text{BB}} \in \mathbb{C}^{N_x \times L_t}$ denotes the training precoder. The transmitted symbol $s^{(m)}[k] \in \mathbb{C}^{N_r \times 1}$ satisfies the power constraint that $E\left\{ s^{(m)}[k] s^{(m)}[k]^* \right\} = \frac{P}{L_i} I_{N_r}$ with the total transmitted power $P$ and $N_s = L_i$ in the considered system. Moreover, $n^{(m)}_{\text{e}}[k] \in \mathbb{C}^{L_r \times 1}$ refers to the combined noise vector in the $m^{\text{th}}$ training frame.

To reduce the complexity of the system, we consider that $s^{(m)}[k]$ can be decomposed as $s^{(m)}[k] = q^{(m)}_{\text{tr}}(t^{(m)}[k])$, in which $q^{(m)}_{\text{tr}} \in \mathbb{C}^{N_r \times 1}$ is independent of frequency. The value of $t^{(m)}[k] \in \mathbb{C}$ varies among different subcarriers, and is pre-defined in the training codebook, which is known at both Tx and Rx. Moreover, to obtain a subcarrier independent measurement matrix, the transmitted symbol $t^{(m)}[k]$ can be inverted by multiplying $(t^{(m)}[k])^{-1}$. As a result, the received signal after the pre-processing is given by

$$y^{(m)}[k] = W^{(m)}_{\text{tr}} T^{(m)} H[k] F^{(m)}_{\text{ref}} q^{(m)}_{\text{tr}} + n^{(m)}_{\text{e}}[k],$$  \hspace{1cm} (10)$$

where $y^{(m)}[k]$ and $n^{(m)}_{\text{e}}[k]$ are both changed by multiplying $(t^{(m)}[k])^{-1}$ compared to (9). After inverting the transmitted symbol $t^{(m)}[k]$, the received signal can be vectorized to obtain

$$\text{vec}\{y^{(m)}[k]\} = (q^{(m)^T}_{\text{tr}} F^{(m)^T}_{\text{ref}} W^{(m)}_{\text{tr}}) \text{vec}\{H[k]\} + \text{vec}\{n^{(m)}_{\text{e}}[k]\},$$  \hspace{1cm} (11)$$

The measurement matrix $\Phi^{(m)}$ is defined as $\Phi^{(m)} = (q^{(m)^T}_{\text{tr}} F^{(m)^T}_{\text{ref}} W^{(m)}_{\text{tr}}) \in \mathbb{C}^{L_r \times N_r N_t}$. Moreover, according to (8b), the vectorized channel can be represented as $\text{vec}\{H[k]\} = (A_R \otimes A_g) \text{vec}\{\Delta^y[k]\}$. The dictionary matrix $\Psi$ is defined as $\Psi = (A_R \otimes A_g) \in \mathbb{C}^{N_r N_t \times G, G_t}$ and the sparse vector $h[k] = \text{vec}\{\Delta^y[k]\} \in \mathbb{C}^{G, G_t \times 1}$ contains the complex gains of the channel. Then, (11) can be expressed as

$$\text{vec}\{y^{(m)}[k]\} = \Phi^{(m)} \Psi h[k] + \text{vec}\{n^{(m)}_{\text{e}}[k]\}.$$  \hspace{1cm} (12)$$

During the $M$ training symbol durations, different precoders and combiners are used to compose different $\Phi^{(m)}$. By collecting the received signals as (11) together, we obtain the received signal model as

$$\begin{bmatrix} \text{vec}\{y^{(1)}[k]\} \\ \vdots \\ \text{vec}\{y^{(M)}[k]\} \end{bmatrix} = \Phi^{(1)} H[k] + \begin{bmatrix} \text{vec}\{n^{(1)}_{\text{e}}[k]\} \\ \vdots \\ \text{vec}\{n^{(M)}_{\text{e}}[k]\} \end{bmatrix},$$  \hspace{1cm} (13)$$

Finally, CE is to solve the sparse reconstruction problem that estimates the sparse vector $h[k]$ in (13), given by

$$\begin{align*}
\min_{\|h[k]\|_1} & \|h[k]\|_1 , \\
\text{subject to} & \|y[k] - \Phi^{(1)} h[k]\|_2^2 < \epsilon,
\end{align*}$$  \hspace{1cm} (14)$$

where $\|h[k]\|_1$ denotes the sum of the absolute values of elements in $h[k]$. Instead of minimizing the number of nonzero elements in $h[k]$ to exploit the sparsity, the $\|h[k]\|_1$ is chosen for simplifying the sparse recovery problem into a convex optimization problem [38], [50]. Moreover, $\epsilon$ is the parameter measuring the maximum estimation error, which is usually set as the estimated noise variance.

