

Deep Learning-Based Frequency-Selective Channel Estimation for Hybrid mmWave MIMO Systems

Asmaa Abdallah¹, Member, IEEE, Abdulkadir Celik², Senior Member, IEEE,
 Mohammad M. Mansour³, Senior Member, IEEE, and Ahmed M. Eltawil³, Senior Member, IEEE

Abstract—Millimeter wave (mmWave) massive multiple-input multiple-output (MIMO) systems typically employ hybrid mixed signal processing to avoid expensive hardware and high training overheads. However, the lack of fully digital beamforming at mmWave bands imposes additional challenges in channel estimation. Prior art on hybrid architectures has mainly focused on greedy optimization algorithms to estimate frequency-flat narrowband mmWave channels, despite the fact that in practice, the large bandwidth associated with mmWave channels results in frequency-selective channels. In this paper, we consider a frequency-selective wideband mmWave system and propose two deep learning (DL) compressive sensing (CS) based algorithms for channel estimation. The proposed algorithms learn critical a priori information from training data to provide highly accurate channel estimates with low training overhead. In the first approach, a DL-CS based algorithm simultaneously estimates the channel supports in the frequency domain, which are then used for channel reconstruction. The second approach exploits the estimated supports to apply a low-complexity multi-resolution fine-tuning method to further enhance the estimation performance. Simulation results demonstrate that the proposed DL-based schemes significantly outperform conventional orthogonal matching pursuit (OMP) techniques in terms of the normalized mean-squared error (NMSE), computational complexity, and spectral efficiency, particularly in the low signal-to-noise ratio regime. When compared to OMP approaches that achieve an NMSE gap of $\{4 - 10\}$ dB with respect to the Cramer Rao Lower Bound (CRLB), the proposed algorithms reduce the CRLB gap to only $\{1 - 1.5\}$ dB, while reducing complexity by two orders of magnitude.

Index Terms—Deep learning, channel estimation, compressive sensing, frequency-selective channel, mmWave, MIMO, convolutional neural networks, denoising, sparse recovery.

I. INTRODUCTION

MILLIMETER wave (mmWave) communication has emerged as a key technology to fulfill beyond fifth-generation (5G) network requirements, such as

Manuscript received February 9, 2021; revised June 23, 2021 and September 7, 2021; accepted October 24, 2021. Date of publication November 5, 2021; date of current version June 10, 2022. The associate editor coordinating the review of this article and approving it for publication was Y. Shen. (Corresponding author: Asmaa Abdallah.)

Asmaa Abdallah, Abdulkadir Celik, and Ahmed M. Eltawil are with the Department of Electrical and Computer Engineering, King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia (e-mail: asmaa.abdallah@kaust.edu.sa).

Mohammad M. Mansour is with the Department of Electrical and Computer Engineering, American University of Beirut (AUB), Beirut 1107 2020, Lebanon.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TWC.2021.3124202>.

Digital Object Identifier 10.1109/TWC.2021.3124202

enhanced mobile broadband, massive connectivity, and ultra-reliable low-latency communications. The mmWave band offers an abundant frequency spectrum (30-300 GHz) at the cost of low penetration depth and high propagation losses. Fortunately, its short-wavelength mitigates these drawbacks by allowing the deployment of large antenna arrays into small form factor transceivers, paving the way for multiple-input multiple-output (MIMO) systems with high directivity gains [1]–[4].

Hybrid MIMO structures have been introduced to operate at mmWave frequencies because an all-digital architecture, with a dedicated radio frequency (RF) chain for each antenna element, results in expensive system architecture and high power consumption at these frequencies [2]. In these hybrid architectures, phase-only analog beamformers are employed to steer the beams using steering vectors of quantized angles. The down-converted signal is then processed by low-dimensional baseband beamformers, each of which is dedicated to a single RF chain [5], [6]. The number of RF chains is significantly reduced with this combination of high-dimensional phase-only analog and low-dimensional baseband digital beamformers [6]. Moreover, optimal configuration of the digital/analog precoders and combiners requires instantaneous channel state information (CSI) to achieve spatial diversity and multiplexing gain [7]. However, acquiring mmWave CSI is challenging with a hybrid architecture due to the following reasons [5]: 1) There is no direct access to the different antenna elements in the array since the channel is seen through the analog combining network, which forms a compression stage for the received signal when the number of RF chains is much smaller than the number of antennas, 2) the large channel bandwidth yields high noise power and low received signal-to-noise-ratio (SNR) before beamforming, and 3) the large size of channel matrices increases the complexity and overhead associated with traditional precoding and channel estimation algorithms. Therefore, low complexity channel estimation for mmWave MIMO systems with hybrid architecture is necessary.

A. Related Work

Channel estimation techniques typically leverage the sparse nature of mmWave MIMO channels by formulating the estimation as a sparse recovery problem and apply compressive sensing (CS) methods to solve it. Compressive sensing is a general framework for estimation of sparse vectors from linear measurements [8]. The estimated *supports* of the sparse

vectors using CS help identify the indices of Angle-of-Arrival (AoA) and Angle-of-Departure (AoD) pairs for each path in the mmWave channel, while the *amplitudes* of the non-zero coefficients in the sparse vectors represent the channel gains for each path. Therefore, these *supports* and *amplitudes* are key components to be estimated to obtain accurate CSI. Moreover, it has been shown that pilot training overhead can be reduced with compressive estimation, unlike the conventional approaches such as those based on least squares (LS) estimation [6].

Several channel estimation methods based on CS tools that explore the mmWave channel sparsity have been investigated in the literature [6], [9]–[13]. A distributed grid matching pursuit (DGMP) channel estimation scheme is presented in [9], where the dominant entries of the line-of-sight (LoS) channel path are detected and updated iteratively. In [10], an orthogonal matching pursuit (OMP) channel estimation scheme to detect multiple channel paths support entries is also considered. Likewise, a simultaneous weighted orthogonal matching pursuit (SW-OMP) channel estimation scheme based on a weighted OMP method is developed in [11] for frequency-selective mmWave systems. A sparse reconstruction problem is formulated in [11] to estimate the channel independently for every subcarrier by exploiting common sparsity in the frequency domain. In [13], SIMGW-OMP is presented based on maximum likelihood where the authors use the SW-OMP algorithm proposed in [11] to find an initial coarse estimate for the AoA/AoD. Afterward, the SIMGW-OMP iteratively refines this coarse estimate using gradient computations to approximate the maximum likelihood estimator. Therefore, the SIMGW-OMP approach has a higher complexity order than the SW-OMP as it iteratively computes the gradient updates for each channel multi-path component, where the gradient is calculated distinctively for each AoA and AoD. However, such optimization and CS-based channel estimation schemes detect the support indices of the mmWave channel sequentially and greedily and hence are not globally optimal [12]. Therefore, it is difficult to directly find the channel supports, especially when mmWave channels exhibit wideband frequency-selective fading.

Alternatively, deep learning (DL) approaches and data-driven algorithms have recently received much attention as key enablers for beyond 5G networks. Traditionally, signal processing and numerical optimization techniques have been heavily used to address channel estimation at mmWave bands [9]–[12]. However, optimization algorithms often demand considerable computational complexity overhead, which creates a barrier between theoretical design/analysis and real-time processing requirements. Hence, the prior data-set observations and deep neural network (DNN) models can be leveraged to learn the non-trivial mapping from compressed received pilots to channels. DNNs can be used to approximate the optimization problems by selecting the suitable set of parameters that minimize the approximation error. The use of DNNs is expected to substantially reduce computational complexity and processing overhead since it only requires several layers of simple operations such as matrix-vector multiplications. Moreover, several successful DL applications

have been demonstrated in wireless communications problems such as channel estimation [14]–[25], analog beam selection [26], [27], and hybrid beamforming [26], [28]–[32]. Besides, DL-based techniques, when compared with other conventional optimization methods, have been shown [15], [30], [31], [33] to be more computationally efficient in searching for beamformers and more tolerant to imperfect channel inputs. Denoising convolutional neural networks (DnCNNs) are widely used in the context of image denoising [34], [35]. Interestingly, DnCNN denoises images by learning Gaussian noise rather than the content of images. However, it is challenging to estimate channels with DnCNN directly. The DnCNN processes real numbers and cannot be directly adopted in channel estimation where the signals are complex. For instance, authors of [16] use complex DnCNN (CDnCNN), a variant of DnCNN, to estimate channels where the real and imaginary parts of the received sounding signals are separated into different channels. Moreover, in [17], the authors considered channel estimation for MISO Communications with Large Intelligent Surfaces (LIS), where they use the least square estimate of the channel as input to the DnCNN. They separate the real and imaginary parts into two different input channels and perform the denoising using DnCNN. This paper is distinct from [16], [17] in 1) considering a system model based on hybrid MIMO architecture, 2) estimating wideband frequency-selective mmWave channels, and 3) using the DnCNN to simultaneously detect the channel supports that correspond to the indices of Angle-of-Arrival (AoA) and Angle-of-Departure (AoD) pairs for each path in the frequency-selective mmWave channels. In [18], a learned denoising-based approximate message passing (LDAMP) network is presented to estimate the mmWave communication system with lens antenna array, where the noise term is detected and removed to estimate the channel. However, channel estimation for mmWave massive MIMO systems with hybrid architecture is not considered in [18].

Prior work on channel estimation for hybrid mmWave MIMO architecture [18]–[25], [28]–[30], [36]–[39] consider the narrow-band flat fading channel model for tractability, while the practical mmWave channels exhibit the wideband frequency-selective fading due to the very large bandwidth, short coherence time and different delays of multipath [11], [40], [41]. MmWave environments such as indoor and vehicular environments are highly variable with short coherence time [41] which requires channel estimation techniques that are robust to the rapidly changing channel characteristics.¹ Accordingly, this paper presents combination of DL and CS methods to identify AoA/AoD pairs' indices and estimate the channel amplitudes for frequency-selective channel estimation of hybrid MIMO systems.

B. Contributions of the Paper

In this paper, we propose a frequency-selective channel estimation framework for mmWave MIMO systems with a hybrid architecture. We leverage the sparse nature of mmWave

¹The coherence time is within few milliseconds such as 5 ms when operating at 60 GHz with 1 GHz bandwidth [41].

MIMO channels by formulating the estimation as a sparse recovery problem and apply data-driven methods along with CS to find the channel supports which will help identify the indices of AoA and AoD pairs for each path in the frequency-selective mmWave channels. By considering the mmWave channel sparsity, the developed method aims at reaping the full advantages of both CS and DL methods. We consider the received pilot signal as an image, and then employ a denoising convolutional neural network (DnCNN) from [34] for channel amplitude estimation. Thereby, we treat image denoising as a plain discriminative learning problem, i.e., separating the noise from a noisy image by feed-forward convolutional neural networks (CNNs). The main motivations behind using CNNs are twofold: First, deep CNNs have been recognized to effectively extract image features [34]. Second, considerable advances have been achieved on regularization and learning methods for training CNNs, including Rectifier Linear Unit (ReLU), batch normalization, and residual learning [35]. These methods can be adopted in CNNs to speed up the training process and improve the denoising performance. The main contributions of the paper can be summarized as follows:

- 1) We propose a deep learning compressed sensing channel estimation (DL-CS-CE) scheme for wideband mmWave massive MIMO systems. The proposed DL-CS-based channel estimation (DL-CS-CE) algorithm aims at exploiting the information on the support coming from every subcarrier in the MIMO-OFDM system. It is executed in two steps: channel amplitude estimation through deep learning and channel reconstruction. We train a DnCNN using *real* mmWave channel realizations obtained from Raymobtime.² The correlation between the received signal vectors and the measurement matrix is fed into the trained DnCNN to predict the channel amplitudes. Using the obtained channel amplitudes, the indices of dominant entries of the channel are obtained, based on which the channel can be reconstructed. Unlike the existing work of [9]–[11] that estimate the dominant channel entries sequentially, we estimate dominant entries simultaneously (in one-shot), which is able to save in computational complexity and improve estimation performance.
- 2) Using the DL-CS-CE for support detection, we propose a refined DL-CS-CE algorithm that exploits the spatially common sparsity within the system bandwidth. A channel reconstruction with a low complexity multi-resolution fine-tuning approach is developed that further improves NMSE performance by enhancing the accuracy of the estimated AoAs/AoDs. The channel reconstruction is performed by consuming a very small

²We consider the challenging off-grid mmWave channel realizations based on the realistic dataset from Raymobtime where AoAs and AoDs do not lie on dictionary grids. Raymobtime is a collection of realistic ray-tracing datasets for wireless communications. It uses ray-tracing and 3D scenarios with mobility and time evolution for obtaining consistency over time, frequency, and space. We refer interested readers to <https://www.lasse.ufpa.br/raymobtime/> for more information about Raymobtime dataset.

amount of pilot training frames, which significantly reduces the training overhead and computational complexity.

- 3) Simulation results in the low SNR regime show that both proposed algorithms significantly outperform the frequency domain approach developed in [11]. Numerical results also show that using a reasonably small pilot training frames, approximately in the range of 60–100 frames, leads to substantially low channel estimation errors. The proposed algorithms are also compared with existing solutions by analyzing the trade-off between delivered performance and incurred computational complexity. Our analysis reveals that both proposed channel estimation methods achieve the desired performance at significant lower complexity. The developed approaches are shown to attain an NMSE gap of 1 – 1.5 dB with the Cramer Rao Lower Bound (CRLB) compared to the 4 – 10 dB gap attained by the SW-OMP technique, while reducing the computational complexity by two orders of magnitude.

C. Notation and Paper Organization

Bold upper case, bold lower case, and lower case letters correspond to matrices, vectors, and scalars, respectively. Scalar norms, vector L_2 norms, and Frobenius norms, are denoted by $|\cdot|$, $\|\cdot\|_2$, and $\|\cdot\|_F$, respectively. We use \mathcal{X} to denote a set. \mathbf{I}_X denotes a $X \times X$ identity matrix. $\mathbb{E}[\cdot]$, $(\cdot)^T$, $(\cdot)^*$, and $(\cdot)^\dagger$ stand for expected value, transpose, complex conjugate, and Hermitian. \mathbf{X}^\dagger stands for the Moore-Penrose pseudo-inverse of \mathbf{X} . $[\mathbf{x}]_i$ represents i^{th} element of a vector \mathbf{x} . The $(i, j)^{\text{th}}$ entry of a matrix \mathbf{X} is denoted by $[\mathbf{X}]_{i,j}$. In addition, $[\mathbf{X}]_{:,j}$ and $[\mathbf{X}]_{:, \Omega}$ denote the j^{th} column vector of matrix \mathbf{X} and the sub-matrix consisting of the columns of matrix \mathbf{X} with indices in set Ω . $\{a\} \bmod b$ means a modulo b . $\mathcal{CN}(\boldsymbol{\mu}, \mathbf{C})$ refers to a circularly-symmetric complex Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix \mathbf{C} . The operations $\text{vec}(\mathbf{X})$, $\text{vec2mat}(\mathbf{x}, sz)$, $\text{sub2ind}(sz, [r, c])$, and $\text{ind2sub}(sz, i)$ correspond to transforming a matrix into a vector, transforming a vector into a matrix for a defined size (sz), transforming the row r and column c subscripts of a matrix into their corresponding linear index, and transforming the linear index i into its corresponding row and column subscripts for a matrix of a defined size (sz), respectively. $\mathbf{X} \otimes \mathbf{Y}$ is the Kronecker product of \mathbf{X} and \mathbf{Y} . Key model-related notation is listed in Table I. For the sake of fair comparison and consistency, the notations and system model used in this work is similar to [10], [11], [13].

The rest of the paper is organized as follows. The system model for the frequency selective mmwave MIMO system is described in Section II. In Section III, the proposed two deep learning-based compressive sensing channel estimation schemes in the frequency domain are introduced. Moreover, complexity analysis in terms of convergence and computational analysis is presented in Section IV. Case studies with numerical results are simulated and analyzed based on the proposed schemes in Section V. Section VI concludes the paper.

