Deep Learning-Based Channel Estimation for Wideband Hybrid MmWave Massive MIMO

Jiabao Gao, Graduate Student Member, IEEE, Caijun Zhong, Senior Member, IEEE, Geoffrey Ye Li, Fellow, IEEE, Joseph B. Soriaga, and Arash Behboodi, Member, IEEE

Abstract—Hybrid analog-digital (HAD) architecture is widely adopted in practical millimeter wave (mmWave) massive multiple-input multiple-output (MIMO) systems to reduce hardware cost and energy consumption. However, channel estimation in the context of HAD is challenging due to only limited radio frequency (RF) chains at transceivers. Although various compressed sensing (CS) algorithms have been developed to solve this problem by exploiting inherent channel sparsity and sparsity structures, practical effects, such as power leakage and beam squint, can still make the real channel features deviate from the assumed models and result in performance degradation. Besides, the high complexity of CS algorithms caused by a large number of iterations hinders their applications in practice. To tackle these issues, we develop a deep learning (DL)-based channel estimation approach where the sparse Bayesian learning (SBL) algorithm is unfolded into a deep neural network (DNN). In each SBL layer, Gaussian variance parameters of the sparse angular domain channel are updated by a tailored DNN, which is able to capture complicated channel sparsity structures in various domains effectively and efficiently. The measurement matrix is jointly optimized for performance improvement. Then, the proposed approach is extended to the multi-block case where channel correlation in time is further exploited to adaptively predict the measurement matrix and facilitate the update of variance parameters. Simulation results show that the proposed approaches outperform existing approaches in terms of both performance and complexity.

Index Terms—Millimeter wave, massive MIMO, hybrid analog-digital, wideband channel estimation, measurement matrix, power leakage, beam squint, sparse Bayesian learning, model-driven deep learning.

I. INTRODUCTION

FUTURE wireless communication systems require extremely high transmission rate to realize various newly emerging applications, such as high quality three-dimensional (3D) video, virtual reality, and augmented reality. By deploying a large number of antennas at transceivers, massive multiple-input multiple-output (MIMO) is able to dramatically improve the spectral efficiency thanks to the extra degrees of freedom in the spatial domain [1]. On the other hand, millimeter wave (mmWave) communication is envisioned as another key enabling technology for higher throughput with huge available bandwidth [2]. Thanks to the naturally complementary features of these two technologies, mmWave band is ideally suited for the deployment of massive MIMO. Specifically, the short wavelength of mmWave enables the integration of massive antenna units in a limited space while its severe path loss and signal blockage can be compensated by the high beamforming gain of massive MIMO. Nevertheless, mmWave massive MIMO faces the practical challenge of prohibitive manufacturing cost and energy consumption of a large number of radio frequency (RF) chains. To alleviate this issue, the hybrid analog-digital (HAD) architecture has been proposed where antennas are connected to much fewer RF chains through phase shifters in the analog domain and the performance is further enhanced by digital processing [3].

It is well known that various gains of massive MIMO heavily rely on accurate channel state information (CSI). In fully-digital systems, linear estimators like least square (LS) can be used to acquire the CSI. However, channel estimation is much harder with HAD since the high-dimensional channel needs to be recovered from the low-dimensional compressed pilot signal. To achieve the same performance as in fully-digital systems, the estimation overhead of linear estimators will increase dramatically due to limited RF chains. Therefore, the development of channel estimation algorithms with both high accuracy and low overhead is critical to advance the deployment of HAD massive MIMO. Next, we will present a comprehensive review of prior works in this context.

A. Prior Works

The key idea of most existing methods is to leverage the inherent sparsity and sparsity structures of the channel. Due to limited scattering caused by the highly directional propagation behavior of mmWave, only few channel paths can reach the receiver. Therefore, the number of effective channel parameters in the angular domain is much smaller
than that of antenna pairs of transceivers [4]. To reduce the overhead of LS, in [5], the slowly changing path angles are obtained first, then only path gains need to be re-estimated for a long period. Nevertheless, the initial preamble stage is still resource-consuming and path angles may suddenly change in practice.