III. AMP-DCNN METHOD FOR CHANNEL ESTIMATION

In this section, we propose the AMP-DCNN method to solve the formulated CE problem. The proposed AMP-DCNN is composed of two parts, including the AMP [10] and DCNN, as shown in Fig. 3. The AMP is used to obtain the coarse channel estimation result from the received signal $y[k]$ and measurement matrix $\Phi$, while the DCNN refines the output of AMP algorithm to yield the final estimation result with high accuracy.

A. AMP Part

Due to the huge number of antennas in mmWave and THz UM-MIMO systems, the sparse signal recovery problem in (14) has very high dimension. Among the various sparse signal recovery algorithms, the iterative based AMP algorithm is attractive due to its low complexity and fast convergence rate for high-dimensional problems [10]. Specifically, the received signals $y[k]$ and the measurement matrix $\Phi$ are input to the algorithm to obtain the estimated result $H_{\text{AMP}}[k]$.

However, the estimation result of AMP is not promisingly accurate due to the grid-mismatch problem, i.e., the actual continuous physical parameters do not match the predefined grid of the parameter space. Moreover, the shrinkage parameter $\lambda_t$ takes the same predefined value for all the

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
iterations, which limits the performance of the AMP algorithm. Therefore, to improve the coarse estimated results from the AMP algorithm, we further introduce DCNN to refine the result.

B. DCNN Part

The network structure of DCNN network is illustrated in Fig. 3, where the green and orange parallelograms represent the input and output of the network, respectively. The blue arrows denote the convolution operations. In addition, the blue parallelograms stand for the feature maps in the CV layers. Overall, there are 13 layers in the DCNN architecture, including one input layer, eleven CV layers with 64 filters of the dimensions of 3C-3, and one CV layer with 2 filters of the size 3C-3, which is indeed the output layer. Moreover, every CV layer except the last one is followed by a batch normalization (BN) layer, and there are eleven BN layers in total.

To start with, the network input of the DCNN network is constructed with the estimated channel matrix \( \mathbf{H}_{\text{AMP}}[k] \in \mathbb{C}^{N_r \times N_t} \). Specifically, \( \mathbf{H}_{\text{AMP}}[k] \) is separated into the real part \( \text{Re}(\mathbf{H}_{\text{AMP}}[k]) \in \mathbb{N}^{N_r \times N_t} \) and the imaginary part \( \text{Im}(\mathbf{H}_{\text{AMP}}[k]) \in \mathbb{N}^{N_r \times N_t} \) to compose the two channels of the input layer. Followed by the input layer, there are twelve CV layers. Zero padding (ZP) is adopted in each CV layer to keep the dimension of feature map unchanged. Moreover, the rectified linear unit (ReLU) activation function is adopted in each CV layer except the last one, which determines the activated neurons in the network. In particular, the ReLU activation function is represented as

\[
    f_{\text{ReLU}}(x) = \max(0, x),
\]

which is commonly used in the hidden layers for its fast speed of convergence.

In the CV layer, each output feature map can be calculated by convolving its corresponding filter with the previous layer. Specifically, the value of the output neuron in the feature map is obtained by convolving the filter with a small region in the previous layer, which is known as the local receptive region. The convolution process for each neuron in the CV layer is given by

\[
    h_{cv,i,j}^{(m)} = f^{(m)}(u_{cv,i,j}^{(m)} \ast c_{cv,i,j}^{(m-1)} + b_{cv,i,j}^{(m)}),
\]

where \( f^{(m)}(\cdot) \) represents the activation function, which describes the nonlinear mapping relationship. Additionally, \( u_{cv,i,j}^{(m)} \) and \( b_{cv,i,j}^{(m)} \) denote the weight and the bias of the \( i^{th} \) filter, and \( c_{cv,i,j}^{(m-1)} \) stands for the local receptive field for the \( j^{th} \) neuron of the \( i^{th} \) channel in the \( (m-1)^{th} \) layer.