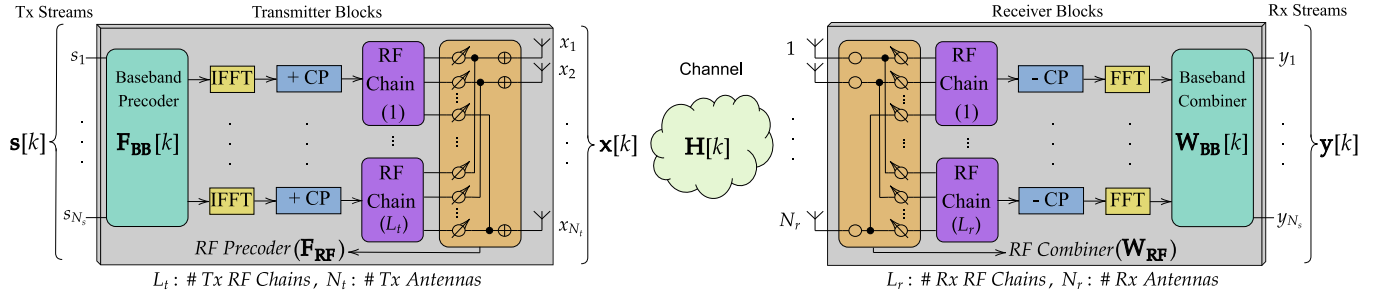


Fig. 1. Hybrid architecture system model of a mmWave MIMO system, which includes analog/digital precoders and combiners.

 TABLE I
 NOTATION

Notation	Definition
$\mathbf{F}_{\text{RF}} \in \mathbb{C}^{N_t \times L_t}$	RF analog precoder (time domain (TD))
$\mathbf{W}_{\text{RF}} \in \mathbb{C}^{N_r \times L_r}$	RF analog combiner (TD)
$\mathbf{F}_{\text{BB}}[k] \in \mathbb{C}^{L_t \times N_s}$	Baseband digital precoder (frequency domain(FD))
$\mathbf{W}_{\text{BB}}[k] \in \mathbb{C}^{L_r \times N_s}$	Baseband digital combiner (FD)
$\mathbf{s}[k] \in \mathbb{C}^{N_s \times 1}$	Data symbol vector (FD)
$\mathbf{H}_d \in \mathbb{C}^{N_r \times N_t}$	d^{th} delay tap of the channel (TD)
$\Delta_d \in \mathbb{C}^{L \times L}$	Complex diagonal matrix (time domain)
$\mathbf{A}_R \in \mathbb{C}^{N_r \times L}$	Receive array steering matrix
$\mathbf{A}_T \in \mathbb{C}^{N_t \times L}$	Transmit array steering matrix
$\mathbf{H}[k] \in \mathbb{C}^{N_r \times N_t}$	Channel at k^{th} subcarrier (FD)
$\Delta[k] \in \mathbb{C}^{L \times L}$	Complex diagonal matrix (FD)
$\mathbf{A}_R \in \mathbb{C}^{N_r \times G_r}$	Dictionary matrix for receive array response
$\mathbf{A}_T \in \mathbb{C}^{N_t \times G_t}$	Dict. matrix for transmit array response
$\hat{\mathbf{A}}_R \in \mathbb{C}^{N_r \times G_r}$	Refining dict. matrix for receive array response
$\hat{\mathbf{A}}_T \in \mathbb{C}^{N_t \times G_t}$	Refining dict. matrix for transmit array response
$\Delta_d^v \in \mathbb{C}^{G_r \times G_t}$	Path gains sparse matrix of the <i>virtual</i> channel (TD)
$\Delta^v[k] \in \mathbb{C}^{G_r \times G_t}$	Path gains sparse matrix of the <i>virtual</i> channel (FD)
$\Phi \in \mathbb{C}^{M_{L_r} \times N_t N_r}$	Measurement matrix
$\Psi \in \mathbb{C}^{N_t N_r \times G_t G_r}$	Dictionary matrix
$\mathbf{h}^v[k] \in \mathbb{C}^{G_r G_t \times 1}$	Sparse vector containing complex channel gains (FD)
$\Upsilon \in \mathbb{C}^{M_{L_r} \times G_t G_r}$	Equivalent measurement matrix
$\mathbf{y}[k] \in \mathbb{C}^{M_{L_r} \times 1}$	Received signal (FD)
$\mathbf{c}[k] \in \mathbb{C}^{G_r G_t}$	Correlation vector (FD)
$\mathbf{C}_w \in \mathbb{C}^{M_{L_r} \times M_{L_r}}$	Noise covariance matrix of $\mathbf{y}[k]$
$\mathbf{D}_w \in \mathbb{C}^{M_{L_r} \times M_{L_r}}$	Whitening matrix (upper triangular matrix)
$\mathbf{y}_w[k] \in \mathbb{C}^{M_{L_r} \times 1}$	Whitened received signal (FD)
$\Upsilon_w \in \mathbb{C}^{M_{L_r} \times G_t G_r}$	Whitened measurement matrix
$\Upsilon_w^d \in \mathbb{C}^{M_{L_r} \times G_t G_r}$	White. meas. matrix to remove detection uncertainty
$\Upsilon_w^r \in \mathbb{C}^{M_{L_r} \times G_t G_r}$	White. meas. matrix for refining
$\mathbf{C}_\alpha[k] \in \mathbb{R}^{G_r \times G_t}$	Input matrix to the DnCNN (FD)
$\mathbf{G}[k] \in \mathbb{R}^{G_r \times G_t}$	Output matrix of the DnCNN (FD)
$\mathbf{g}[k] \in \mathbb{R}^{G_r G_t \times 1}$	Vectorized form of $\mathbf{G}[k]$ (FD)
$\xi[k] \in \mathbb{C}^{L \times 1}$	Vector of actual channel gains (FD)
$\mathbf{P} \in \mathbb{C}^{M_{L_r} \times M_{L_r}}$	Projection matrix
$\mathbf{r}[k] \in \mathbb{C}^{M_{L_r} \times 1}$	Residual vector (FD)
\mathcal{T}	Sparse channel support set
\mathcal{K}	Subset from total K subcarriers

II. SYSTEM MODEL AND PROBLEM FORMULATION

This section first provides the system and channel models of frequency-selective hybrid mmWave transceivers. Then, it formulates a sparse recovery problem to estimate the sparse channel in the frequency domain.

A. System Model

As shown in Fig. 1, we consider an OFDM-based mmWave MIMO link employing a total of K subcarriers to send N_s data streams from a transmitter with N_t antennas to a receiver with N_r antennas. The system is based on a hybrid MIMO

architecture of fully connected network of quantized phase shifters, as described in [6], with $L_t < N_t$ and $L_r < N_r$ radio frequency (RF) chains at the transmitter and receiver sides. Following the notation of [11], we define a frequency-selective hybrid precoder $\mathbf{F}[k] = \mathbf{F}_{\text{RF}}\mathbf{F}_{\text{BB}}[k] \in \mathbb{C}^{N_t \times N_s}$, $k = 0, \dots, K-1$, where \mathbf{F}_{RF} and $\mathbf{F}_{\text{BB}}[k]$ are the analog and digital precoders, respectively. Although, the analog precoder is considered to be frequency-flat, the digital precoder is different for every subcarrier. During transmission, the transmitter (TX) first precodes data symbols $\mathbf{s}[k] \in \mathbb{C}^{N_s \times 1}$ at each subcarrier by applying the subcarrier-dependent baseband precoder $\mathbf{F}_{\text{BB}}[k]$. The symbol blocks are then transformed into the time domain using L_t parallel K -point inverse Fast Fourier transform (IFFT). After adding the cyclic prefix (CP), the transmitter employs the subcarrier-independent RF precoder \mathbf{F}_{RF} to form the transmitted signal. The complex baseband signal at the k^{th} subcarrier can be expressed as $\mathbf{x}[k] = \mathbf{F}_{\text{RF}}\mathbf{F}_{\text{BB}}[k]\mathbf{s}[k]$, where $\mathbf{s}[k]$ denotes the transmitted symbol sequence at the k^{th} subcarrier of size $N_s \times 1$.

1) *Channel Model:* We consider a frequency-selective MIMO channel between the TX and the receiver (RX), with a delay tap length of N_c in the time domain. The d^{th} delay tap of the channel is denoted by an $N_r \times N_t$ matrix \mathbf{H}_d , $d = 0, 1, \dots, N_c - 1$. Assuming a geometric channel model [11], \mathbf{H}_d can be written as

$$\mathbf{H}_d = \sqrt{\frac{N_t N_r}{L \rho_L}} \sum_{\ell=1}^L \alpha_\ell p_{\text{rc}}(dT_s - \tau_\ell) \mathbf{a}_R(\phi_\ell) \mathbf{a}_T^*(\theta_\ell), \quad (1)$$

where ρ_L represents the path loss between the TX and the RX; L corresponds to the number of channel paths; T_s represents the sampling period; $p_{\text{rc}}(\tau)$ denotes a filter that incorporates the effects of pulse-shaping and other lowpass filtering evaluated at τ ; $\alpha_\ell \in \mathbb{C}$ is the complex gain of the ℓ^{th} path; $\tau_\ell \in \mathbb{R}$ is the delay of the ℓ^{th} path; $\phi_\ell \in [0, 2\pi]$ and $\theta_\ell \in [0, 2\pi]$ are the AoA and AoD of the ℓ^{th} path, respectively; and $\mathbf{a}_R(\phi_\ell) \in \mathbb{C}^{N_r \times 1}$ and $\mathbf{a}_T(\theta_\ell) \in \mathbb{C}^{N_t \times 1}$ are the array steering vectors for the receive and transmit antennas, respectively. Both the TX and the RX are assumed to use Uniform Linear Arrays (ULAs) with half-wavelength separation such that the n^{th} and the m^{th} elements of these array steering vectors are given by, respectively:

$$\begin{aligned} [\mathbf{a}_T(\theta_\ell)]_n &= \sqrt{\frac{1}{N_t}} e^{jn\pi \cos(\theta_\ell)}, \quad n = 0, \dots, N_t - 1, \\ [\mathbf{a}_R(\phi_\ell)]_m &= \sqrt{\frac{1}{N_r}} e^{jm\pi \cos(\phi_\ell)}, \quad m = 0, \dots, N_r - 1. \end{aligned}$$

The channel can be expressed more compactly in the following form: $\mathbf{H}_d = \mathbf{A}_R \mathbf{\Delta}_d \mathbf{A}_T^*$ where $\mathbf{\Delta}_d \in \mathbb{C}^{L \times L}$ is diagonal with non-zero complex diagonal entries, and $\mathbf{A}_R \in \mathbb{C}^{N_r \times L}$ and $\mathbf{A}_T \in \mathbb{C}^{N_t \times L}$ contain the receive and transmit array steering vectors $\mathbf{a}_R(\phi_i)$ and $\mathbf{a}_T(\theta_i)$, respectively. The channel at subcarrier k can be written in terms of the different delay taps as

$$\mathbf{H}[k] = \sum_{d=0}^{N_c-1} \mathbf{H}_d e^{-j \frac{2\pi k d}{N}} = \mathbf{A}_R \mathbf{\Delta}[k] \mathbf{A}_T^*. \quad (2)$$

where $\mathbf{\Delta}[k] \in \mathbb{C}^{L \times L}$ is diagonal with non-zero complex diagonal entries such that $\mathbf{\Delta}[k] = \sum_{d=0}^{N_c-1} \mathbf{\Delta}_d e^{-j \frac{2\pi k d}{N}}$, $k = 0, \dots, K-1$.

2) *Extended Virtual Channel Model*: According to [2], we can further approximate the channel \mathbf{H}_d using the extended virtual channel model as $\mathbf{H}_d \approx \tilde{\mathbf{A}}_R \mathbf{\Delta}_d^v \tilde{\mathbf{A}}_T^*$, where $\mathbf{\Delta}_d^v \in \mathbb{C}^{G_r \times G_t}$ corresponds to a sparse matrix that contains the path gains in the non-zero elements. Moreover, the dictionary matrices $\tilde{\mathbf{A}}_T$ and $\tilde{\mathbf{A}}_R$ contain the TX and RX array response vectors evaluated on a grid of size $G_r \gg L$ for the AoA and a grid of size $G_t \gg L$ for the AoD, i.e., $\tilde{\theta}_\ell \in \{0, \frac{2\pi}{G_r}, \dots, \frac{2\pi(G_r-1)}{G_r}\}$ and $\tilde{\phi}_\ell \in \{0, \frac{2\pi}{G_t}, \dots, \frac{2\pi(G_t-1)}{G_t}\}$, respectively:

$$\tilde{\mathbf{A}}_T = [\mathbf{a}_T(\tilde{\theta}_1) \dots \mathbf{a}_T(\tilde{\theta}_{G_t})], \quad (3)$$

$$\tilde{\mathbf{A}}_R = [\mathbf{a}_R(\tilde{\phi}_1) \dots \mathbf{a}_R(\tilde{\phi}_{G_r})]. \quad (4)$$

Since we have few scattering clusters in mmWave channels, the sparse assumption for $\mathbf{\Delta}_d^v \in \mathbb{C}^{G_r \times G_t}$ is commonly accepted. To help expose the sparse structure, we can express the channel at subcarrier k in terms of the sparse matrices $\mathbf{\Delta}_d^v$ and the dictionaries as follows

$$\mathbf{H}[k] \approx \tilde{\mathbf{A}}_R \left(\sum_{d=0}^{N_c-1} \mathbf{\Delta}_d^v e^{-j \frac{2\pi k d}{N}} \right) \tilde{\mathbf{A}}_T^* \approx \tilde{\mathbf{A}}_R \mathbf{\Delta}^v[k] \tilde{\mathbf{A}}_T^*. \quad (5)$$

where $\mathbf{\Delta}[k] = \sum_{d=0}^{N_c-1} \mathbf{\Delta}_d^v e^{-j \frac{2\pi k d}{N}}$, $k = 0, \dots, K-1$, is a $G_r \times G_t$ complex sparse matrix containing the channel gains of the virtual channel.

3) *Signal Reception*: Considering that the receiver (RX) applies a hybrid combiner $\mathbf{W}[k] = \mathbf{W}_{RF} \mathbf{W}_{BB}[k] \in \mathbb{C}^{N_r \times N_s}$, the received signal at subcarrier k , during the data transmission phase, can be expressed as

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

where $\mathbf{n}[k] \sim \mathcal{CN}(0, \sigma^2 \mathbf{I})$ corresponds to the circularly symmetric complex Gaussian distributed additive noise vector. As will be discussed in Section III, during the channel training phase, frequency-flat training precoders and combiners will be considered to reduce complexity.

B. Problem Formulation

During the training phase, transmitter and receiver use a training precoder $\mathbf{F}_t^{(m)} \in \mathbb{C}^{N_t \times L_t}$ and a training combiner $\mathbf{W}_t^{(m)} \in \mathbb{C}^{N_r \times L_r}$ for the m^{th} pilot training frame, respectively. The precoders and combiners considered in this phase are frequency-flat to keep the complexity of the sparse recovery algorithms low. The transmitted symbols are assumed to

satisfy $\mathbb{E}\{\mathbf{s}^{(m)}[k] \mathbf{s}^{(m)*}[k]\} = \frac{P}{N_s} \mathbf{I}_{N_s}$, where P is the total transmitted power and $N_s = L_t$. The transmitted symbol $\mathbf{s}^{(m)}[k]$ is decomposed as $\mathbf{s}^{(m)}[k] = \mathbf{q}^{(m)} \mathbf{t}^{(m)}[k]$, with $\mathbf{q}^{(m)} \in \mathbb{C}^{L_t \times 1}$ is a frequency-flat vector and $\mathbf{t}^{(m)}[k]$ is a pilot symbol known at the receiver. This decomposition is used to reduce computational complexity since it allows simultaneous use of the L_t spatial degrees of freedom coming from L_t RF chains and enables channel estimation using a single subcarrier-independent measurement matrix. Moreover, each entry in $\mathbf{F}_t^{(m)}$ and in $\mathbf{W}_t^{(m)}$ are normalized such that their squared-modulus would be $\frac{1}{N_t}$ and $\frac{1}{N_r}$, respectively. Then, the received samples in the frequency domain for the m^{th} training frame can be expressed as

$$\mathbf{y}^{(m)}[k] = \mathbf{W}_t^{(m)*} \mathbf{H}[k] \mathbf{F}_t^{(m)} \mathbf{q}^{(m)} \mathbf{t}^{(m)}[k] + \mathbf{n}_c^{(m)}[k], \quad (6)$$

where $\mathbf{H}[k] \in \mathbb{C}^{N_r \times N_t}$ denotes the frequency-domain MIMO channel response at the k^{th} subcarrier and $\mathbf{n}_c^{(m)}[k] \in \mathbb{C}^{L_r \times 1}$, $\mathbf{n}_c^{(m)}[k] = \mathbf{W}_t^{(m)*} \mathbf{n}^{(m)}[k]$, represents the frequency-domain combined noise vector received at the k^{th} subcarrier. The average received SNR is given by $\text{SNR} = \frac{P}{\rho_t \sigma^2}$. Furthermore, the channel coherence time is assumed to be larger than the frame duration and that the same channel can be considered for several consecutive frames.