The compressive sensing (CS) technique is commonly used to exploit the channel sparsity in prior works, where channel estimation is formulated as a sparse recovery problem and effective channel parameters are estimated instead of the original channel [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. In [6], the orthogonal matching pursuit (OMP) algorithm is used to estimate the sparse angular domain channel. Later on, an improved version of SOMP is proposed in [7] by further exploiting the common sparsity structure among different subcarriers’ channels. In [8] and [9], the sparse Bayesian learning (SBL) algorithm is proposed for channel estimation and several variants are developed to exploit channel correlations and sparsity structures in time and frequency domains. The parameterized Gaussian priors of SBL can also be tailored to capture the block sparsity structure in the clustered mmWave channel [10] and the one-ring low-frequency channel [11]. Similar idea is adopted in [12] where approximate message passing (AMP) with neighbor pattern learning is proposed to exploit the clustered sparsity structure in the AoA-AoD-delay domain.

In spite of reduced overhead, the performance of CS algorithms heavily relies on the sparsity assumptions and will be severely degraded when real channel features deviate from the assumed models. For instance, the power leakage effect caused by grid mismatch will decrease the sparsity level of channel [5] while the beam squint effect in wideband mmWave massive MIMO systems will damage the common sparsity among different subchannels [13]. To mitigate these effects, grid refinement is performed on initial estimated grids with a higher resolution dictionary in [14]. In [15], an off-grid SBL algorithm is proposed to jointly estimate grid mismatch parameters with channel parameters. Two ESPRIT-based algorithms [16] are developed for high-resolution channel estimation with closed-form solutions for path angles. In [13], the majorization-minimization iterative approach is used to estimate squint-independent time domain channel parameters, while a special support detection window is designed to capture the beam squint pattern in [17].

Recently, deep learning (DL) has attracted widespread attention in wireless communication due to its universal approximation ability and low testing complexity. Up to now, DL has been successfully applied to many physical layer problems [18], such as CSI feedback [19], beamforming [20], signal detection [21], end-to-end transceiver design [22], and channel estimation [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]. In general, DL-based channel estimation methods can be divided into two categories, namely data-driven ones and model-driven ones. Data-driven methods directly approximate the mapping function from received signal or estimation to channel or effective channel parameters by a black-box deep neural network (DNN). In [23], a convolutional neural network (CNN) has been proposed to refine the coarse LS estimation in HAD massive MIMO systems where channel correlations in the frequency and time domains are exploited for performance improvement and overhead reduction. The initial coarse estimation can also be obtained by a conventional CS algorithm [24] or a fully-connected network (FNN) [25]. In [26], the self-attention mechanism is embedded into the FNN to realize divide-and-conquer by exploiting the separability of channel distribution, which is further extended to the more practical scenario where the environment is dynamically changing in [27] by using the cross attention technique to achieve few-shot learning. In [28], an FNN is trained to predict the amplitudes of beamspace channel entries, which is extended to the multi-carrier case in [14] using CNN.

Different from data-driven methods, model-driven methods combine the domain knowledge in communication with the power of DL by inserting trainable parameters into conventional algorithms, therefore usually having better interpretability and robustness. In [29], the AMP algorithm is unfolded with a deep CNN-based channel denoiser for beamspace mmWave massive MIMO channel estimation, which is extended to the system with beam squint in [30]. To fully exploit channel sparsity, a learnable iterative shrinkage thresholding algorithm (LISTA) is proposed in [31] where channel denoising is further performed in the sparse transform domain. In [32], deep unfolding is conducted on the multiple-measurement-vectors (MMV) version of AMP where the common sparsity among subchannels is embedded. To reduce the complexity of the high-performance off-grid SBL algorithm, deep reinforcement learning is applied to update the inserted learnable parameters in [33].

Another important part of sparse recovery is the measurement matrix. Conventional matrices are designed based on restricted isometry property (RIP), low mutual coherence, or equal received signal power at all angles [4], [11], [16]. In [34], the matrix elements are treated as trainable network parameters and the learned data-driven measurement matrix outperforms various conventional measurement matrices when applied to the existing CS algorithms while the measurement matrix and the deep learning based channel estimator are jointly trained in [25], [32], and [35].