Every CV layer is followed by a BN layer except the output layer, which is used to expedite the convergence of neural networks. The BN performs standardization and normalization operations on the input batch of data, described as

\[
    \hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}},
\]

\[
    y = \gamma \hat{x} + \beta,
\]

where \( x \) and \( y \) are the input and output of batch normalization, the subscript \( B \) represents the current mini-batch, \( \mu_B \) and \( \sigma_B \) denote the mean and standard deviation of the mini-batch of data, \( \varepsilon \) is a small number to avoid dividing by zero. Moreover, \( \gamma \) and \( \beta \) represent trainable parameters to scale and shift the normalized value such that the loss of the neural network is minimized.

The last layer of the DCNN network is the output layer, which produces the estimated channel matrix. We do not adopt any activation function in the output layer of the proposed DCNN network, since the activation functions might restrict the range of the output value which is not pre-determined. For example, the output of ReLU activation function in (15) is in the range of \([0, +\infty)\), while the values of the elements in the target channel matrix can be less than zero and has no specific bound. By denoting the input channel matrix as \( \mathbf{H}_{\text{AMP}}[k] \) and the output matrix as \( \mathbf{H}_{\text{AMP}-\text{DCNN}}[k] \), the end-to-end relationship of the DCNN network is given by

\[
    \mathbf{H}_{\text{AMP}-\text{DCNN}}[k] = f^{(M)}(f^{(M-1)}(\cdots f^{(1)}(\mathbf{H}_{\text{AMP}}[k]))),
\]

where \( M \) denotes the number of layers in the DCNN network, \( f^{(M)} \) refers to the transformation of the \( M^{th} \) layer. During the training process, the mean squared error (MSE) loss function denoted by \( l_{\text{MSE}} \) is deployed, which is defined as

\[
    l_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\mathbf{H}}^i - \mathbf{H}^i \right\|_F^2,
\]

where \( N \) denotes the size of the considered dataset of the channel matrix, and \( \hat{\mathbf{H}}^i \) is the \( i^{th} \) estimated channel matrix. Since the real channel data for mmWave and THz UM-MIMO is currently unavailable due to the hardware constraints of implementing large antenna arrays, the dataset of \( \mathbf{H}^i \) is generated by Wireless Insite [51] as described in Sec. V-A. Moreover, \( \mathbf{H}^i \) is the \( i^{th} \) estimated channel matrix through the DCNN network.

Since the DCNN network is designed to minimize the MSE, the resulting number of CV layers increases until the performance of DCNN network saturates, which unfortunately leads that the network involves with a large number feature maps in each layer. To avoid the network redundancy and improve the efficiency and applicability of the DCNN network in practice, we further propose the PRINCE method to prune the DCNN network.

IV. PRUNED AMP INTEGRATED DCNN CE METHOD

In this section, we develop the PRINCE method to improve the network efficiency and robustness. The essence of the PRINCE is to prune the DCNN network in AMP-DCNN by truncating the insignificant feature maps in CV layers [43]. As shown in Fig. 4, the process of obtaining the pruned DCNN...
network can be divided into three parts, namely, training with regularization, pruning and refining, respectively.

A. Training With Regularization

In the DCNN network, CV layers are often used to extract features from data, and encode the features in the feature maps. However, not every feature map can detect the important features. There exist insignificant feature maps in CV layers of the trained AMP-DCNN network, whose output values are close to zero. It is reasonable to prune these feature maps, which contribute little to the final result of DCNN network. However, the number of insignificant feature maps in the DCNN network under normal training is actually small. This means that pruning them brings little compression. Therefore, it is important to create more insignificant feature maps, by condensing more features into a few important feature maps. This can be accomplished by training with regularization.

Training with regularization prepares more insignificant feature maps by forcing the output of the feature maps close to zero. Specifically, as mentioned in (17b), each feature map is associated with a scaling factor $\gamma$ in the BN layer, which is multiplied with the output of that feature map. If the scaling factor is close to zero, the output of the corresponding feature map can be scaled close to zero. Training with regularization enables the scaling factors to be near zero by penalizing the value of the scaling factors with regularization. Specifically, the training objective function can be represented as

$$L = \sum_{(\hat{H}_{AMP}[k], H[k])} l_{MSE}(f(\hat{H}_{AMP}[k], W_{DCNN}, H[k])) + \lambda \sum_{\gamma \in \Gamma} g(\gamma), \quad (20)$$

where $\hat{H}_{AMP}[k]$ and $H[k]$ denote the training input and target output, respectively. In addition, $f$ represents the transformation of the network on the input, $W_{DCNN}$ describes the weights of DCNN network to be trained. Moreover, $l_{MSE}$ states the training loss function of the DCNN network, $g(\gamma) = |\gamma|$ is the penalty function to make the value of $\gamma$ close to zero, and $\lambda$ denotes the regularization factor to balance the training loss and penalty term. Through training with regularization, more insignificant feature maps arise with the scale factors close to zero.