1) *Measurement Matrix*: In order to apply sparse reconstruction with a single subcarrier-independent measurement matrix, we first remove the effect of the scalar $\mathbf{t}^{(m)}[k]$ by multiplying the received signal by $\mathbf{t}^{(m)}[k]^{-1}$. Using the following property $\text{vec}\{\mathbf{A}\mathbf{X}\mathbf{C}\} = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}\{\mathbf{X}\}$, the vectorized received signal is given by

$$\text{vec}\{\mathbf{y}^{(m)}[k]\} = (\mathbf{q}^{(m)T} \mathbf{F}_t^{(m)T} \otimes \mathbf{W}_t^{(m)*}) \text{vec}\{\mathbf{H}[k]\} + \mathbf{n}_c^{(m)}[k]. \quad (7)$$

The vectorized channel matrix can be expressed as

$$\text{vec}\{\mathbf{H}[k]\} = (\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R) \text{vec}\{\mathbf{\Delta}^v[k]\}. \quad (8)$$

Furthermore, we define the measurement matrix $\mathbf{\Phi}^{(m)} \in \mathbb{C}^{L_r \times N_t N_r}$: $\mathbf{\Phi}^{(m)} = (\mathbf{q}^{(m)T} \mathbf{F}_t^{(m)T} \otimes \mathbf{W}_t^{(m)*})$, and the dictionary $\mathbf{\Psi} \in \mathbb{C}^{N_t N_r \times G_t G_r}$ as $\mathbf{\Psi} = (\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R)$. Then, the vectorized received pilot signal $L_r \times 1$ at the m^{th} training symbol can be written as

$$\text{vec}\{\mathbf{y}^{(m)}[k]\} = \mathbf{\Phi}^{(m)} \mathbf{\Psi} \mathbf{h}^v[k] + \mathbf{n}_c^{(m)}[k], \quad (9)$$

where $\mathbf{h}^v[k] = \text{vec}\{\mathbf{\Delta}^v[k]\} \in \mathbb{C}^{G_r G_t \times 1}$ is the sparse vector containing the complex channel gains. Moreover, we use several training frames to get enough measurements and accurately reconstruct the sparse vector $\mathbf{h}^v[k]$, especially in the very-low SNR regime. Therefore, when the transmitter and receiver communicate during M training steps using different pseudorandomly built precoders and combiners, (9) can be extended to M received signals given by

$$\mathbf{y}[k] = \mathbf{\Phi} \mathbf{\Psi} \mathbf{h}^v[k] + \mathbf{n}_c[k]. \quad (10)$$

where $\mathbf{y}[k] = [\mathbf{y}^{(1)}[k], \dots, \mathbf{y}^{(M)}[k]]^T$ is of size $ML_r \times 1$, $\mathbf{\Phi} = [\mathbf{\Phi}^{(1)}, \dots, \mathbf{\Phi}^{(M)}]^T$ of size $ML_r \times N_t N_r$, and $\mathbf{n}_c[k] = [\mathbf{n}_c^{(1)}[k], \dots, \mathbf{n}_c^{(M)}[k]]^T$ of size $ML_r \times 1$.

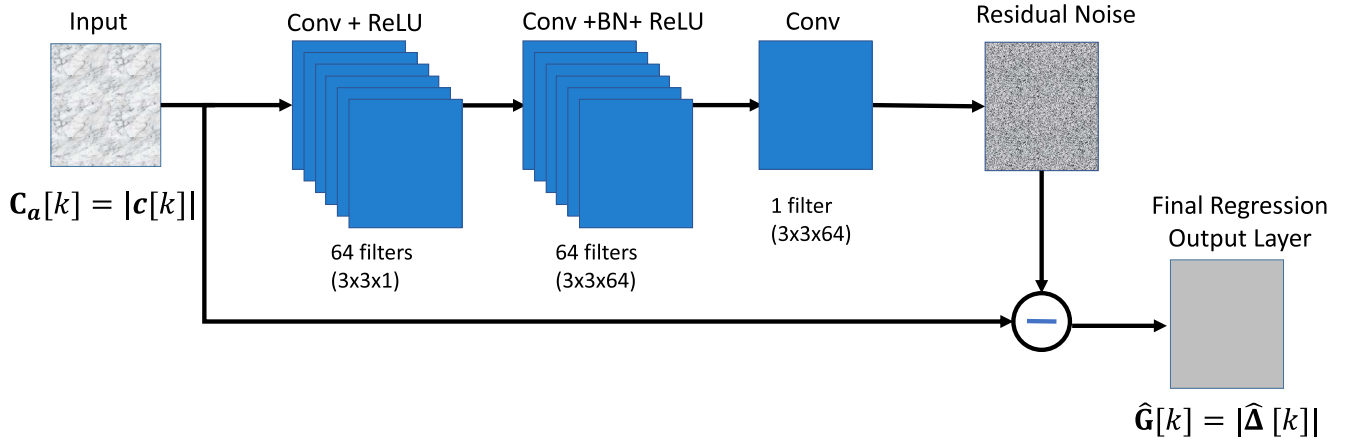


Fig. 2. Proposed denoising convolutional neural network (DnCNN) for multicarrier channel amplitude estimation.

Hence, the vector $\mathbf{h}^v[k]$ can be estimated by solving the sparse reconstruction problem as done in [11],

$$\min \|\mathbf{h}^v[k]\|_1 \quad \text{subject to } \|\mathbf{y}[k] - \Phi\Psi\mathbf{h}^v[k]\|_2^2 < \epsilon, \quad (11)$$

where ϵ represents a tunable parameter defining the maximum error between the reconstructed channel and the received signal. In realistic scenarios, the sparsity (number of channel paths) is usually unknown, therefore the choice of ϵ is critical to solve (11) and estimate the sparsity level. The choice of this parameter is explained in Section III-D.

Interestingly, the matrices in (5) exhibit the same sparse structure for all k , since the AoA and AoD do not change with frequency in the transmission bandwidth. This is an interesting property that can be leveraged when solving the compressed channel estimation problem defined in (11). Moreover, we denote the supports of the virtual channel matrices Δ_d^v as $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_{N_c-1}$, $d = 0, \dots, N_c - 1$. Then, knowing $\mathbf{h}^v[k] = \text{vec}\{\Delta^v[k]\}$, with $\Delta^v[k] = \sum_{d=0}^{N_c-1} \Delta_d^v e^{-j\frac{2\pi k}{N}d}$, $k = 0, \dots, K - 1$, the supports of $\mathbf{h}^v[k]$ are defined as

$$\text{supp}\{\mathbf{h}^v[k]\} = \bigcup_{d=0}^{N_c-1} \text{supp}\{\text{vec}\{\Delta_d^v\}\} \quad k = 0, \dots, K - 1, \quad (12)$$

where the union of the supports of the time-domain virtual channel matrices is due to the additive nature of the Fourier transform. Therefore, as shown in (12), where the union is independent of the subcarrier k , $\Delta[k]$ has the same supports for all k .

2) *Correlation Matrix*: To estimate multi-path components of the channel, i.e., AoAs/AoDs and channel gains, we first need to compute the atom, which is defined as the vector that produces the largest sum-correlation with the received signals in the measurement matrix. The sum-correlation is especially considered as the support of the different sparse vectors is the same over the K subcarriers. The correlation vector $\mathbf{c}[k] \in \mathbb{C}^{G_r G_t}$ is given by

$$\mathbf{c}[k] = \Upsilon^* \mathbf{y}[k], \quad (13)$$

where $\Upsilon \in \mathbb{C}^{ML_r \times G_t G_r}$, $\Upsilon = \Phi\Psi$ represents the equivalent measurement matrix which is the same $\forall k$

and $\mathbf{y}[k] \in \mathbb{C}^{ML_r \times 1}$ is the received signal for a given k , $k = 0, \dots, K - 1$.

One can note that if there exists a correlation between noise components, the atom estimated from the projection in (13) might not be the correct one. In order to compensate for this error in estimation, we consider the noise covariance matrix when performing the correlation step. In particular, we consider two arbitrary (hybrid) combiners $\mathbf{W}_{tr}^{(m)(i)}$, $\mathbf{W}_{tr}^{(m)(j)} \in \mathbb{C}^{N_r \times L_r}$ for two arbitrary training steps i, j and a given subcarrier k . Hence, the combined noise at a given training step i and subcarrier k is represented as $\mathbf{n}_c^{(i)}[k] = \mathbf{W}_{tr}^{(i)*} \mathbf{n}^{(i)}[k]$, with $\mathbf{n}^{(i)}[k] \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{L_r})$, which results in noise cross-covariance matrix given by $\mathbb{E}\{\mathbf{n}_c^{(i)}[k] \mathbf{n}_c^{(j)*}[k]\} = \mathbf{W}_{tr}^{(i)*} \sigma^2 \delta[i - j] \mathbf{W}_{tr}^{(j)}$. We can further write the noise covariance matrix of $\mathbf{y}[k]$ as a block diagonal matrix $\mathbf{C}_w \in \mathbb{C}^{ML_r \times ML_r}$,

$$\mathbf{C}_w = \text{blkdiag}\{\mathbf{W}_{tr}^{(1)*} \mathbf{W}_{tr}^{(1)}, \dots, \mathbf{W}_{tr}^{(M)*} \mathbf{W}_{tr}^{(M)}\}. \quad (14)$$

Moreover, Cholesky factorization can be used to factorize \mathbf{C}_w into $\mathbf{C}_w = \mathbf{D}_w^* \mathbf{D}_w$, where $\mathbf{D}_w \in \mathbb{C}^{ML_r \times ML_r}$ is an upper triangular matrix. Then, by taking into consideration the noise covariance matrix, the correlation step is given by

$$\mathbf{c}[k] = \Upsilon_w^* \mathbf{y}_w[k], \quad (15)$$

where $\Upsilon_w \in \mathbb{C}^{ML_r \times G_t G_r}$ represents the whitened measurement matrix given by $\Upsilon_w = \mathbf{D}_w^{-*} \Upsilon$. And, the $ML_r \times 1$ whitened received signal $\mathbf{y}_w[k]$ is given by $\mathbf{y}_w[k] = \mathbf{D}_w^{-*} \mathbf{y}[k]$. The matrix $\mathbf{D}_w^{-1} \in \mathbb{C}^{ML_r \times ML_r}$ is given by $\mathbf{D}_w^{-1} = \text{blkdiag}\left\{\left(\mathbf{D}_w^{(1)}\right)^{-1}, \dots, \left(\mathbf{D}_w^{(M)}\right)^{-1}\right\}$, where $\left(\mathbf{D}_w^{(m)}\right)^{-1}$ can be considered as a frequency-flat baseband combiner $\mathbf{W}_{BB,tr}^{(m)}$ used in the m -th training step. Therefore, by applying the whitened measurement matrix, the resulting correlation would simultaneously whiten the spatial noise components and estimate a more accurate support index in the sparse vectors $\mathbf{h}^v[k]$.

III. DEEP LEARNING AND COMPRESSIVE-SENSING BASED CHANNEL ESTIMATION (DL-CS-CE)

To solve the CS channel estimation problem formulated in (11), this section proposes two DL-based algorithms.

Both leverage the common support between the channel matrices for every subcarrier and provide different complexity-performance trade-offs. The former simultaneously estimate the support using an offline-trained DnCNN and then reconstruct the channel. On the other hand, the latter applies further fine-tuning to accurately estimate the AoAs and AoDs with higher resolution dictionary matrices while keeping computational complexity low.

A. Offline Training and Online Deployment of DnCNN

Before delving into the proposed solutions' details, let us first provide insights into the considered DnCNN architecture as well as its offline training and online deployment.

1) *DnCNN Architecture*: Fig. 2 illustrates the network architecture of the DnCNN denoiser that consists of L_C convolutional (Conv) layers. Each layer uses $c_{\text{CL}}^{(l)}$ different $D_x^{(l)} \times D_y^{(l)} \times D_z^{(l)}$ filters. The first convolutional layer is followed by a rectified linear unit (ReLU). The succeeding $L_C - 2$ convolutional layers are followed by batch-normalization (BN) and a ReLU. The final L_C^{th} convolutional layer uses one separate $D_x^{(L_C)} \times D_y^{(L_C)} \times D_z^{(L_C)}$ filter to reconstruct the signal. Here, $D_x^{(l)}$, $D_y^{(l)}$ and $D_z^{(l)}$ are the convolutional kernel dimensions, and $c_{\text{CL}}^{(l)}$ is the number of filters in the l^{th} layer.

We present three pseudo-color images of the noisy channel, residual noise, and estimated output channel in Fig. 2. The DnCNN considers the amplitude of the correlation $G_r \times G_t$ matrix, i.e.,

$$\mathbf{C}_\alpha[k] = \text{vec2mat}(|\mathbf{c}[k]|, [G_r, G_t]), \quad \forall k, \quad (16)$$

as input (where $\mathbf{c}[k]$ is defined in (15)) produces residual noise as an output, rather than estimated channel amplitudes, where we define a $G_r \times G_t$ matrix of channel amplitudes as

$$\mathbf{G}[k] = |\Delta^v[k]| \in \mathbb{R}^{G_r \times G_t}, \quad \forall k. \quad (17)$$

The DnCNN aims to learn a mapping function $\mathcal{F}(\mathbf{C}_\alpha[k]) = \mathbf{G}[k]$ to predict the latent clean image from noisy observation $\mathbf{C}_\alpha[k]$. We adopt the residual learning formulation to train a residual mapping $\mathcal{R}(\mathbf{C}_\alpha[k]) \approx \mathbf{V}$ where \mathbf{V} is the residual noise, and then we have $\mathbf{G}[k] = \mathbf{C}_\alpha[k] - \mathcal{R}(\mathbf{C}_\alpha[k])$. Instead of learning a mapping directly from a noisy image to a denoised image, learning the residual noise is beneficial [34], [35]. Furthermore, the averaged mean squared error between the desired residual images and estimated ones from noisy input is adopted as the loss function to learn the trainable parameters Θ of the DnCNN. This loss function is given by

$$\ell(\Theta) = \frac{1}{2N} \sum_{i=1}^N \|\mathcal{R}(\mathbf{C}_\alpha[k]^i; \Theta) - (\mathbf{C}_\alpha[k]^i - \mathbf{G}[k]^i)\|_F^2 \quad (18)$$

where $(\mathbf{C}_\alpha[k]^i, \mathbf{G}[k]^i)_{i=1}^N$ represents N noisy-clean training patch pairs. This method is also known as residual learning [35] and renders the DnCNN to remove the highly structured natural image rather than the unstructured noise. Consequently, residual learning improves both the training times and accuracy of a network. In this way, combining batch normalization and residual learning techniques can accelerate

the training speed and improve the denoising performance. Besides, batch normalization has been shown to offer some merits for residual learning, such as alleviating internal covariate shift problem in [23], [34].

2) *Offline Training of the DnCNN*: During offline training of the DnCNN, the dataset of $\mathbf{C}_\alpha[k], \forall k$ and $\mathbf{G}[k], \forall k$ is generated based on the realistic Raymobtime dataset for mmWave frequency selective channel environment. With the mmWave channel amplitude in (17) and the correlation of the received signals and the measurement matrix in (16), the training data of $\mathbf{C}_\alpha[k]$ and $\mathbf{G}[k]$ can be obtained. In particular, the process to obtain $\mathbf{C}_\alpha[k]$ and $\mathbf{G}[k]$ involves the following four steps: i) generation of channel matrices based on the mmWave channel model from the Raymobtime dataset ii) obtaining $\mathbf{G}[k]$ based on (17); iii) computing the whitened received signal vector $\mathbf{y}_w[k] \forall k$; and iv) acquiring the amplitudes of the correlation vector $\mathbf{c}[k]$ as per (15) and transforming it into a matrix form $\mathbf{C}_\alpha[k]$ as per (16).