B. Contributions

Most of current DL-based channel estimation works have not fully exploited channel sparsity structures, and the jointly trained long-term fixed measurement matrix cannot flexibly match time-varying channel distributions. To overcome the shortcomings of existing approaches, in this paper, we propose a novel joint channel estimation and measurement matrix optimization approach for wideband HAD mmWave massive MIMO systems. The main contributions are as follows:

- A model-driven channel estimation approach is proposed first in the single-block case. The SBL algorithm with multiple iterations is deeply unfolded into cascaded SBL layers, where the update of Gaussian variance parameters in each SBL layer is realized by a tailored DNN.
- The sparsity structures of wideband mmWave channel with practical effects including power leakage and beam
squint are analyzed. Then, the architecture of the DNN for variance parameter update in each SBL layer is carefully designed to capture channel sparsity structures in various domains effectively and efficiently.

- The proposed approach is extended to the multi-block case by further exploiting channel correlation in the time domain. Based on previous estimation results, the measurement matrix matching the time-varying short-term channel sparsity pattern is adaptively predicted and the update of variance parameters is facilitated.

C. Organization and Notations

The rest of this paper is organized as follows. In Section II, the HAD massive MIMO system, the wideband mmWave channel model, and the uplink channel estimation problem are introduced. In Section III, conventional SBL-based channel estimation approaches are presented. Section IV elaborates the proposed DL-based channel estimation approach, which is further extended to the multi-block case in Section V. Simulation results are provided in Section VI to validate the superiority of the proposed approaches, and the paper is concluded in Section VII.

Notations: Italic, bold-face lower-case and bold-face upper-case letters denote scalar, vector, and matrix, respectively. $\| \cdot \|$, $(\cdot)^*$, $(\cdot)^T$, $(\cdot)^{-1}$, $\cdot^\dagger$, $\{ \cdot \}$, and $\otimes$ denote l-2 norm, conjugate, transpose, conjugate transpose, inverse, modulus, expectation, and Kronecker product, respectively. $I_a$ denotes the $(a, a)$-dimensional identity matrix. $[\cdot]_{i,j}$ denotes the element at the $i$-th row and the $j$-th column. vec($\cdot$) denotes the vectorization of a matrix by stacking its columns while vec$^{-1}$($\cdot$) denotes the inverse operation. diag($\cdot$) denotes both the diagonal matrix and diagonalization operation. Bilkl.diag($\cdot$) denotes the block diagonal matrix. $\mathbb{C}^{x \times y}$ and $\mathbb{R}^{x \times y}$ denote the $x \times y$ complex and real spaces, respectively. $\mathcal{CN}(\mu, \sigma^2)$ denotes a circularly symmetric complex Gaussian (CSCG) random variable with mean $\mu$ and variance $\sigma^2$ while $\mathcal{CN}(\mu, \Sigma)$ denotes a CSCG random vector with mean $\mu$ and covariance $\Sigma$. $\mathcal{U}[a, b]$ denotes the uniform distribution between $a$ and $b$.

II. System Model and Problem Formulation

In this section, we first introduce the system model and channel model. Then, the sparsity structures of channel are analyzed. At last, the channel estimation problem is formulated.

A. System Model

Consider the HAD massive MIMO system illustrated in Fig. 1, where a base station (BS) with an $N_R$-antenna uniform linear array (ULA) and $N_{RF}^R$ RF chains serves a user with an $N_T$-antenna ULA and $N_{RF}^T$ RF chains. Time division duplex (TDD) uplink and downlink transmissions are adopted so that only the uplink channel needs to be estimated thanks to channel reciprocity [4].

Following the common practice, in each channel use, the user only activates a single RF chain to transmit the pilot signal on one beam while the BS combines the received pilot signal using all RF chains associated with different beams [23]. Denote the total number of transmit beams and receive beams as $M_T$ and $M_R$, respectively, and assume $M_R$ is an integer multiple of $N_{RF}^R$: the BS needs $M_T$ channel uses to traverse all receive beams given a fixed transmit beam. Then, the user changes the transmit beam every $\frac{M_T}{M_R}$ channel uses for $M_T$ times.