B. Pruning

After training with regularization, the insignificant feature maps can be pruned to shrink the network scale. The pruning process is demonstrated in Fig. 4. In particular, the feature maps with the absolute values of the scaling factors below a certain threshold are pruned. To prune a certain ratio $p$ of feature maps for the network, the threshold value can be determined by collecting the absolute values of the scale factors across all the BN layers into a set. Then, the threshold value is chosen as the value, which is greater than the smallest ratio $p$ of the values in the set and smaller than the rest values. The ratio of pruned feature maps is named as the pruning ratio, denoted by $P_r$. For example, $80\%$ of the feature maps in the CV layers are pruned when the $P_r$ is set as $80\%$. It is worth noticing that the pruning ratio can not be achieved and further increased when all the feature maps in a CV layer are pruned. This is because that the DCNN network would fail with the intermediate CV layer removed. The pruning procedure leads the termination of all incoming and outgoing connections for the pruned feature maps. In the meantime, the weights and biases of other feature maps still maintain the same after pruning, which keep the important features extracted. Through the pruning procedure, the size of the...
DCNN network can be reduced by removing the insignificant feature maps, which leads to less computation complexity. Furthermore, the over-fitting problem can be mitigated by removing the unnecessary feature maps, and the robustness of the PRINCE method is enhanced.

C. Refining

Since pruning the feature maps may degrade the estimation performance of the original DCNN network, refining of the pruned network is needed to mitigate this effect. The refining process is mainly to fine-tune the pruned network by training, which has fast convergence. The number of feature maps in the CV layers is substantially decreased after pruning and refining. As a result, the refined pruned network is slimmer and more efficient with a smaller model size and less computational operations, which can adapt to the limitations of storage and computational resource in different environments. Remarkably, the estimation performance of the pruned network only degrades slightly after the refining process. The procedures of the PRINCE algorithm are summarized in Algorithm 1.

Algorithm 1 PRINCE Algorithm

Input: Receive signal $y[k]$, measurement matrix $\Phi$, number of iterations $T$, regularization factor $\lambda$, pruning ratio $p$

1. $\hat{H}_{\text{AMP}}[k] = \text{AMP}(y[k], \Phi, T)$
2. Train the DCNN network by (20) with factor $\lambda$
3. Prune the trained DCNN network with pruning ratio $p$
4. Refine the pruned DCNN network
5. Input $\hat{H}_{\text{AMP}}[k]$ into the refined DCNN to obtain $\hat{H}_{\text{PRINCE}}[k]$

Output: $\hat{H}_{\text{PRINCE}}[k]$.

V. PERFORMANCE EVALUATION AND ANALYSIS

In this section, we evaluate the performance of the AMP-DCNN and PRINCE, in terms of convergence, effect of limited training data, estimation accuracy and computational complexity. The estimation accuracy is revealed based on the normalized mean squared error (NMSE), which is defined as

$$\text{NMSE} = \frac{\sum_{k=0}^{K-1} \| \hat{H}[k] - \hat{H}[k] \|^2_F}{\sum_{k=0}^{K-1} \| H[k] \|^2_F}.$$  (21)

A. Datasets and Simulation Setup

The proposed AMP-DCNN and PRINCE are trained and tested on two different channel datasets generated by the ray-tracing (RT) method, including a publicly accessible mmWave dataset at 60 GHz, and a self-generated THz dataset at 0.3 THz. The mmWave dataset is selected from the Raymobtime datasets [52], which consider 3D scenarios and take the mobility and time evolution of the receivers into consideration. An example of the simulation scenario of the Raymobtime is shown in Fig. 5(a). Moreover, the Raymobtime datasets incorporate Remcom Wireless Insite [51] and a open source Simulator of Urban Mobility (SUMO) [53] for mobility simulation. During our evaluation, we select the s002 dataset in the Raymobtime datasets. This dataset is generated at 60 GHz and contains 1000 channels by considering 10 fixed receivers.