3) *Online Deployment of the DnCNN*: During the online deployment of the DL-CS-CE, we obtain the measured received signal $\mathbf{y}_w[k]$ from the realistic mmWave channel environments. We compute $\mathbf{C}_\alpha[k]$ based on (16), which is then fed to the offline-trained DnCNN. Then, the trained DnCNN would predict $\hat{\mathbf{G}}[k]$, from which we can estimate the supports of $\Delta^v[k]$. An interesting and noteworthy issue is that we can feed the trained DnCNN a subset K_p of K subcarriers of the amplitudes of the correlation matrices $\mathbf{C}_\alpha[k]$, to eventually estimate the support of $\Delta^v[k]$, since as shown in Section II-B1 $\Delta^v[k]$ have the same support for all k . In particular, the support can be estimated if a small number of subcarriers $K_p \ll K$ is used instead. This will eliminate the need for computing $\mathbf{C}_\alpha[k]$ for all subcarriers and eventually reduce the overall computational complexity at the cost of a negligible performance degradation. By leveraging from triangle inequality, $\|\mathbf{y}[k]\|_2^2 \leq \|\Phi \mathbf{h}^v[k]\|_2^2 + \|\mathbf{n}_c[k]\|_2^2$, such that the K_p selected signals are expected to exhibit the strongest channel response. Therefore, the K_p subcarriers having largest ℓ_2 -norm will be exploited to derive an estimate of the support of the already defined sparse channel matrix $\Delta^v[k]$, $k = 0, \dots, K - 1$.

B. Algorithm 1: DL-CS-CE

The state-of-the-art sparse channel estimation schemes [11, and references therein] depend on greedy algorithms to detect the supports sequentially, which naturally yield suboptimal solutions. This motivated us to exploit the neural networks to estimate all supports simultaneously rather than sequentially. In addition, the proposed DL-CS-CE algorithm exploits the common support between the different channel matrices associated with different subcarriers. Since the DL-CS-CE processes OFDM signals simultaneously received from K subcarriers, the amount of obtained information is K times larger than a single carrier system [11]. Accordingly, exploiting common supports across multiple carriers increases the likelihood of estimating AoA/AoD, or equivalently the support of channel. The algorithmic implementation of the proposed DL-CS-CE solution is presented in Algorithm 1. After initialization steps between lines 1-3 and the computation of the

Algorithm 1 DL-CS-CE

Input: $\mathbf{y}[k]$, Φ , Ψ , $\tilde{\mathbf{A}}_T$, $\tilde{\mathbf{A}}_R$, K_p , ϵ

- 1: $\mathbf{y}_w[k] \leftarrow \mathbf{D}_w^{-*} \mathbf{y}[k] \quad \forall k$
- 2: $\mathbf{r}[k] \leftarrow \mathbf{y}_w[k] \quad \forall k$
- 3: $\hat{\mathcal{T}}, \mathcal{K} \leftarrow \{\emptyset\}$
- 4: $\Upsilon_w \leftarrow \mathbf{D}_w^{-*} \Phi \Psi$
- 5: $\mathcal{K} \leftarrow \text{FIND STRONGEST SUBCARRIERS}(\mathbf{y}[k])$
- 6: $\hat{\mathbf{g}}[k] \leftarrow \text{ESTIMATE AMPLITUDES}(\Upsilon_w^*, \mathbf{r}[k], \mathcal{K})$
- 7: $\hat{\mathbf{H}}[k] \leftarrow \text{RECONSTRUCT CHANNEL}(\hat{\mathbf{g}}[k])$

return $\hat{\mathbf{H}}[k]$

- 8: **procedure** FIND STRONGEST SUBCARRIERS($\mathbf{y}[k]$)
- 9: **for** $i = 1 : K_p$ **do**

$$\mathcal{K} = \mathcal{K} \cup \arg \max_{k \notin \mathcal{K}} \|\mathbf{y}[k]\|_2^2$$
- 10: **end for**
- 11: **return** \mathcal{K}
- 12: **end procedure**
- 13: **procedure** ESTIMATE AMPLITUDES($\Upsilon_w^*, \mathbf{r}[k], \mathcal{K}$)
- 14: $\mathbf{c}[k] \leftarrow \Upsilon_w^* \mathbf{r}[k]$, $k \in \mathcal{K}$ // as per (15)
- 15: $\mathbf{C}_\alpha[k] \leftarrow \text{vec2mat}([\mathbf{c}[k]], [G_r, G_t])$ // as per (16)
- 16: $\hat{\mathbf{G}}[k] \leftarrow \overset{\text{Online}}{\underset{\text{DnCNN}}{\mathbf{C}_\alpha[k]}}$ // [c.f. Fig. 3.b]
- 17: $\hat{\mathbf{g}}[k] \leftarrow \text{vec}(\hat{\mathbf{G}}[k])$ // as per (19)
- 18: **return** $\hat{\mathbf{g}}[k]$, $\forall k \in \mathcal{K}$
- 19: **end procedure**
- 20: **procedure** RECONSTRUCT CHANNEL($\hat{\mathbf{g}}[k]$, $\forall k$)
- 21: MSE $\leftarrow \infty$
- 22: $i \leftarrow 1$
- 23: $\mathcal{I} \leftarrow \text{INDEXSORTDESCEND}(\sum_{k \in \mathcal{K}} |\hat{\mathbf{g}}[k]|)$
- 24: **while** MSE $> \epsilon$ & $i \leq G_t G_r$ **do**
- 25: $\hat{\mathcal{T}} \leftarrow \hat{\mathcal{T}} \cup \mathcal{I}(i)$
- 26: $\hat{\xi}[k] \leftarrow \left([\Upsilon_w]_{:, \hat{\mathcal{T}}}\right)^\dagger \mathbf{y}_w[k]$, $\forall k$
- 27: $\mathbf{r}[k] \leftarrow \mathbf{y}_w[k] - [\Upsilon_w]_{:, \hat{\mathcal{T}}} \hat{\xi}[k]$, $\forall k$
- 28: MSE $\leftarrow \frac{1}{KML_r} \sum_{k=0}^{K-1} \mathbf{r}^* [k] \mathbf{r}[k]$
- 29: $i \leftarrow i + 1$
- 30: **end while**
- 31: $\hat{L} \leftarrow i$ // Estimate # paths [c.f. Section III-D]
- 32: $\hat{\mathbf{h}}^v[k] \leftarrow$ as per (23).
- 33: $\text{vec}\{\hat{\Delta}^v[k]\} \leftarrow \hat{\mathbf{h}}^v[k]$
- 34: $\text{vec}\{\hat{\mathbf{H}}[k]\} \leftarrow (\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R) \text{vec}\{\hat{\Delta}^v[k]\}$.
- 35: **return** $\hat{\mathbf{H}}[k]$
- 36: **end procedure**

whitened equivalent observation matrix in line 4, DL-CS-CE is structured based on three main procedures:

- Estimation of the channel amplitudes by using an offline-trained DnCNN,
- Sorting the estimated channel amplitudes in descending order to select the supports of dominant entries,
- Reconstruction of the channel according to the selected indices,

which are explained in the sequel.

1) *Strongest Subcarriers Selection*: This procedure is represented in lines 8-11 of Algorithm 1, where the algorithm

TABLE II
AVERAGE SIZE OF ESTIMATED SUPPORT $\hat{L} = |\hat{\mathcal{T}}|$

SNR	-15 dB	-10 dB	-5 dB	-0 dB	5 dB
\hat{L}	4	5	9	12	15

iteratively finds a subset $\mathcal{K} \in K$ containing the K_p strongest subcarriers which are expected to exhibit the strongest channel response as explained in Section III-A3.

2) *Amplitude Estimation*: As depicted in Fig. 3, the lines 13 and 14 of Algorithm 1 first compute the correlation vector as per (15) and then create the DnCNN input $\mathbf{C}_\alpha[k]$ by putting correlation vectors into a matrix form as per (16), respectively. In line 15, the offline trained DnCNN is used as the kernel of the channel amplitude estimation to obtain the DnCNN output $\hat{\mathbf{G}}[k]$ of size $G_r \times G_t$, which is the estimate of $\mathbf{G}[k]$ given in (17). It is worth noting that we only use a subset \mathcal{K} of the correlation matrices $\mathbf{C}_\alpha[k] \forall k \in \mathcal{K}$ as an input to the DnCNN. In line 16, the output channel amplitude estimation matrix $\hat{\mathbf{G}}[k]$ is then vectorized into the following $G_t G_r \times 1$ vector form

$$\hat{\mathbf{g}}[k] = \text{vec}(\hat{\mathbf{G}}[k]), \quad \forall k \in \mathcal{K} \quad (19)$$

where the indices of the maximum amplitudes of $\hat{\mathbf{g}}[k]$ will be exploited for support detection.

3) *Multicarrier Channel Reconstruction*: This procedure corresponds to the last block depicted in the last stage of the block diagram in Fig.3.b. It detects supports by iteratively updating residual until the MSE falls below a predetermined threshold, ϵ . After initialization steps in lines 19 and 20, line 19 first sums the amplitudes of predicted $\hat{\mathbf{g}}[k]$ over the subcarriers $k \in \mathcal{K}$ as the supports are the same for all k [c.f. Section II-B1]. Then, INDEXSORTDESCEND function sorts the sum vector in descending order and return corresponding index set \mathcal{I} , $|\mathcal{I}| = G_r G_t$. Thereafter, the while loop between lines 22 and 28 follows the below steps until the termination condition is satisfied:

Line 23 updates the detected support set $\hat{\mathcal{T}}$ by adding the i^{th} element of ordered index set \mathcal{I} . Then, line 24 projects the input signal $\mathbf{y}_w[k] \forall k$ onto the subspace given by the detected support \mathcal{T} using Weighted Least-Squares (WLS) $\left([\Upsilon_w]_{:, \hat{\mathcal{T}}}\right)^\dagger$, which is followed by residual update and MSE computation in lines 25 and 26, respectively. It is also worth noting that $\left([\Upsilon_w]_{:, \hat{\mathcal{T}}}\right)^\dagger$ corresponds to a WLS estimator, with the corresponding weights given by the inverse noise covariance matrix. Lastly, line 26 increments the loop index i for the next iteration. The final value of $i = |\hat{\mathcal{T}}|$ provides us with one of the key parameters: \hat{L} , the estimate of the sufficient number of paths that guarantees $\text{MSE} > \epsilon$, i.e., L . Thereby, it is closely tied with the choice of ϵ , which will be explained in details in Section III-D. We should also note that the while loop is terminated by the $\text{MSE} > \epsilon$ condition almost all the time since $G_t G_r \gg \hat{L}$ as shown in Table II.³

³This assumption holds since mmWave channels are known to have limited number of paths.

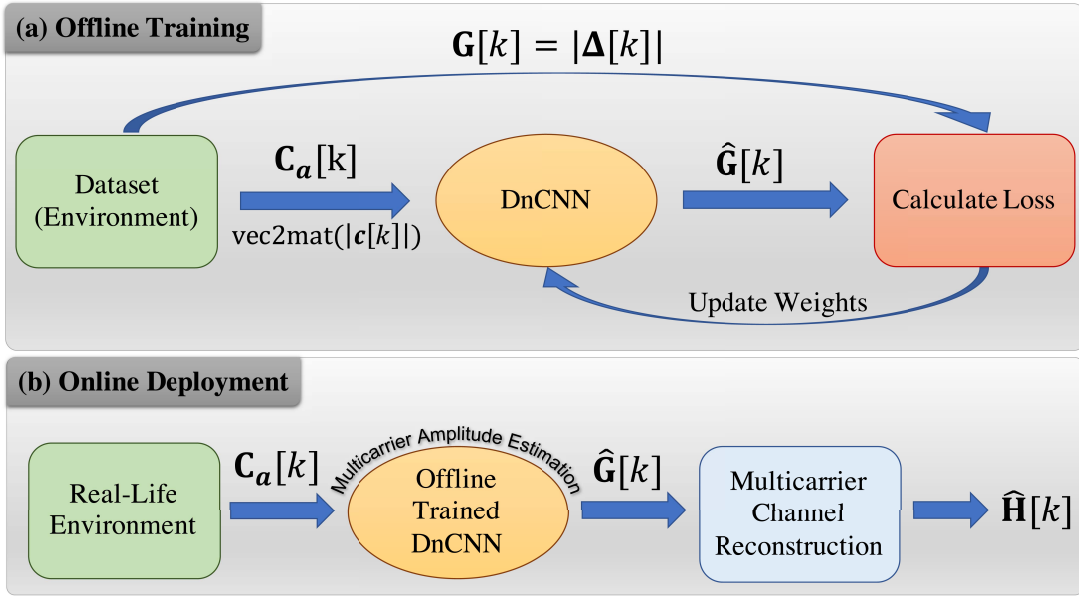


Fig. 3. Block diagram of the DL-CS-CE Scheme: offline training and online deployment.

After estimating the indices of the channel supports where they represent the indices of AoA and AoD pairs for each path, we use them to reconstruct the channel based on the virtual channel model representation. Since the support of sparse channel vectors is already represented by $\hat{\mathcal{T}}$, the measurement matrix can now be defined as $[\mathbf{Y}]_{:, \hat{\mathcal{T}}} \in \mathbb{C}^{M L_r \times \hat{L}}$ such that $[\mathbf{Y}]_{:, \hat{\mathcal{T}}} = [\Phi \Psi]_{:, \hat{\mathcal{T}}}$. Hence, the received signal model for the k^{th} subcarrier can be rewritten as

$$\mathbf{y}[k] = [\mathbf{Y}]_{:, \hat{\mathcal{T}}} \tilde{\boldsymbol{\xi}}[k] + \tilde{\mathbf{n}}_c[k], \quad (20)$$

where $\tilde{\mathbf{n}}_c[k] \in \mathbb{C}^{M L_r \times 1}$ represents the residual noise after estimating the channel support and $\tilde{\boldsymbol{\xi}}[k] \in \mathbb{C}^{\hat{L} \times 1}$ is the vector containing the channel gains to be estimated after sparse recovery. If the support estimation is accurate enough, $\tilde{\mathbf{n}}_c[k]$ will be approximately similar to the post-combining noise vector $\mathbf{n}_c[k]$ [11]. It is important to remark that the indices obtained by the trained DnCNN may be different from the actual channel support. In this case, the support detected $\hat{\mathcal{T}}$ may also be different from the actual support. Likewise, the channel gains to be estimated $\tilde{\boldsymbol{\xi}}[k]$, can also be different from actual vector, $\boldsymbol{\xi}[k] = \text{vec}\{\text{diag}\{\boldsymbol{\Delta}[k]\}\}$.

The mathematical model in (20) is usually considered as the General Linear Model (GLM), where the solution of $\tilde{\boldsymbol{\xi}}[k]$ for real parameters is provided in [42]. For the case with complex-valued parameters, the solution is straightforward and given by

$$\hat{\boldsymbol{\xi}}[k] = \left([\mathbf{Y}]_{:, \hat{\mathcal{T}}}^* \mathbf{C}_w^{-1} [\mathbf{Y}]_{:, \hat{\mathcal{T}}} \right)^{-1} [\mathbf{Y}]_{:, \hat{\mathcal{T}}}^* \mathbf{C}_w^{-1} \mathbf{y}[k], \quad (21)$$

which can be further reduced to

$$\hat{\boldsymbol{\xi}}[k] = \left([\mathbf{Y}_w]_{:, \hat{\mathcal{T}}} \right)^\dagger \mathbf{y}_w[k]. \quad (22)$$

Therefore, $\hat{\boldsymbol{\xi}}[k]$ is considered as the Minimum Variance Unbiased (MVU) estimator for the complex parameter vector $\boldsymbol{\xi}[k]$, $k = 0, \dots, K - 1$. Hence, it is unbiased and attains the

Cramér-Rao Lower Bound (CRLB) if the support is correctly estimated [11].⁴

Once all the supports are detected, line 29 computes the sparse channel vector $\hat{\mathbf{h}}^v[k]$ where its non-zero elements are obtained according to

$$[\hat{\mathbf{h}}^v[k]]_{\hat{\mathcal{T}}} = \left([\mathbf{Y}_w]_{:, \hat{\mathcal{T}}} \right)^\dagger \mathbf{y}_w[k]. \quad (23)$$

Finally, knowing that $\text{vec}\{\hat{\boldsymbol{\Delta}}^v[k]\} = \hat{\mathbf{h}}^v[k]$, line 32 reconstructs the channel based on (8) as follows

$$\text{vec}\{\hat{\mathbf{H}}[k]\} = (\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R) \text{vec}\{\hat{\boldsymbol{\Delta}}^v[k]\}. \quad (24)$$

C. Algorithm 2: Refined DL-CS-CE

The sparsity of $\mathbf{h}^v[k]$ can be impaired by channel power leakage caused by the limited resolution of the chosen dictionary matrices [44]. Therefore, even though the DL-CS-CE provides reasonable AoD/AoA estimates, the adopted virtual quantized dictionary matrices may not obtain the exact AoDs/AoAs, that lie in the off-grid regions of the dictionary in reality. In this section, we combat this issue by developing a method to obtain more accurate AoDs/AoAs. This new procedure is called refined DL-CS-CE and improves NMSE performance of Algorithm 1 (DL-CS-CE) while reducing the incurring computational complexity at the same time. The refined DL-CS-CE presents a refined extension of the DL-CS-CE approach where we first obtain a coarse estimate of the supports from the output of the DnCNN (as done in Algorithm 1). Then, we improve the estimates using developed refining steps, which are explained next.