To combat the frequency selectivity of channel, the wideband OFDM system is considered, where the beams implemented by phase shifters are shared by all subcarriers and a channel use is equivalent to an OFDM symbol. In each block, the first $\frac{M_T}{M_R}N_{RF}^R$ symbols are pilots for channel estimation while the rest symbols are for data transmission. In the frequency domain, $K$ subcarriers are uniformly selected from all subcarriers to carry pilots and the channels of the rest subcarriers carrying data symbols can be obtained by methods like interpolation.

Denote the transmit beamforming matrix and the receive combining matrix as $F = [f_1, \ldots, f_{M_T}] \in \mathbb{C}^{N_T \times M_T}$ and $W = [w_1, \ldots, w_{M_R}] \in \mathbb{C}^{N_R \times M_R}$, respectively. When the user adopts the $p$-th ($1 \leq p \leq M_T$) transmit beam $f_p \in \mathbb{C}^{N_T \times 1}$ and the BS adopts the $q$-th ($1 \leq q \leq M_R$) group of receive beams $W_q = [w_{(q-1)N_{RF}^R+1}, \ldots, w_{qN_{RF}^R}] \in \mathbb{C}^{N_R \times N_{RF}^R}$, the received signal at the BS of the $k$-th subcarrier can be expressed as

$$y_{p,q}^k = W_q^H (Hk f_p + n_{p,q}^k) \in \mathbb{C}^{N_{RF}^R \times 1},$$

where $H_k \in \mathbb{C}^{N_R \times N_T}$, $s_{k}^p \sim \mathcal{CN}(0, \sigma^2 I_{N_p})$ denote the channel matrix, the transmitted pilot signal, and the noise vector before combining of the $k$-th subcarrier, respectively. Without loss of generality, $s_{k}^p = 1$ is assumed for all $p, q, k$ since it is known at the BS and can be readily eliminated while the signal-to-noise-ratio (SNR), defined as $1/\sigma^2$, can be adjusted by changing the noise variance $\sigma^2$. Concatenating the received signals corresponding to all receive beams with the $p$-th transmit beam, we have

$$y_p = W^H H k f_p + \tilde{n}_p^k \in \mathbb{C}^{M_R \times 1},$$

where the effective noise vector and its covariance matrix are expressed as follows:

$$\tilde{n}_p^k = [(W_H n_{p,1}^k)^T, \ldots, (W_H n_{p,M_R}^k)^T]^T,$$

$$R_{\tilde{n}_p^k} = \mathbb{E}[\tilde{n}_p^k(\tilde{n}_p^k)^H] = \text{Bilkl.diag}(\sigma^2 W_H^H W, \ldots, \sigma^2 W_H^H W) W^H.$$

Further concatenating the received signals corresponding to all transmit beams, we have

$$Y^k = W^H H k F + \tilde{N}^k \in \mathbb{C}^{M_R \times M_T},$$

where $\tilde{N}^k = [\tilde{n}_1^k, \ldots, \tilde{n}_{M_T}^k]$.  

1Digital beamforming and combining are not considered during channel estimation here as in [23] and [28].
B. Channel Model

In this paper, the typical clustered mmWave channel model [8], [15] is adopted and the practical beam squint effect [13], [36] in wideband mmWave massive MIMO systems is considered. Specifically, the $k$-th uplink subchannel can be expressed as

$$
H^k = \sqrt{\frac{N_T N_R}{N_c N_p}} \sum_{i=1}^{N_c} \sum_{j=1}^{N_p} \alpha_{i,j} e^{-j2\pi \tau_i f_s T} a_R(\theta^R_{i,j}, k) a_T(\theta^T_{i,j}, k)^H,
$$