To obtain the THz dataset, we construct a 3D scenario using Remcom Wireless Insite and measure 1000 channel realizations. Specifically, as illustrated in Fig. 5(b), a typical street with several concrete buildings of different heights and flat terrain are considered in the simulation scenario. We fix Tx at the top of a building of height 30 m, and randomly select 1000 Rx points. Moreover, we fix the operation frequency at 0.3 THz to obtain 1000 different channel realizations. In Fig. 5(b), we show an example of 5 Rxs, the propagation paths of different path gains are also illustrated.

The simulation parameters are selected as follows. For the mmWave channel, the number of antennas and RF chains at Tx and Rx are $N_t = 64$, $L_t = 2$, $N_r = 16$ and $L_r = 4$, respectively. Moreover, the number of subcarriers and pilots are $K = 256$ and $M = 100$. The corresponding

![Fig. 5. Ray-tracing simulation in a 3D scenario. The intensity of the received power of each ray is represented by colors.](image-url)
parameters for the THz channel are $N_t = 512$, $L_t = 4$, $N_r = 32$, $L_r = 8$, $K = 16$ and $M = 1000$, respectively. The bandwidth is set as 1.76 GHz for mmWave channel and 17.6 GHz for THz channel. The resolution of the phase shifter is set as $N_b = 4$. The received signal $y[k]$ for both mmWave and THz frequencies is obtained as in (13), in which the measurement matrix $\Phi$ is the same for different subcarriers over every channel, which however varies for different channel realizations. To generate the measurement matrix $\Phi$, the symbol $q^{(m)}$ is generated according to the normal distribution. Considering the resolution of the phase shifter, the phase values can be chosen from a limited set $\Omega = \{1, e^{j2\pi/2^b}, e^{j4\pi/2^b}, \ldots, e^{j2\pi(2^b-1)/2^b}\}$. The precoder $F_{tr}^{(m)}$ and the combiner $W_{tr}^{(m)}$ are generated with random phase shifts as in [12] and [54], and the phase values are uniformly distributed in the set of phase values $\Omega$. In terms of noise, 6 different values of SNR from $-5$ dB to 20 dB are considered. Next, the received signals are pre-processed by AMP to generate the coarse estimated channel matrices. Followed by that, the estimated channel matrices at different SNRs are mixed together, which compose the dataset to train the DCNN network. Therefore, both mmWave and THz datasets contain 6000 samples. Moreover, the datasets are splitted, in which 90% for training and 10% for testing. After generating the datasets, the DCNN network is trained with the adaptive moment estimation (Adam) optimizer for its fast convergence rate, and the learning rate is set as $\ell_r = 10^{-3}$. Finally, the PRINCE method is operated with the regularization factor $\lambda = 10^{-5}$, which is empirically chosen from the values $10^{-3}$, $10^{-4}$ and $10^{-5}$. Moreover, the pruning ratios for PRINCE are selected as values from 10% to 70% and from 10% to 80% for mmWave dataset and THz dataset, respectively.

All the experimental results are implemented on a PC with Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz and an Nvidia GeForce RTX 2080 Ti GPU. In addition, the simulation of the UM-MIMO system and the conventional CE methods including OMP [9], AMP [10], SOMP [13] and MMSE [20], are operated in Matlab (R2018a) environment, while SF-CNN [42], the proposed DCNN network and the pruning method are carried out in the Visual Studio framework.

### B. Convergence Evaluation

The convergence performance of the proposed DCNN model for both mmWave and THz channels is evaluated in Fig. 6 by plotting the training loss and testing loss versus the number of epochs. One epoch is defined as a complete training cycle through the training dataset. By observing that the training loss is close to the corresponding testing loss in different setups, we state that there is no over-fitting problem. Moreover, the testing losses are stable after 10 epochs for the two scenarios, which verifies the fast convergence performance of the DCNN model. By contrast, the refining step after pruning 10% feature maps takes less than 5 epochs to converge, which incurs little training overhead for PRINCE and still yields expedite convergence. The fast convergence is because that the weights and biases of the unpruned feature maps maintain the same after pruning, and only minor changes to these parameters in the refining step are required to converge.