Using the superscript r for referring to the refining phase, we consider higher resolution refining dictionary matrices

⁴This is considered as Cramér-Rao Lower Bound of a Genie-aided estimation problem, in which the estimator knows the location of the nonzero taps i.e., \mathcal{T} , as if a Genie has aided the estimator with the location of the taps [43].

Algorithm 2 Refined DL-CS-CE

Input: $\mathbf{y}[k]$, Φ , Ψ , $\tilde{\mathbf{A}}_T$, $\tilde{\mathbf{A}}_R$, $\tilde{\mathbf{A}}_T^r$, $\tilde{\mathbf{A}}_R^r$, K_p , ϵ

- 1: $\mathbf{y}_w[k] \leftarrow \mathbf{D}_w^{-*} \mathbf{y}[k] \quad \forall k$
- 2: $\mathbf{r}[k] \leftarrow \mathbf{y}_w[k] \quad \forall k$
- 3: $\hat{\mathcal{T}}, \mathcal{K} \leftarrow \{\emptyset\}$
- 4: $\Phi_w = \mathbf{D}_w^{-*} \Phi$
- 5: $\Psi \leftarrow (\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R) \quad // \text{ For Detection}$
- 6: $\Upsilon_w \leftarrow \mathbf{D}_w^{-*} \Phi \Psi$
- 7: $\Psi^r = (\tilde{\mathbf{A}}_T^r \otimes \tilde{\mathbf{A}}_R^r) \quad // \text{ For Refining}$
- 8: $\Upsilon_w^r \leftarrow \mathbf{D}_w^{-*} \Phi \Psi^r$
- 9: $\mathcal{K} \leftarrow \text{FIND STRONGEST SUBCARRIERS}(\mathbf{y}[k])$
- 10: $\hat{\mathbf{g}}[k] \leftarrow \text{ESTIMATE AMPLITUDES}(\Upsilon_w^*, \mathbf{r}[k], \mathcal{K})$
- 11: $\mathbf{H}[k] \leftarrow \text{RECONSTRUCT CHANNEL \& REFINE}(\hat{\mathbf{g}}[k])$

return $\mathbf{H}[k]$

- 12: **procedure** FIND STRONGEST SUBCARRIERS($\mathbf{y}[k]$)
- 13: Lines 9-10 in Algorithm 1
- 14: **end procedure**
- 15: **procedure** ESTIMATE AMPLITUDES(Υ_w^* , $\mathbf{r}[k]$, \mathcal{K})
- 16: Lines 13-16 in Algorithm 1
- 17: **end procedure**
- 18: **procedure** RECONSTRUCT CHANNEL & REFINE($\hat{\mathbf{g}}[k]$)
- 19: $\mathcal{I} \leftarrow \text{INDEXSORTDESCEND}(\sum_{k \in \mathcal{K}} |\hat{\mathbf{g}}[k]|)$
- 20: $\text{MSE} \leftarrow \infty$
- 21: $i \leftarrow 1$
- 22: **while** $\text{MSE} > \epsilon$ & $i \leq G_t G_r$ **do**
- 23: $[i_{\text{AoA}}^d, i_{\text{AoD}}^d] \leftarrow \text{ind2sub}([G_r, G_t], \mathcal{I}(i))$
- 24: $\hat{\mathcal{T}} \leftarrow \text{REFINE}(i_{\text{AoA}}^d, i_{\text{AoD}}^d)$
- 25: $\hat{\xi}[k] \leftarrow ([\Upsilon_w^r]_{:, \hat{\mathcal{T}}})^\dagger \mathbf{y}_w[k], \forall k$
- 26: $\mathbf{r}[k] \leftarrow \mathbf{y}_w[k] - [\Upsilon_w^r]_{:, \hat{\mathcal{T}}} \hat{\xi}[k], \forall k$
- 27: $\text{MSE} \leftarrow \frac{1}{KML_r} \sum_{k=0}^{K-1} \mathbf{r}^*[k] \mathbf{r}[k]$
- 28: $i \leftarrow i + 1$
- 29: **end while**
- 30: $\hat{L} \leftarrow i \quad // \text{ Estimate \# paths [c.f. Section III-D]}$
- 31: $\hat{\mathbf{h}}^v[k] \leftarrow \text{as per (23) but using } [\Upsilon_w^r]_{:, \hat{\mathcal{T}}} \text{ instead}$
- 32: $\text{vec}\{\hat{\Delta}^v[k]\} \leftarrow \hat{\mathbf{h}}^v[k]$
- 33: $\text{vec}\{\hat{\mathbf{H}}[k]\} \leftarrow \Psi^r \text{vec}\{\hat{\Delta}^v[k]\}.$
- 34: **return** $\mathbf{H}[k]$
- 35: **end procedure**
- 36: **procedure** REFINE($i_{\text{AoA}}^d, i_{\text{AoD}}^d$)
- 37: $i_{\text{AoA}}^r \leftarrow \text{as per in (25)}$
- 38: $i_{\text{AoD}}^r \leftarrow \text{as per (26)}$
- 39: $i_{\text{AoA}}^{r*} \leftarrow \text{as per (28)}$
- 40: $i_{\text{AoD}}^{r*} \leftarrow \text{as per (26) by using } i_{\text{AoA}}^{r*} \text{ instead of } i_{\text{AoA}}^r$
- 41: $j^* \leftarrow \text{sub2ind}([G_r^r, G_t^r], [i_{\text{AoA}}^{r*}, i_{\text{AoD}}^{r*}])$
- 42: $\hat{\mathcal{T}} \leftarrow \hat{\mathcal{T}} \cup j^*$
- 43: **return** $\hat{\mathcal{T}}$
- 44: **end procedure**

$\tilde{\mathbf{A}}_R^r$ and $\tilde{\mathbf{A}}_T^r$ with grid sizes G_r^r and G_t^r , respectively. Based on these notations, the refined DL-CS-CE summarized in Algorithm 2 follows the same implementation as that of Algorithm 1 except some technical differences during the

channel reconstruction stage, on which we focus our attention in the sequel.

Multicarrier Channel Reconstruction and Refinement: The while loop between lines 22 and 29 refines the path components by iterative projections. In line 23, the detected support $\mathcal{I}(i)$ is first transformed into column and row indices of a $G_r \times G_t$ matrix representing the indices ($i_{\text{AoA}}^d, i_{\text{AoD}}^d$) of the detected AoAs and AoDs in the original lower resolution dictionary matrices $\tilde{\mathbf{A}}_R$ and $\tilde{\mathbf{A}}_T$, respectively. In line 24, a multi-resolution fine-tuning method is applied to enhance the resolution of the detected AoAs and AoDs. The refining procedure consists of two steps as shown between lines 36 and 39 of Algorithm 2. In what follows, these steps are explained based on the column index set notation $\Omega_{K,q}$, where $K \in \{A, D\}$ represents arrival or departure, and $q \in \{d, r\}$ refer to detection or refinement, respectively.

- 1) The first step starts with line 36 which refine the angle components with the highest number of antennas. For instance, let's assume that $N_r > N_t$. By increasing the resolution of $\hat{\phi}_l$ to $G_r^r \gg G_r$, the maximum projection along the refined receiving array steering matrix $\tilde{\mathbf{A}}_R^r$, while the corresponding AoD $\hat{\theta}_l$ is fixed, can be expressed as

$$i_{\text{AoA}}^r = \arg \max_i \left[\sum_{k \in \mathcal{K}} \left| \left([\Upsilon_w^d]_{:, \Omega_{D,d}} \right)^* \mathbf{y}_w[k] \right| \right]_i, \quad (25)$$

where Υ_w^d is an $ML_r \times G_r^r G_t$ matrix such that $\Upsilon_w^d = \Phi_w (\tilde{\mathbf{A}}_T \otimes \tilde{\mathbf{A}}_R)$, and $[\Upsilon_w^d]_{:, \Omega_{D,d}}$ is an $ML_r \times G_r^r$ sub-matrix with the column index set defined as $\Omega_{D,d} = \{i_{\text{AoD}}^d : i_{\text{AoD}}^d \cdot G_r^r\}$; i_{AoD}^d corresponds to the index of the previously detected AoD before refining. Then, line 37 continues with the remaining angle by increasing the resolution of $\hat{\theta}_l$ to $G_t^r \gg G_t$. Similar to (25), the maximum projection along the refined transmit array steering matrix $\tilde{\mathbf{A}}_T^r$, while the corresponding obtained refined AoA $\hat{\phi}_l$ is fixed, can be expressed as

$$i_{\text{AoD}}^r = \arg \max_i \left[\sum_{k \in \mathcal{K}} \left| \left([\Upsilon_w^r]_{:, \Omega_{A,r}} \right)^* \mathbf{y}_w[k] \right| \right]_i, \quad (26)$$

where Υ_w^r is an $ML_r \times G_r^r G_t^r$ matrix such that $\Upsilon_w^r = \Phi_w (\tilde{\mathbf{A}}_T^r \otimes \tilde{\mathbf{A}}_R^r)$, and $[\Upsilon_w^r]_{:, \Omega_{A,r}}$ is an $ML_r \times G_t^r$ sub-matrix with the column index set defined as

$$\Omega_{A,r} = \{0 : G_r^r G_t^r - 1\} \bmod G_r^r + i_{\text{AoA}}^r. \quad (27)$$

Here, i_{AoA}^r is the index of the refined AoA obtained from (25).

- 2) The second step: In line 38, after removing the angle uncertainty caused by the detection phase, we can proceed to repeat the same step by substituting all angles with their corresponding refined angles. The maximum projection along the refined received array is given by

$$i_{\text{AoA}}^{r*} = \arg \max_i \left[\sum_{k \in \mathcal{K}} \left| \left([\Upsilon_w^r]_{:, \Omega_{D,r}} \right)^* \mathbf{y}_w[k] \right| \right]_i, \quad (28)$$

where $\Omega_{D,r} = \{i_{\text{AoD}}^r : i_{\text{AoD}}^r \cdot G_r^r\}$, and i_{AoD}^r corresponds to the index obtained from the previous step in (26).

Similarly in line 39, i_{AoD}^r is now obtained using equation (26) but by substituting i_{AoA}^r in (27) with the obtained i_{AoA}^r (the result from equation (28)). Next, line 40 transforms the row and column indices $[i_{\text{AoA}}^r, i_{\text{AoD}}^r]$ into a linear index j^* . The refining procedure lastly updates the refined support estimation set $\hat{\mathcal{T}}$ by admitting index j^* into $\hat{\mathcal{T}}$.

D. Estimation of the Sufficient Number of Paths

After estimating the channel amplitudes using the trained DnCNN, it is necessary to determine the sufficient support indices representing the number of paths needed to reconstruct the channel. An important metric to estimate the sufficient number of channel paths is the *residual* computed in our algorithms at the i^{th} iteration, which is given by

$$\mathbf{r}^{(i)}[k] = \mathbf{y}_w[k] - [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}}^r \hat{\boldsymbol{\xi}}[k], \quad \forall k, \quad (29)$$

where $\hat{\mathcal{T}}$ is obtained either from the output of the trained DnCNN ($\hat{\mathcal{T}} = \hat{\mathcal{T}} \cup \mathcal{I}(i)$ such that $\mathcal{I} = \text{INDEXSORTDESCEND}(\sum_{k \in \mathcal{K}} |\hat{\mathbf{g}}[k]|)$, as per steps 21 and 23 in the proposed DL-CS-CE Algorithm 1) or from the refining step $\hat{\mathcal{T}} = \text{REFINE}(i_{\text{AoA}}^d, i_{\text{AoD}}^d)$ in the proposed refined DL-CS-CE (as per step 24 in Algorithm 2). Hence, the residual is computed based on different metrics than that in [11].

To solve this detection problem, some prior information is needed to compare the received signals $\mathbf{y}[k]$ with the reconstructed signals $\hat{\mathbf{x}}_{\text{rec}}[k] = [\mathbf{\Upsilon}]_{:, \hat{\mathcal{T}}}^d \hat{\boldsymbol{\xi}}[k]$. For instance, the noise variance is assumed to be known at the receiver in which the receiver can accurately estimate the noise variance before the training stage takes place. Hence, the received signal $\mathbf{y}[k]$ can be approximately modeled as $\mathbf{y}[k] \approx \hat{\mathbf{x}}_{\text{rec}}[k] + \tilde{\mathbf{n}}_c[k]$, since $\hat{\mathbf{x}}_{\text{rec}}[k]$ is an estimate of the mean of $\mathbf{y}[k]$.

Moreover, using the *classical* maximum likelihood estimator, the estimation of the noise variance can be formulated using the widely known Maximum-Likelihood estimation problem [11], [42]. The log likelihood function of $\mathbf{y}(\mathbf{y}, \hat{\mathbf{x}}_{\text{rec}}, \sigma^2)$ is given by

$$\begin{aligned} \mathcal{L}(\mathbf{y}, \hat{\mathbf{x}}_{\text{rec}}, \sigma^2) &= -KML_r \ln \pi \sigma^2 - \ln \det\{\mathbf{C}_w\} \\ &\quad - \frac{1}{\sigma^2} \sum_{k=0}^{K-1} (\mathbf{y}[k] - \hat{\mathbf{x}}_{\text{rec}}[k])^* \mathbf{C}_w^{-1} (\mathbf{y}[k] - \hat{\mathbf{x}}_{\text{rec}}[k]), \end{aligned}$$

where $\mathbf{y} \triangleq \text{vec}\{\mathbf{y}[0], \dots, \mathbf{y}[K-1]\}$ represents the complete received signal, $\hat{\mathbf{x}}_{\text{rec}} \triangleq \text{vec}\{\hat{\mathbf{x}}_{\text{rec}}[0], \dots, \hat{\mathbf{x}}_{\text{rec}}[K-1]\}$ is the complete reconstructed signal. Knowing that the ML estimator of the noise variance is formulated as follows $\hat{\sigma}_{\text{ML}}^2 = \arg \max_{\sigma^2} \mathcal{L}(\mathbf{y}, \hat{\mathbf{x}}_{\text{rec}}, \sigma^2)$, a closed form expression of the ML estimate of the noise variance can be obtained by taking partial derivative with respect to σ^2 where $\partial \mathcal{L}(\mathbf{y}, \hat{\mathbf{x}}_{\text{rec}}, \sigma^2) / \partial \sigma^2 = 0$. Hence, $\hat{\sigma}_{\text{ML}}^2$ is given by

$$\begin{aligned} \hat{\sigma}_{\text{ML}}^2 &= \frac{1}{KML_r} \sum_{k=0}^{K-1} (\mathbf{y}[k] - \hat{\mathbf{x}}_{\text{rec}}[k])^* \mathbf{C}_w^{-1} (\mathbf{y}[k] - \hat{\mathbf{x}}_{\text{rec}}[k]) \\ &= \frac{1}{KML_r} \sum_{k=0}^{K-1} \mathbf{r}^*[k] \mathbf{r}[k], \end{aligned} \quad (30)$$

where the $ML_r \times 1$ vector $\mathbf{r}[k] \triangleq \mathbf{y}_w[k] - \mathbf{D}_w^{-*} \hat{\mathbf{x}}_{\text{rec}}$ matches the residual expression obtained in Algorithm 1 (Step 26) and

Algorithm 2 (Step 27) that depends on the detected supports from the trained DnCNN and the refining steps. One can note that $\mathbf{r}[k]$ can also be expressed as $\mathbf{r}[k] = (\mathbf{I}_{ML_r} - \mathbf{P}) \mathbf{y}_w[k]$, where $\mathbf{P} \in \mathbb{C}^{ML_r \times ML_r}$ represents the projection matrix given by $\mathbf{P} = [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}}^\dagger [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}}$. Therefore, the halting criterion to determine whether a sufficient number of paths has already been detected is the noise variance. The detection process is achieved when the estimated noise variance becomes equal to the true noise variance of the received signal by setting ϵ to σ^2 in (11).