### C. Effect of Limited Training Data

In the practical deployment of CE schemes, DL methods are usually faced with the problem of limited training data set due to the limitation of time and resources, and it is essential for the DL method to maintain high estimation accuracy with a limited training dataset. To evaluate the performance of the proposed DCNN network with limited training resources, we train the proposed DCNN network under different sizes of training sets for 20 epochs. As shown in Fig. 7, the proposed DCNN network still has the ability to converge even with only 1000 samples in the training set. Moreover, the testing loss decreases with the size of the training set, which indicates that the performance of the DCNN network improves with the increasing of data. In addition, when the size of the dataset decreases by 83% from 6000 to 1000, the testing loss only increases by 28% for 60 GHz and 17% for 0.3 THz channels, respectively. Therefore, there is no drastic increase of the testing loss as the size of dataset decreases. We can thereby state that the performance of the proposed DCNN network is robust with limited training data.

### D. Estimation Accuracy

To assess the estimation accuracy of the proposed AMP-DCNN model, the CS-based methods including the OMP [9], AMP [10], SOMP [13], the conventional CE schemes of non-ideal MMSE estimator [20] and the data-based DL method SF-CNN [42] are included for comparison. In addition, to show the advantage of AMP algorithm against other CS-based methods in the AMP-DCNN method, OMP-DCNN is constructed for comparison by replacing the AMP algorithm with the OMP algorithm. For fair comparison, the same datasets including the mmWave channel and the THz channel are used in these methods. Moreover, the training is conducted...
with the training dataset containing channels over six different SNR values from -5 dB to 20 dB, while the testing is carried out for each SNR separately, to evaluate the testing accuracy and robustness at various SNR values.

As shown in Fig. 8(a), based on the mmWave channel dataset, the proposed AMP-DCNN method achieves the lowest NMSE among the six different methods, in which the NMSE of the proposed AMP-DCNN method reaches below -1% degradation by less than 1% by the AMP-DCNN method. In Fig. 9(a), based on the THz channel dataset, the proposed AMP-DCNN method outperforms the benchmark methods as shown in Fig. 8(b). The NMSE is lower than -10 dB at 5 dB SNR for the AMP-DCNN method. No matter which dataset the estimation is evaluated on, AMP-DCNN has shown high accuracy in CE.

Moreover, we vary the pruning ratio, and analyze the estimation accuracy of the PRINCE algorithm, upper-bounded by the AMP-DCNN method. In Fig. 9(a), based on the mmWave channel, we find that the estimation accuracy degrades by less than 1% and 15%, when the pruning ratio increases from 10% to 40%. At 10 dB SNR, the NMSE lies below -6 dB for the PRINCE with the pruning ratio at 70%. In addition, the performance of PRINCE based on the THz channel with different pruning ratios is shown in Fig. 9(b). The performance of the network pruning thirty percent is nearly the same as the uncontacted AMP-DCNN network. There is roughly 2 dB degradation on average for different SNR values, after pruning eighty percent of feature maps of the DCNN.

Overall, AMP-DCNN has advantages over the conventional methods in terms of the estimation accuracy in both mmWave and THz channel datasets, which presents the superiority in accurate CE. Moreover, AMP-DCNN can achieve good performance on the dataset consisting of channels in different cases over different SNR values, which shows that AMP-DCNN is robust to the varying environment. More importantly, the pruning of insignificant channels compromises the accuracy reasonably, which makes PRINCE method appealing for practical implementation.

E. Computational Complexity

The computational complexities of the aforementioned methods are compared in Table II. Particularly, the computational complexity of the OMP algorithm is $O(SN_tN_r)$, where $S$ denotes the sparsity of the channel, $N_t$ and $N_r$ represent the number of antennas at Tx and Rx, respectively. Similarly, the computational complexity of AMP algorithm is $O(TN_tN_r)$, where $T$ denotes the number of iterations in the AMP algorithm. The complexity of the SOMP algorithm is $O(S_oN_tN_r)$, which is less than OMP, since SOMP saves computation complexity by finding the common sparse support over different subcarriers. The MMSE estimator has the complexity of $O(N_t^3N_r^3)$. By contrast, the complexity of the DCNN network is calculated by the dimensions of CV layers, which equals to $O(N_tN_r\sum_{l=1}^{L}F_l^2N_{l-1}N_l)$, where $l_c$ is the number of CV layers, $F_l$ is the side length of the filters, and $N_l$ is the number of feature maps in the $l$th CV layer. Combining with the tentative estimation module having the complexity of $O(N_tN_r(N_t + N_r))$, SF-DCNN possesses the complexity of $O(N_tN_r((N_t + N_r) + \sum_{l=1}^{L}F_l^2N_{l-1}N_l))$. Similarly, the complexity of OMP-DCNN is $O(N_tN_r(S + \sum_{l=1}^{L}F_l^2N_{l-1}N_l))$, and the complexity of AMP-DCNN is $O(N_tN_r(T + \sum_{l=1}^{L}F_l^2N_{l-1}N_l))$. After pruning the redundant feature maps in the DCNN network, the complexity of the PRINCE is reduced to $O(N_tN_r(T + \sum_{l=1}^{L}F_l^2(N_{l-1} - p_l)(N_l - p_l)))$, where $p_l$ describes the reduced number of feature maps for the $l$th CV layer.

The complexities of OMP, AMP, SOMP and AMP-DCNN increase linearly with the dimension of channel matrix $N_cN_t$, while AMP-DCNN has much better estimation accuracy than the others of them. On the contrary, the complexity of MMSE estimator grows cubically with the channel dimension, while the estimation accuracy of MMSE is still inferior to AMP-DCNN. SF-DCNN, OMP-DCNN and AMP-DCNN have the same complexity for the DL part, since the number of CV layers and feature maps are set the same for the three methods.

In terms of the preprocessing module before the DL part, AMP-DCNN has the lowest complexity. Furthermore, the PRINCE is derived from the AMP-DCNN model, and has even lower computation complexity than the original AMP-DCNN thanks to the pruning of redundant feature maps in CV layers. Moreover, the inference time for the AMP-DCNN network

![Fig. 7. Testing loss with different sizes of training data set.](image-url)
Fig. 8. Channel estimation accuracy of the AMP-DCNN algorithm.

Fig. 9. Channel estimation accuracy of the pruned AMP-DCNN algorithm.

Fig. 10. Number of parameters in the pruned DCNN with different pruning ratios.
is 0.12 ms, which is practical for applications considering the channel coherence time on the order of millisecond in the THz band [49].

The complexity of PRINCE decreases as the pruning ratio rises. To gain vivid comparison of the complexity between the AMP-DCNN and PRINCE for the mmWave and THz UM-MIMO systems, the number of parameters including the weights and biases for the DCNN part is plotted in Fig. 10. By pruning the feature maps in the CV layers, the parameters of the corresponding filters and the incoming connections from BN layer are reduced. Specifically, based on the mmWave dataset, the number of parameters is reduced from 372352 to 32579 when seventy percent of feature maps are pruned. Similarly, there is a large reduction in the number of parameters from 373352 to 16597 when the pruning ratio is 80% for THz dataset. Moreover, the number of computation operations of the DCNN part measured by the giga floating point operations (GFLOPs) is depicted in Fig 11. The pruning of the feature maps in CV layers helps reducing the computationally intensive convolution operations, leading that the number of operations decreases from 0.38 GFLOPs to 0.03 GFLOPs for mmWave dataset when the pruning ratio is 70%. Moreover, the number of operations for the THz dataset is reduced from 6.12 GFLOPs to 0.28 GFLOPs with a pruning ratio of 80%. From Fig. 10 and Fig. 11, we can observe that the number of parameters and the number of computation operations both shrink quickly with the pruning ratio, which verifies the significant reduction of the complexity of the PRINCE method.

VI. CONCLUSION

In this paper, we first proposed a novel AMP-DCNN method, which exploits the benefits of traditional AMP CE method and the DL tool for CE of the mmWave and THz UM-MIMO systems. As a generalized example in deploying DL for CE, AMP is first exploited to obtain the coarse CE result. Then, a designed DCNN refines the results from AMP by further enhancing the estimation accuracy with its powerful learning ability. Based on AMP-DCNN, we further developed a PRINCE method, which reduces the DCNN network size, by truncating the significant feature maps in the CV layers through training with regularization, pruning and refining.

Extensive simulations validate the remarkable estimation accuracy of the proposed AMP-DCNN, whose estimation performance outperforms the benchmark solutions and reaches NMSE of −12 dB when SNR is 10 dB. Moreover, the proposed AMP-DCNN demonstrates robustness in different sizes of training dataset. Furthermore, PRINCE achieves significantly reduced complexity compared to the AMP-DCNN, with negligible performance degradation. With 80% truncated feature maps, the estimation NMSE of PRINCE remains −10 dB at SNR = 10 dB.