Therefore, for a sufficient number of iterations, \hat{L} sufficient paths are expected to be detected as those \hat{L} paths correspond to the dominant \hat{L} entries of $\sum_{k \in \mathcal{K}} |\mathbf{h}^v[k]|$.

IV. CONVERGENCE AND COMPLEXITY ANALYSIS

In this section, we analyze the convergence of the proposed algorithms to a local optimum, which is then followed by their step-by-step computational complexity analysis.

A. Convergence Analysis

We assume that the dictionary sizes G_t and G_r are large enough⁵ to have coarsely quantized AoAs/AoDs are accurately estimated. For the sake of simplicity, we build the convergence analysis based on the notation for Algorithm 1 to analyze the convergence, which is also applicable for Algorithm 2. In order to insure convergence to a local optimum, the energy of the residual computed at the $i+1^{\text{th}}$ iteration should be strictly smaller than that of the previous i^{th} iteration, i.e.,

$$\|\mathbf{r}^{(i+1)}[k]\|_2^2 < \|\mathbf{r}^{(i)}[k]\|_2^2, \quad k = 0, \dots, K-1. \quad (31)$$

Noting that the residual computation in compressive sensing approaches (OMP and its variant for SW-OMP in [11]) and the proposed algorithms are different in the sense that they depend on the support estimated, but they follow the same criteria to guarantee convergence.

The residual for a given iteration i is expressed as $\mathbf{r}^{(i)}[k] = (\mathbf{I}_{ML_r} - \mathbf{P}^{(i)}) \mathbf{y}_w[k]$, where $\mathbf{P}^{(i)} \in \mathbb{C}^{ML_r \times ML_r}$ corresponds to a projection matrix given by $\mathbf{P}^{(i)} \triangleq [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i)}} [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i)}}^\dagger$ and $\hat{\mathcal{T}}^{(i)}$ represent the chosen detected supports at the i^{th} iteration of Algorithms 1 and 2. It is worth mentioning that the residual $\mathbf{r}^{(i)}[k]$ is the vector resulting from projecting $\mathbf{y}_w[k]$ onto the subspace orthogonal to the column space of $[\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i)}}$. Consequently, the condition in (31) is equivalent to $\|\mathbf{P}^{(i+1)} \mathbf{y}_w[k]\|_2^2 > \|\mathbf{P}^{(i)} \mathbf{y}_w[k]\|_2^2$ where $\mathbf{P}^{(i+1)} = [[\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i)}} [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i+1)*}}] \times [[\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i)}} [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i+1)*}}]^\dagger$, and $\hat{\mathcal{T}}^{(i+1)*}$ is the estimate for the support index found during the $i+1^{\text{th}}$ iteration, such that $\hat{\mathcal{T}}^{(i+1)*} \notin \hat{\mathcal{T}}^{(i)}$.

By using the formula for the inverse of a 2×2 block matrix (from Appendix 8B in [42]), the projection matrix $\mathbf{P}^{(i+1)}$ can be recursively written as a function of $\mathbf{P}^{(i)}$ as

$$\begin{aligned} \mathbf{P}^{(i+1)} &= \mathbf{P}^{(i)} + \underbrace{\frac{(\mathbf{I}_{ML_r} - \mathbf{P}^{(i)}) [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i+1)*}} [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i+1)*}}^* (\mathbf{I}_{ML_r} - \mathbf{P}^{(i)})}{[\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i+1)*}}^* (\mathbf{I}_{ML_r} - \mathbf{P}^{(i)}) [\mathbf{\Upsilon}_w]_{:, \hat{\mathcal{T}}^{(i+1)*}}}}_{\Delta \mathbf{P}^{(i+1)}}, \end{aligned} \quad (32)$$

⁵This assumption holds for large enough values of M and K [6].

TABLE III
 ONLINE COMPUTATIONAL COMPLEXITY OF ALGORITHM 1

Operation	Complexity
$K_p \times \mathbf{c}[k] = \mathbf{Y}_w^* \mathbf{r}[k]$	$\mathcal{O}(K_p(G_r G_t) M L_r)$
Estimation using DnCNN	(34)
$\max_p \sum_{k \in \mathcal{K}} \mathbf{h}^v[k] $	$\mathcal{O}(K_p(G_t G_r) \hat{L})$
$(K) \times \mathbf{x}_{\hat{\mathcal{T}}}[k] = \left([\mathbf{Y}_w]_{:, \hat{\mathcal{T}}} \right)^\dagger \mathbf{y}_w[k]$	$\mathcal{O}(2\hat{L}^2 L_r M + \hat{L}^3)$
$(K) \times \mathbf{r}[k] = \mathbf{y}_w[k] - [\mathbf{Y}_w]_{:, \hat{\mathcal{T}}} \hat{\xi}[k]$	$\mathcal{O}(K L_r M \hat{L})$
$\text{MSE} = \frac{1}{K M L_r} \sum_{k=0}^{K-1} \mathbf{r}^* [k] \mathbf{r}[k]$	$\mathcal{O}(K L_r M \hat{L})$
Overall	$\mathcal{O}(K_p(G_r G_t) M L_r)$

 TABLE IV
 ONLINE COMPUTATIONAL COMPLEXITY OF ALGORITHM 2

Operation	Complexity
$K_p \times \mathbf{c}[k] = \mathbf{Y}_w^* \mathbf{r}[k]$	$\mathcal{O}(K_p(G_r G_t) M L_r)$
Estimation using DnCNN	(34)
$\max_p \sum_{k \in \mathcal{K}} \mathbf{h}^v[k] $	$\mathcal{O}(K_p(G_t G_r) \hat{L})$
$\arg \max_i \left[\sum_{k \in \mathcal{K}} \left \left([\mathbf{Y}_w^d]_{:, \Omega} \right)^* \mathbf{y}_w[k] \right _i \right]$	$\mathcal{O}(K_p M L_r G_t^r \hat{L})$
$\arg \max_i \left[\sum_{k \in \mathcal{K}} \left \left([\mathbf{Y}_w^r]_{:, \Omega} \right)^* \mathbf{y}_w[k] \right _i \right]$	$\mathcal{O}(K_p M L_r G_t^r \hat{L})$
$(K) \times \mathbf{x}_{\hat{\mathcal{T}}}[k] = \left([\mathbf{Y}_w]_{:, \hat{\mathcal{T}}} \right)^\dagger \mathbf{y}_w[k]$	$\mathcal{O}(2\hat{L}^2 L_r M + \hat{L}^3)$
$(K) \times \mathbf{r}[k] = \mathbf{y}_w[k] - [\mathbf{Y}_w]_{:, \hat{\mathcal{T}}} \hat{\xi}[k]$	$\mathcal{O}(K L_r M \hat{L})$
$\text{MSE} = \frac{1}{K M L_r} \sum_{k=0}^{K-1} \mathbf{r}^* [k] \mathbf{r}[k]$	$\mathcal{O}(K L_r M \hat{L})$
Overall	$\mathcal{O}(K_p M L_r G_t^r \hat{L})$

with $\Delta \mathbf{P}^{(i+1)} \in \mathbb{C}^{M L_r \times M L_r}$ is another projection matrix that considers the relation between the projections at the i^{th} and $i+1^{\text{th}}$ iterations. The equation in (32) can be easily shown to fulfill the orthogonality principle, i.e., $\mathbf{P}^{(i+1)} \Delta \mathbf{P}^{(i+1)} = \mathbf{0}$. Then, $\|\mathbf{P}^{(i+1)} \mathbf{y}_w[k]\|_2^2$ can be expressed as

$$\begin{aligned} \|\mathbf{P}^{(i+1)} \mathbf{y}_w[k]\|_2^2 &= \|\mathbf{P}^{(i)} \mathbf{y}_w[k] + \Delta \mathbf{P}^{(i+1)} \mathbf{y}_w[k]\|_2^2 \\ &= \|\mathbf{P}^{(i)} \mathbf{y}_w[k]\|_2^2 + \|\Delta \mathbf{P}^{(i+1)} \mathbf{y}_w[k]\|_2^2, \end{aligned} \quad (33)$$

which satisfies the triangle equality. Moreover, $\Delta \mathbf{P}^{(i+1)}$ is *idempotent* [42] in which, using straight-forward linear algebraic manipulations, it is easy to show that $\Delta \mathbf{P}^{(i+1)} = (\Delta \mathbf{P}^{(i+1)})^2$. Hence, the eigen values of $\Delta \mathbf{P}^{(i+1)}$ are either 0 or 1, thereby, $\|\mathbf{P}^{(i+1)} \mathbf{y}_w[k]\|_2^2 > \|\mathbf{P}^{(i)} \mathbf{y}_w[k]\|_2^2$. Since the condition is satisfied, the proposed algorithms are therefore guaranteed to converge to a local optimum. Moreover, Table II shows the average number of sufficient iterations $|\hat{\mathcal{T}}| = \hat{L}$ for a range of SNR values. The results in the table confirms that the proposed support detection method using the trained DnCNN needs few iterations to converge.

B. Computational Analysis

The computational complexity for Algorithm 1 and Algorithm 2 are provided in Table III and Table IV, respectively. For comparison purposes, the overall computational complexity of SW-OMP [11] benchmark is also provided in Table V. Since some steps can be performed before running the channel estimation algorithms, we will distinguish between online and offline operations. For instance, the matrices $\mathbf{Y}_w = \mathbf{D}_w^* \mathbf{Y}$, \mathbf{C}_w , \mathbf{D}_w , \mathbf{Y}_w^d , and \mathbf{Y}_w^r can be computed offline before explicit channel estimation.

 TABLE V
 ONLINE COMPUTATIONAL COMPLEXITY OF SW-OMP [11]

Operation	Overall Complexity
For grid size dict. matrices $G_r G_t$	$\mathcal{O}(K(G_r G_t) M L_r \hat{L})$
For grid size dict. matrices $G_r^r G_t^r$	$\mathcal{O}(K(G_r^r G_t^r) M L_r \hat{L})$

Besides, the computational complexity of the proposed DnCNN arises from both online deployment and offline training. Although the online complexity is easier to compute, the offline training complexity is still an open issue due to a more involved implementation of the backpropagation process during training [45]. Therefore, we only consider the complexity of the online deployment which is based on simple matrix-vector multiplications.

For a deep neural network with L_C convolutional layers [46], the total time complexity of is given by

$$\mathcal{O} \left(\sum_{l=1}^{L_C} D_x^{(l)} D_y^{(l)} D_z^{(l)} b_x^{(l)} b_y^{(l)} c_{\text{CL}}^{(l-1)} c_{\text{CL}}^{(l)} \right) \quad (34)$$

where $D_x^{(l)}$, $D_y^{(l)}$ and $D_z^{(l)}$ are the convolutional kernel dimensions, $b_x^{(l)}$ and $b_y^{(l)}$ are the dimensions of the l^{th} convolutional layer output; and $c_{\text{CL}}^{(l)}$ is the number of filters in the l^{th} layer. We should also note that DL enjoys the advantages of graphics processing units (GPUs) and parallel processing, and hence, the overall time complexity is dominated by the analytical operations performed in the proposed algorithms.

Note that the complexity given in [11] is per iteration, where the computations will be repeated for a total of \hat{L} iterations that is why we multiply the overall complexity from [11] with \hat{L} . Hence, the overall complexity of SW-OMP is approximately $\mathcal{O}(K(G_r G_t) M L_r \hat{L})$, whereas the complexity of the proposed DL-CS-CE is $\mathcal{O}(K_p(G_r G_t) M L_r)$ such that K_p is a fraction of K subcarriers. We note that the complexity of DL-CS-CE is not multiplied by \hat{L} , since the supports are detected in one-shot from the output of the DnCNN. Consequently, the overall complexity of DL-CS-CE is lower than that of SW-OMP. Furthermore, the same applies for the refined DL-CS-CE approach of overall complexity $\mathcal{O}(K_p M L_r G_t^r \hat{L})$ in which we use higher resolution dictionary matrices where the grid sizes are much higher than the grid sizes used in DL-CS-CE ($G_r^r \gg G_r$ and $G_t^r \gg G_t$). The complexity of the refined DL-CS-CE is still lower than that of SW-OMP where the complexity becomes $\mathcal{O}(K(G_r^r G_t^r) M L_r)$ after considering the same higher resolution dictionary matrices for the sake of fair comparison.

In addition, if we increase the resolution of these dictionary matrices in Algorithm 1 (DL-CS-CE), the complexity order, which is in the order of $\mathcal{O}(K_p(G_r G_t) M L_r)$, will also increase. For example, if the resolution of dictionary matrices are increased from $G_r = 2N_r/G_t = 2N_t$ to $G_r = G_r^r = 8N_r/(G_t^r = G_t = 8N_t)$, the complexity order of Algorithm 1 (DL-CS-CE) becomes $\mathcal{O}(K_p(G_r^r G_t^r) M L_r)$. On the other hand, the complexity order of Algorithm 2 (refined DL-CS-CE) is $\mathcal{O}(K_p G_t^r M L_r)$. For the same set of parameters, complexity order of Algorithm 2 is less than Algorithm 1 by a factor of

$\frac{L}{G_r^t} = \frac{L}{8N_t}$ knowing that $8N_t \gg L$, since the number of paths (L) are known to be limited in mmWave systems (from the sparsity nature).

In Section V-E, we compare the computation times of the proposed methods with that of SW-OMP.

V. SIMULATION RESULTS

This section evaluates the performance of the proposed algorithms and compares empirical results with benchmark frequency-domain channel estimation algorithms, including SW-OMP [11]. The results are obtained through extensive Monte Carlo simulations to evaluate the average normalized mean squared error (NMSE), and the ergodic rate as a function of SNR and the number of training frames M . The simulations are performed based on realistic channel realizations from Raymobtime channel datasets.

The main parameters used for system configuration are as follows. The phase-shifters used in both the transmitter and the receiver are assumed to have N_Q quantization bits, so that the entries of the training vectors $\mathbf{f}_t^{(m)}$, $\mathbf{w}_r^{(m)}$, $m = 1, 2, \dots, M$ are drawn from the set $\mathcal{A} = \left\{0, \frac{2\pi}{2^{N_Q}}, \dots, \frac{2\pi(2^{N_Q}-1)}{2^{N_Q}}\right\}$. The number of quantization bits is set to $N_Q = 2$. The band-limiting filter $p_{rc}(t)$ is assumed to be a raised-cosine filter with roll-off factor of 0.8.

The DnCNN adopted in this work has $L_C = 3$ convolutional layers. The first convolutional layer uses $c_{CL}^1 = 64$ different $3 \times 3 \times 1$ filters. The succeeding convolutional layer uses 64 different $3 \times 3 \times 64$ filters. The final convolutional layer uses one separate $3 \times 3 \times 64$ filter. Moreover, we divide the dataset into the training set and the validation set randomly, where the size of the training set is 70% of the total set and the validation set is the other 30%. We adopt the adaptive moment estimation (Adam) optimizer to train the DnCNN. The DnCNN is trained for 10 epochs, where 256 mini-batches are utilized in each epoch. The learning rate is set to 0.01. The training process terminates when the validation accuracy does not improve in ten consecutive iterations.

Unless stated explicitly otherwise, the default system parameters used throughout the experimental simulations are summarized in Table VI, where $\mathcal{U}(\cdot, \cdot)$ represents the uniform distribution.

A. DnCNN Convergence and Network Architecture

In Fig. 4a, we plot the progression of the loss versus the number of training iterations, and show how the loss changes with different learning rates. Fig. 4a verifies the model's convergence and generality by showing that validation and training losses converge and match well. Moreover, it is worth noting that setting the learning rate to 0.01 is the best option as it guarantees the lowest loss value with a relatively high convergence speed.

In Fig. 4b, we plot the training loss progression for different network architectures, i.e., we compare the performance for different L_C depths of convolutional layers. It can be seen that using $L_C = 3$ provides the optimal performance in terms of

TABLE VI
SIMULATION PARAMETERS

Parameter	Value
Total size of dataset	10,000
Total number of subcarriers (K)	16
Subset number of subcarriers (K_p)	$K/4$
Operating frequency	60 GHz
Number of TX (RX) antennas N_t (N_r)	16(64)
Number of TX (RX) RF chains L_t (L_r)	2(4)
Grid size of TX (RX) detecting dictionary steering vectors G_t (G_r)	$2N_t(2N_r)$
Grid size of TX (RX) refining dictionary steering vectors G_t^r (G_r^r)	$8N_t(8N_r)$
Channel paths L	16
Number of delay taps of the channel N_c	16
Distribution of AoAs/AoDs	$\mathcal{U}(0, \pi)$

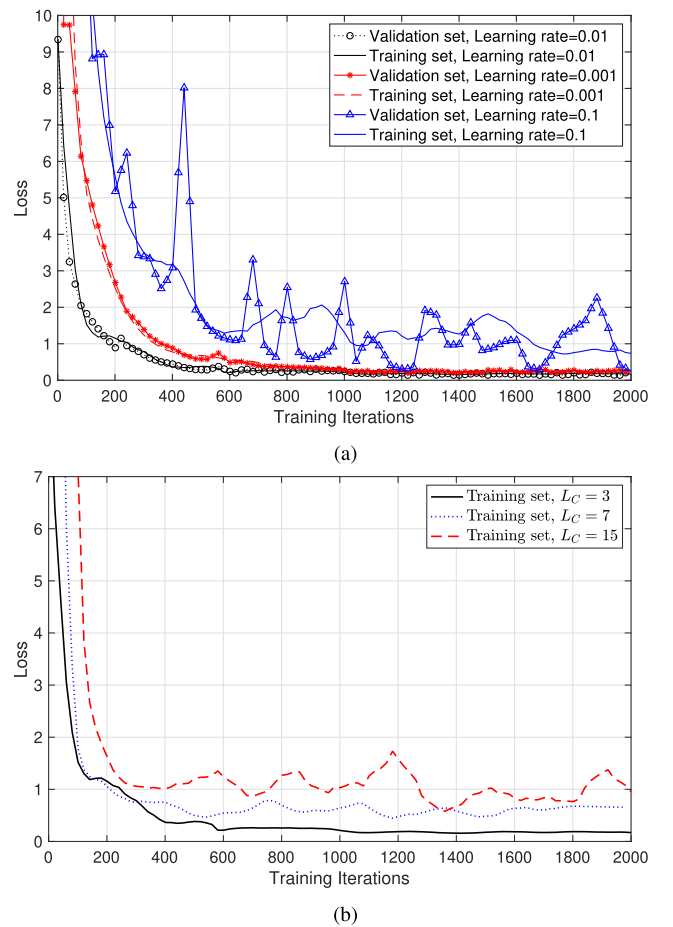


Fig. 4. (a) Training loss versus validation loss performance for a selection of Learning rates with number of layers Depth $L_C = 3$, and (b) training loss versus validation loss performance for varying number of layers $L_C = \{3, 7, 15\}$ with learning rate = 0.01.

convergence and training loss. When we increased the hidden layers, the model learned more parameters than needed to solve the problem.

B. Comparison of the Normalized Mean Squared Errors

One of the key performance metrics for the channel estimate $\hat{\mathbf{H}}[k]$ is the NMSE, which is expressed for a given

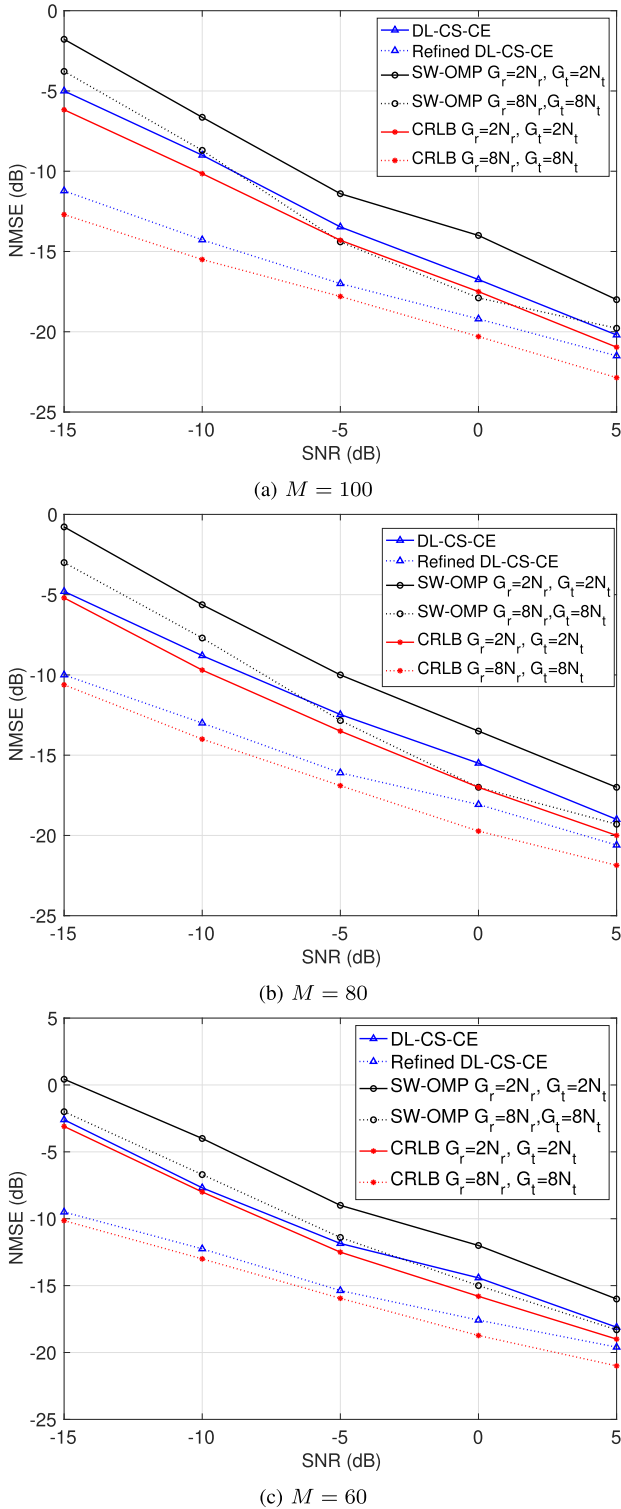


Fig. 5. The NMSE vs. SNR for the DL-CS-CE, the refined DL-CS-CE, and the SW-OMP ($N_t = 16$, $N_r = 64$, $K = 16$).

realization as

$$\text{NMSE} = \frac{\sum_{k=0}^{K-1} \|\hat{\mathbf{H}}[k] - \mathbf{H}[k]\|_F^2}{\sum_{k=0}^{K-1} \|\mathbf{H}[k]\|_F^2}. \quad (35)$$

The NMSE is considered our baseline metric to compute the proposed algorithms' performance and will be averaged over many channel realizations. While the DnCNN is used

to detect the supports of the channel matrices, the CRLB gives the lower bound when the supports are perfectly known. Therefore, it is necessary to compare the NMSE performance of the proposed approaches against CRLB to see how close is the support detection from the target supports. Accordingly, the normalized CRLB (NCRLB), from which the supports are perfectly estimated [11], is also provided as a benchmark to compare each algorithm's average performance with the lowest achievable NMSE.

We compare the average NMSE versus SNR obtained for the different channel estimation algorithms in Figs. 5 for a practical SNR range of -15 dB to 5 dB and three different lengths of training frames $M = \{100, 80, 60\}$. It is worth noting that the choice of the SNR range is based on the fact that the expected SNR typically ranges between -20 dB and 0 dB in mmWave communication systems. Using a large number of training frames M increases performance at the cost of both higher overhead and computational complexity since the complexity of estimating the support, and channel gains grows linearly with $L_r M$.

In Fig. 5, DL-CS-CE with refining performs the best, achieving NMSE values very close to the NCRLB (around 1 dB gap). The performance difference between SW-OMP and proposed algorithms is noticeable, which comes from the fact that SW-OMP estimates the mmWave channel dominant entries sequentially rather than at a single shot. The DL-CS-CE obviously deliver an NMSE lower than that of SW-OMP by -3 dB. The refined DL-CS-CE achieves even lower NMSE values below -10 dB especially for low SNR values such as $\text{SNR} = -15$ dB whereas SW-OMP with higher resolution grid sizes achieves NMSE around -3 dB and -4 dB for $\text{SNR} = -15$ dB.

In Fig. 7a, we compare the NMSE of the DL-CS-CE with $G_r = 2N_r$ and $G_t = 2N_t$ and the refined DL-CS-CE with refining grid sizes $G_r^r = \{2N_r, 4N_r, 8N_r, 16N_r\}$ and $G_t^r = \{2N_t, 4N_t, 8N_t, 16N_t\}$. It is obvious from Fig. 7a that setting the dictionary sizes to twice the number of antennas at transmitter and receiver is not enough to estimate the exact AoDs/AoAs that lie in the off grid regions of the dictionary. At this very point, the refining method introduced in Algorithm 2 is shown to greatly enhance the NMSE performance especially for the low SNR regime, at the cost of increased computational complexity as the refining resolution increases as shown in Table IV. Hence, a trade-off exists between attaining good NMSE performance and keeping the computational complexity order low. However, even with the proposed refining approach, the complexity remains lower than that of SW-OMP for the same high resolution dictionary matrices by at least two orders of magnitude. For instance, by taking $M = 100$, $K_p = K/4$, $G_t^r = 8N_t$, and $G_r^r = 8N_r$, the complexity order of SW-OMP is $\mathcal{O}(K(G_r^r G_t^r) M L_r L) = \mathcal{O}(6.7 \times 10^9)$, while the complexity order of the refined DL-CS-CE is $\mathcal{O}(K_p G_r^r M L_r L) = \mathcal{O}(1.3 \times 10^7)$.

In Fig. 6, we compare the proposed channel estimation algorithms with SIMGW-OMP from [13] for $M = 80$, $K = 64$, $K_p = 16$, $L_r = 4$, $L_t = 4$, $N_t = 32$, $N_r = 32$. The proposed refined DL-CS-CE approach provides an NMSE performance superior to SIMGW-OMP, especially at the low

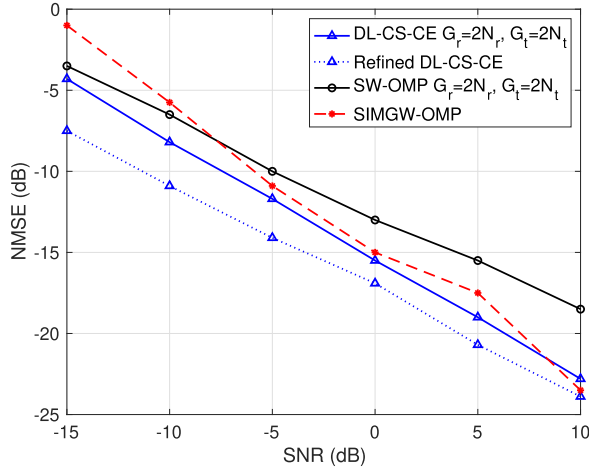


Fig. 6. NMSE performance for $M = 80$, $K = 64$, $K_p = 16$, $L_r = 4$, $L_t = 4$, $N_t = 32$, $N_r = 32$.

SNR regime ($\text{SNR} < 0$ dB). Therefore, the refined DL-CS-CE is especially more suitable for mmWave systems, where the expected SNR typically ranges between -20 dB and 0 dB.

Moreover, Fig. 7a shows the NMSE performance of Algorithm 1 and 2 under different refining resolutions $G_r^r = \{2N_r, 4N_r, 8N_r, 16N_r\}$, $G_t^r = \{2N_t, 4N_t, 8N_t, 16N_t\}$. It is worth noting that Algorithm 2 is an extension of Algorithm 1 with an extra refining applied for more accurate AoA/AoD estimation. It is shown in Fig. 7a that as the refining resolution increases (i.e., $G_r^r > 2N_r$, $G_t^r > 2N_t$), the NMSE performance is enhanced. Therefore, Algorithm 1 (DL-CS-CE) provides a coarse estimate of the AoAs/AoDs by just relying on the output of the DnCNN and low resolution dictionary matrices. On the other hand, Algorithm 2 (refined DL-CS-CE) presents a refined performance that accounts for the continuous nature of the AoAs and AoDs to provide better NMSE performance, especially in the low SNR regime (i.e., $\text{SNR} < 0$ dB).

C. Comparisons for the Probability of Successful Support Estimation for L Paths

In Fig. 7b, we compare the successful support detection probability versus SNR for the proposed DnCNN-based amplitude estimation and that of SW-OMP. It can be seen that the proposed DnCNN outperforms SW-OMP over the whole SNR range as the trained DnCNN can efficiently denoise the correlated input image and obtain a sparse matrix of the channel amplitudes. From this denoised sparse matrix, the indices of the supports (i.e., dominant entries of the obtained sparse matrix) are detected. Moreover, we show that when we set $K_p \ll K$, the support detection is not affected, since as shown in Section II-B1 $\Delta[k]$ have the same support for all k . Therefore, we can reduce computational complexity since there is no need to compute the correlation step (given in (15)) for all subcarriers. Thus, a smaller subset of subcarriers can also provide a high probability of correct support detection.

D. Spectral Efficiency Comparison

Another key performance metric is the spectral efficiency, which is computed by assuming fully-digital precoding

TABLE VII
AVERAGE RUNNING TIME FOR $M = 100$ AND $\text{SNR} = -5$ dB

Algorithm	Run time [seconds]
DL-CS-CE $G_r = 2N_r$ and $G_t = 2N_t$	0.144
Refined DL-CS-CE $G_r^r = 2N_r$ and $G_t^r = 2N_t$	0.201
Refined DL-CS-CE $G_r^r = 8N_r$ and $G_t^r = 8N_t$	0.464
SW-OMP for grids $G_r = 2N_r$ and $G_t = 2N_t$	0.25
SW-OMP for grids $G_r = 8N_r$ and $G_t = 8N_t$	0.97

and combining. In this way, using estimates for the N_s dominant left and right singular vectors of the channel estimate gives K parallel effective channels $\mathbf{H}_{\text{eff}}[k] = \begin{bmatrix} \hat{\mathbf{U}}[k] \\ \vdots, 1:N_s \end{bmatrix}^* \mathbf{H}[k] \begin{bmatrix} \hat{\mathbf{V}}[k] \\ \vdots, 1:N_s \end{bmatrix}$. Accordingly, the average spectral efficiency can be expressed as

$$R = \frac{1}{K} \sum_{k=0}^{K-1} \sum_{n=1}^{N_s} \log_2 \left(1 + \frac{\text{SNR}}{N_s} \lambda_n(\mathbf{H}_{\text{eff}}[k])^2 \right), \quad (36)$$

with $\lambda_n(\mathbf{H}_{\text{eff}}[k])$, $n = 1, \dots, N_s$ the eigenvalues of each effective channel $\mathbf{H}_{\text{eff}}[k]$.

In Fig. 7c, we show the achievable spectral efficiency as a function of the SNR for the different channel estimation algorithms. The proposed DL-CS-CE approach provides at least 3.6% performance improvement over the SW-OMP. The refined DL-CS-CE provides near-optimal achievable rates with at least 12.6% performance improvement over the other schemes. The spectral efficiency gap of the different schemes is smaller than that of the NMSE gap, since the NMSE performance is much more sensitive to the success rate of the sparse recovery. However, the spectral efficiency performance is determined by the beamforming gain and is less sensitive to the success rate of the sparse recovery.

In Fig. 7d, we show the achievable spectral efficiency as a function of different training lengths for the proposed schemes under different SNRs. We observe that using $M > 40$ frames does not significantly improve performance, which leverages the robustness of the two proposed approaches. Simulations also show that near-optimal achievable rates can be achieved by using a reasonable number of frames, i.e., $60 \leq M \leq 100$. Therefore, with the proposed schemes, we can save in training overhead.

E. Time Complexity Analysis

Table VII shows online estimation stage computational times of the proposed frameworks and SW-OMP [11]. SW-OMP is the slowest to solve the inherent optimization problem, especially for high-resolution dictionary matrices. The running time of the DL-CS-CE without refining exhibits shorter computational times than the SW-OMP algorithm. However, for fair comparison when refining is applied, we compare the running time of the refined DL-CS-CE with higher resolution SW-OMP where $G_r = G_r^r = 8N_r$ and $G_t = G_t^r = 8N_t$, and it is shown that the refined DL-CS-CE takes less time to perform the channel estimation. Hence, we conclude that the proposed DL-CS-CE frameworks are computationally efficient and tolerant, especially for higher resolution dictionary matrices.