REFERENCES

[1] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, “Toward 6G networks: Use cases and technologies,” IEEE Commun. Mag., vol. 58, no. 3, pp. 55–61, Mar. 2020.
[2] Z. Chen et al., “Terahertz wireless communications for 2030 and beyond: A cutting-edge frontier,” IEEE Commun. Mag., vol. 59, no. 11, pp. 66–72, Nov. 2021.
[3] I. F. Akyildiz, C. Han, and S. Nie, “Combating the distance problem in the millimeter wave and terahertz frequency bands,” IEEE Commun. Mag., vol. 56, no. 6, pp. 102–108, Jun. 2018.
[4] C. Han, L. Yan, and J. Yuan, “Hybrid beamforming for terahertz wireless communications: Challenges, architectures, and open problems,” IEEE Wireless Commun., vol. 28, no. 4, pp. 198–204, Aug. 2021.
[5] I. F. Akyildiz and J. M. Jornet, “Realizing ultra-massive MIMO (1024×1024) communication in the (0.06–10) terahertz band,” Nano Commun. Netw., vol. 8, pp. 46–54, Jun. 2016.
[6] Z. Gao, L. Dai, D. Mi, Z. Wang, M. A. Imran, and M. Z. Shakir, “Mmwave massive-MIMO-based wireless backhaul for the 5G ultra-dense network,” IEEE Wireless Commun., vol. 22, no. 5, pp. 13–21, Oct. 2015.
[7] C. Han, J. M. Jornet, and I. Akyildiz, “Ultra-massive MIMO channel modeling for graphene-enabled terahertz-band communications,” in Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring), Jun. 2018, pp. 1–5.
[8] L. Yan, C. Han, and J. Yuan, “A dynamic array-of-subarrays architecture and hybrid precoding algorithms for terahertz wireless communications,” IEEE J. Sel. Areas Commun., vol. 38, no. 9, pp. 2041–2056, Sep. 2020.
[52] A. Klautau, P. Batista, N. Gonzalez-Prelcic, Y. Wang, and R. W. Heath Jr., “5G MIMO data for machine learning: Application to beam-selection using deep learning,” in Proc. Inf. Theory Appl. Workshop (ITA), Feb. 2018, pp. 1–9.
[53] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, “Recent development and applications of SUMO-simulation of urban mobility,” Int. J. Adv. Syst. Meas., vol. 5, nos. 3–4, pp. 128–138, 2012.
[54] A. Alkhateeb, G. Leus, and R. W. Heath, “Compressed sensing based multi-user millimeter wave systems: How many measurements are needed?” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2015, pp. 2909–2913.

Zhengdong Hu received the B.E. degree from the UM-SJTU Joint Institute, Shanghai Jiao Tong University, China, in 2021, where he is currently pursuing the Ph.D. degree with the Terahertz Wireless Communications (TWC) Laboratory. His current research interests include machine/deep learning for terahertz communications.

Yuhang Chen received the B.E. degree in communication engineering from the Harbin Institute of Technology, China, in 2019. She is currently pursuing the Ph.D. degree with the Terahertz Wireless Communications (TWC) Laboratory, University of Michigan–Shanghai Jiao Tong University (UM–SJTU) Joint Institute, Shanghai Jiao Tong University, China. Her current research interests include terahertz channel estimation and beam tracking.

Chong Han (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology, USA, in 2016. Since 2021, he has been with the Department of Electronic Engineering and the Cooperative Medianet Innovation Center (CMIC), Shanghai Jiao Tong University, China. He is currently a John Wu & Jane Sun Endowed Associate Professor with the University of Michigan–Shanghai Jiao Tong University (UM–SJTU) Joint Institute, Shanghai Jiao Tong University, and the Director of the Terahertz Wireless Communications (TWC) Laboratory. His research interests include terahertz and millimeter-wave communications. He is a member of the ACM. He was a recipient of the 2018 Elsevier NanoComNet (Nano Communication Networks) Young Investigator Award, the 2017 Shanghai Sailing Program, and the 2018 Shanghai Chenguang Program. He is a TPC Chair to organize multiple IEEE and ACM conferences and workshops, including GC 2023 SAC THz communications. He has been the Co-Founder and the Vice-Chair of the IEEE ComSoc Special Interest Group (SIG) on Terahertz Communications since 2021. He is a (Guest) Editor of IEEE Journal on Selected Areas in Communications, IEEE Journal of Selected Topics in Signal Processing, IEEE Transactions on Nanotechnology, and IEEE Open Journal of Vehicular Technology.