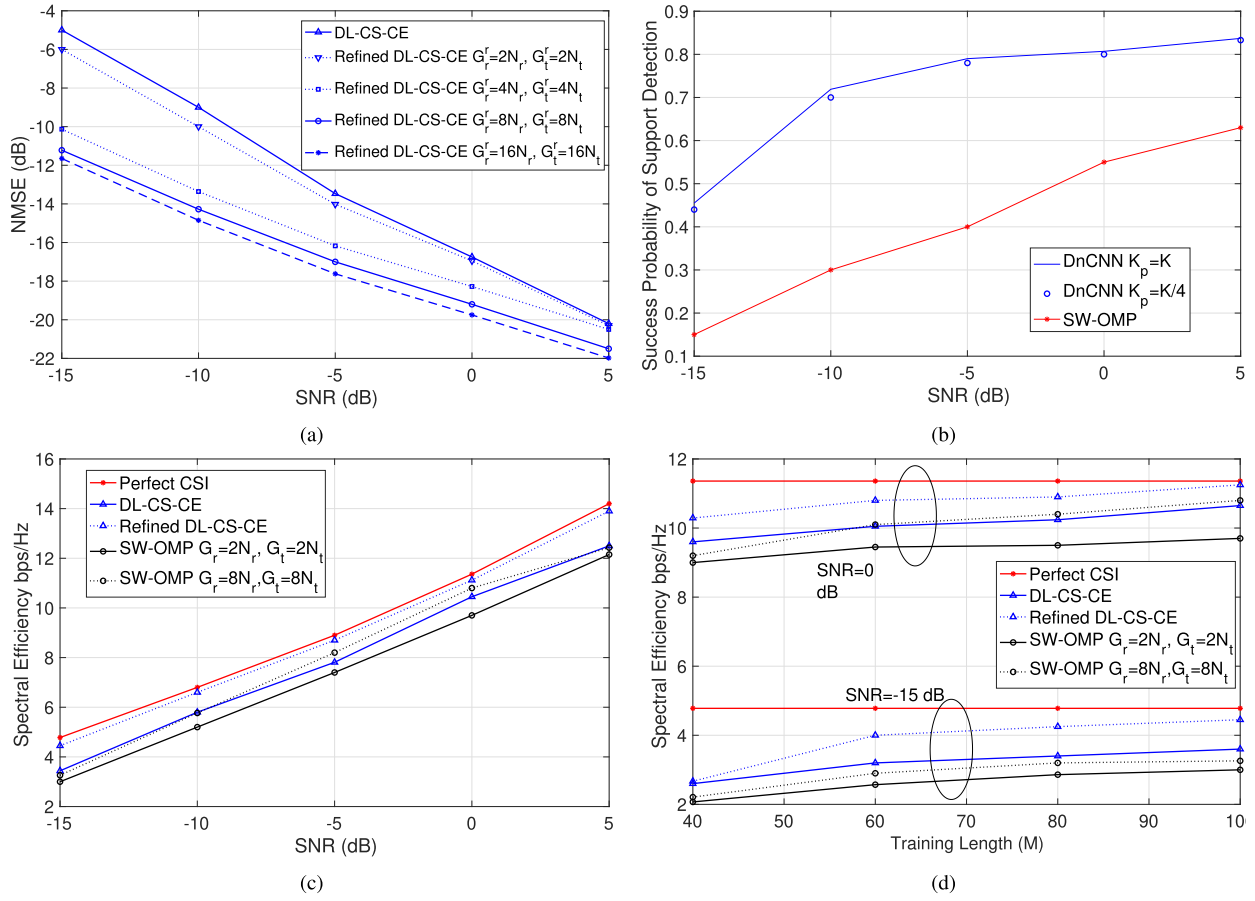


Fig. 7. (a) The NMSE vs. SNR for the DL-CS-CE and the refined DL-CS-CE under different refining grid sizes of G_r^T and G_t^T ($M = 100$), (b) probability of successfully detecting the supports vs. SNR for the DL-CS-CE, the refined DL-CS-CE, and the SW-OMP ($M = 100$), (c) spectral efficiency vs. SNR ($M = 100$), and (d) spectral efficiency vs. M training lengths (SNR = $\{-15, 0$ dB}).

VI. CONCLUSION

In this work, two DL-CS-based frequency-selective channel estimation approaches for mmWave wideband communication systems under hybrid architectures have been proposed. The developed algorithms are based on joint-sparse recovery to exploit information on the common basis shared for every subcarrier. Compared to the state-of-the-art channel estimation techniques that estimate supports iteratively, the proposed solutions reduce computational complexity and estimation error by detecting all supports simultaneously. Simulation results have shown that the DL-CS-CE and the refined DL-CS-CE schemes have better channel estimation performance than existing schemes using a reasonably small training length and low complexity order. It has also been shown that a small number of subcarriers are sufficient for successful support detection during the deep learning prediction phase. Thus, the proposed schemes are able to attain good NMSE performance with low computational complexity. For future work, we intend to extend the proposed approaches and use DnCNN for support estimation for cascaded channels in reconfigurable intelligent surfaces aided mmWave MIMO systems.

REFERENCES

- [1] Z. Pi and F. Khan, "An introduction to millimeter-wave mobile broadband systems," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 101–107, Jun. 2011.
- [2] R. W. Heath, Jr., N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.
- [3] T. Bai, A. Alkhateeb, and R. W. Heath, Jr., "Coverage and capacity of millimeter-wave cellular networks," *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 70–77, Sep. 2014.
- [4] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [5] A. Alkhateeb, J. Mo, N. Gonzalez-Prelcic, and R. W. Heath, Jr., "MIMO precoding and combining solutions for millimeter-wave systems," *IEEE Commun. Mag.*, vol. 52, no. 12, pp. 122–131, Dec. 2014.
- [6] R. Méndez-Rial, C. Rusu, N. González-Prelcic, A. Alkhateeb, and R. W. Heath, Jr., "Hybrid MIMO architectures for millimeter wave communications: Phase shifters or switches?" *IEEE Access*, vol. 4, pp. 247–267, 2016.
- [7] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath, Jr., "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 831–846, Oct. 2014.
- [8] M. F. Duarte and Y. C. Eldar, "Structured compressed sensing: From theory to applications," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4053–4085, Sep. 2011.
- [9] Z. Gao, C. Hu, L. Dai, and Z. Wang, "Channel estimation for millimeter-wave massive MIMO with hybrid precoding over frequency-selective fading channels," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1259–1262, Jun. 2016.
- [10] K. Venugopal, A. Alkhateeb, R. W. Heath, Jr., and N. G. Prelcic, "Time-domain channel estimation for wideband millimeter wave systems with hybrid architecture," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 6493–6497.

- [11] J. Rodríguez-Fernández, N. González-Prelcic, K. Venugopal, and R. W. Heath, Jr., "Frequency-domain compressive channel estimation for frequency-selective hybrid millimeter wave MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 2946–2960, May 2018.
- [12] W. Ma and C. Qi, "Beamspace channel estimation for millimeter wave massive MIMO system with hybrid precoding and combining," *IEEE Trans. Signal Process.*, vol. 66, no. 18, pp. 4839–4853, Sep. 2018.
- [13] J. Rodríguez-Fernández, N. González-Prelcic, and R. W. Heath, Jr., "A compressive sensing-maximum likelihood approach for off-grid wideband channel estimation at mmWave," in *Proc. IEEE 7th Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Dec. 2017, pp. 1–5.
- [14] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
- [15] P. Dong, H. Zhang, G. Y. Li, I. Gaspar, and N. NaderiAlizadeh, "Deep CNN-based channel estimation for mmWave massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019.
- [16] T. Zeng, J. Li, M. Hu, S. Hou, and Q. Zhang, "Toward higher performance for channel estimation with complex DnCNN," *IEEE Commun. Lett.*, vol. 24, no. 1, pp. 198–201, Jan. 2020.
- [17] N. K. Kundu and M. R. McKay, "A deep learning-based channel estimation approach for MISO communications with large intelligent surfaces," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Aug. 2020, pp. 1–6.
- [18] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.
- [19] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.
- [20] M. Wenyang, Q. Chenhao, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2838–2849, Feb. 2020.
- [21] X. Wei, C. Hu, and L. Dai, "Deep learning for beamspace channel estimation in millimeter-wave massive MIMO systems," 2019, *arXiv:1910.12455*.
- [22] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1228–1231, Aug. 2019.
- [23] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10325–10329, Oct. 2019.
- [24] E. Balevi, A. Doshi, and J. G. Andrews, "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, Jan. 2020.
- [25] Ö. T. Demir and E. Björnson, "Channel estimation in massive MIMO under hardware non-linearities: Bayesian methods versus deep learning," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 109–124, 2020.
- [26] Y. Long, Z. Chen, J. Fang, and C. Tellambura, "Data-driven-based analog beam selection for hybrid beamforming under mm-Wave channels," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 2, pp. 340–352, May 2018.
- [27] J. A. Hodge, K. V. Mishra, and A. I. Zaghoul, "Multi-discriminator distributed generative model for multi-layer RF metasurface discovery," in *Proc. IEEE Global Conf. Signal Inform. Process. (GlobalSIP)*, Ottawa, ON, Canada, Nov. 2019, pp. 1–5.
- [28] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, 2018.
- [29] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.
- [30] A. M. Elbir, "CNN-based precoder and combiner design in mmWave MIMO systems," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1240–1243, Jul. 2019.
- [31] A. M. Elbir and K. V. Mishra, "Joint antenna selection and hybrid beamformer design using unquantized and quantized deep learning networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 1677–1688, Mar. 2020.
- [32] A. M. Elbir, K. V. Mishra, M. R. B. Shankar, and B. Ottersten, "A family of deep learning architectures for channel estimation and hybrid beamforming in multi-carrier mm-Wave massive MIMO," 2019, *arXiv:1912.10036*.
- [33] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.
- [34] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [36] J. Xu, P. Zhu, J. Li, and X. You, "Deep learning-based pilot design for multi-user distributed massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1016–1019, Aug. 2019.
- [37] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.
- [38] A. Abdallah and M. M. Mansour, "Efficient angle-domain processing for FDD-based cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2188–2203, Jan. 2020.
- [39] A. Abdallah and M. M. Mansour, "Angle-based multipath estimation and beamforming for FDD cell-free massive MIMO," in *Proc. IEEE Int. Signal Process. Adv. Wireless Commun. Workshop (SPAWC)*, Cannes, France, Jul. 2019, pp. 1–5.
- [40] A. Alkhateeb and R. W. Heath, Jr., "Frequency selective hybrid precoding for limited feedback millimeter wave systems," *IEEE Trans. Commun.*, vol. 64, no. 5, pp. 1801–1818, May 2016.
- [41] E. Björnson, L. Van der Perre, S. Buzzi, and E. G. Larsson, "Massive MIMO in sub-6 GHz and mmWave: Physical, practical, and use-case differences," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 100–108, Apr. 2019.
- [42] S. M. Kay, *Fundamentals of Statistical Signal Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
- [43] R. Niazadeh, M. Babaie-Zadeh, and C. Jutten, "On the achievability of Cramér-Rao bound in noisy compressed sensing," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 518–526, Jan. 2012.
- [44] Z. Qin, J. Fan, Y. Liu, Y. Gao, and G. Y. Li, "Sparse representation for wireless communications: A compressive sensing approach," *IEEE Signal Process. Mag.*, vol. 35, no. 3, pp. 40–58, May 2018.
- [45] B. Matthiesen, A. Zappone, K.-L. Besser, E. A. Jorswieck, and M. Debbah, "A globally optimal energy-efficient power control framework and its efficient implementation in wireless interference networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 3887–3902, 2020.
- [46] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.



Asmaa Abdallah (Member, IEEE) received the B.S. (Hons.) and M.S. degrees in computer and communications engineering from Rafik Hariri University (RHU), Lebanon, in 2013 and 2015, respectively, and the Ph.D. degree in electrical and computer engineering from the American University of Beirut (AUB), Beirut, Lebanon, in 2020.

She has been a Research and Teaching Assistant at AUB since 2015. From July 2019 to December 2019, she was a Research Intern at Nokia Bell Labs, France, where she worked on new hybrid automatic

request (HARQ) mechanisms for long-delay channel in non-terrestrial networks (NTN). She is currently a Post-Doctoral Fellow at the King Abdullah University of Science and Technology (KAUST). Her research interests include communication theory, stochastic geometry for wireless communications, array signal processing, with emphasis on energy and spectral efficient algorithms for device-to-device (D2D) communications, massive multiple-input and multiple-output (MIMO) systems, and cell free massive MIMO systems. She is a member of the Executive Committee of IEEE Young Professionals Lebanon's Section. She was a recipient of the Academic Excellence Award at RHU in 2013 for ranking first on the graduating class. She also received a Scholarship from the Lebanese National Council for Scientific Research (CNRS-L/AUB) to support her Ph.D. studies.



interests include wireless communication systems and networks.

Abdulkadir Celik (Senior Member, IEEE) received the M.S. degree in electrical engineering, the M.S. degree in computer engineering, and the Ph.D. degree in co-majors of electrical engineering and computer engineering from Iowa State University, Ames, IA, USA, in 2013, 2015, and 2016, respectively. From 2016 to 2020, he was a Post-Doctoral Fellow at the King Abdullah University of Science and Technology (KAUST). He is currently a Research Scientist at the Communications and Computing Systems Laboratory, KAUST. His research



he worked on baseband receiver architectures for the IEEE 802.11ax standard. He was a Visiting Researcher at Broadcom, Sunnyvale, CA, from 2012 to 2014, where he worked on the physical layer SoC architecture and algorithm development for LTE-Advanced baseband receivers. He was on research leave with Qualcomm Flarion Technologies in Bridgewater, NJ, USA, from 2006 to 2008, where he worked on modem design and implementation for 3GPP-LTE, 3GPP2-UMB, and peer-to-peer wireless networking physical layer SoC architecture and algorithm development. He was a Research Assistant at the Coordinated Science Laboratory (CSL), UIUC, from 1998 to 2003. He worked at National Semiconductor Corporation, San Francisco, CA, with the Wireless Research Group, in 2000. He was a Research Assistant with the Department of Electrical and Computer Engineering, AUB, in 1997, and a Teaching Assistant in 1996. He joined the Department of Electrical and Computer Engineering, AUB, as a Faculty Member, in 2003, where he is currently a Professor. He has seven issued U.S. patents. His research interests include energy-efficient and high-performance VLSI circuits, architectures, algorithms, and systems for computing, communications, and signal processing.

Mohammad M. Mansour (Senior Member, IEEE) received the B.E. (Hons.) and M.E. degrees in computer and communications engineering from the American University of Beirut (AUB), Beirut, Lebanon, in 1996 and 1998, respectively, and the M.S. degree in mathematics and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2002 and 2003, respectively.

He was a Visiting Researcher at Qualcomm, San Jose, CA, USA, in summer 2016, where

Prof. Mansour is a member of the Design and Implementation of Signal Processing Systems (DISPS) Technical Committee Advisory Board of the IEEE Signal Processing Society. He served as a member for the DISPS Technical Committee from 2006 to 2013. He was a recipient of the PHI Kappa PHI Honor Society Award twice in 2000 and 2001, and the Hewlett Foundation Fellowship Award in 2006. He served as the Technical Co-Chair of the IEEE Workshop on Signal Processing Systems in 2011, and as a member of the technical program committee of various international conferences and workshops. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: EXPRESS BRIEFS (TCAS-II) from 2008 to 2013, as an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS from 2012 to 2016, and as an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 2011 to 2016.



Computer Science, University of California at Irvine, where he founded the Wireless Systems and Circuits Laboratory. His research interests include low power digital circuit and signal processing architectures with an emphasis on mobile systems. He has been on the technical program committees and steering committees for numerous workshops, symposia, and conferences in the areas of low power computing and wireless communication system design. He received several awards, as well as distinguished grants, including the NSF CAREER Grant supporting his research in low power systems.

Ahmed M. Eltawil (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees (Hons.) from Cairo University, Giza, Egypt, in 1997 and 1999, respectively, and the Ph.D. degree from the University of California at Los Angeles, Los Angeles, CA, USA, in 2003. Since 2019, he has been a Professor with the Computer, Electrical and Mathematical Science and Engineering Division (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. Since 2005, he has been with the Department of Electrical Engineering and