where $N_c$ and $N_p$ denote the number of clusters and the number of subpaths in a cluster, respectively, and $\alpha_{i,j} \sim \mathcal{CN}(0,1)$, $\theta^R_{i,j}$ and $\theta^T_{i,j}$ denote the path gain, the angle of arrival (AoA) at the BS, and the angle of departure (AoD) at the user of the $j$-th subpath in the $i$-th cluster, respectively. $\tau_i \sim \mathcal{U}[0, \tau_{max}]$ is the delay of the $i$-th cluster with $\tau_{max}$ denoting the maximum delay. Consider half-wavelength antenna spacing, the frequency-dependent response vector of ULA\(^2\) at the BS side can be expressed as

$$
a_R(\theta^R_{i,j}, k) = \frac{1}{\sqrt{N_R}} [1, e^{-j\pi \theta^R_{i,j,k}}, \ldots, e^{-j\pi (N_R-1) \theta^R_{i,j,k}}]^T,
$$

(7)

where $\theta^R_{i,j,k} \triangleq (1 + \frac{k\tau_i}{f_s}) \sin(\theta^R_{i,j})$, $f_s$ denotes the total system bandwidth, and $f_s$ denotes the carrier frequency. Similarly, the response vector at the user side can be expressed as

$$
a_T(\theta^T_{i,j,k}) = \frac{1}{\sqrt{N_T}} [1, e^{-j\pi \theta^T_{i,j,k}}, \ldots, e^{-j\pi (N_T-1) \theta^T_{i,j,k}}]^T,
$$

(8)

where $\theta^T_{i,j,k} \triangleq (1 + \frac{k\tau_i}{f_s}) \sin(\theta^T_{i,j})$. Denote the mean AoA, mean AoD, angular spread of AoA, and angular spread of AoD of the $i$-th cluster as $\bar{\theta}^R_{i}, \bar{\theta}^T_{i}, \Delta \theta^R_{i}, \text{ and } \Delta \theta^T_{i}$, respectively, we have $\theta^R_{i,j} \sim \mathcal{U}(\bar{\theta}^R_{i} - \Delta \theta^R_{i}, \bar{\theta}^R_{i} + \Delta \theta^R_{i})$, $\theta^T_{i,j} \sim \mathcal{U}(\bar{\theta}^T_{i} - \Delta \theta^T_{i}, \bar{\theta}^T_{i} + \Delta \theta^T_{i})$, $\forall i, j$, where $\bar{\theta}^R_{i}, \bar{\theta}^T_{i} \sim \mathcal{U}[0, 2\pi]$ and $\Delta \theta^R_{i}, \Delta \theta^T_{i} \ll \pi$.

To facilitate channel estimation, the original channel needs to be converted to a sparse domain first, which is essentially the angular domain in this case due to limited channel clusters and narrow angular spread within clusters. Sample $G$ discrete angular grids, $\phi_1, \ldots, \phi_G$, from the angle space such that $\sin(\phi_i) = \frac{2i-1-G}{G}, \forall 1 \leq i \leq G$, we can obtain the transmit and receive dictionary matrices as $A_T(\phi) \triangleq [a_T(\phi_1), \ldots, a_T(\phi_G)] \in \mathbb{C}^{N_T \times G}$ and $A_R(\phi) \triangleq [a_R(\phi_1), \ldots, a_R(\phi_G)] \in \mathbb{C}^{N_R \times G}$, respectively, where $a_T(\cdot)$ and $a_R(\cdot)$ are normal frequency-independent ULA response vectors with $\varphi_{i,j,k}$ reduced to $\sin(\theta_{i,j})$ in Equations (7) and (8). Denote the equivalent angular domain channel matrix of the $k$-th subcarrier as $X^k \in \mathbb{C}^{G \times G}$, its transformation to the original channel is as

$$
H^k \approx A_R(\phi) X^k A_T^H(\phi),
$$

(9)

where the approximation error comes from the finite angular resolutions of dictionary matrices. Since the number of clusters is usually small and the angular spreads of clusters are usually narrow, $X^k$ is sparse with most of its elements close to 0.

C. Channel Sparsity Structures

Proper exploitation of channel sparsity structures is critical to improve channel estimation performance [8], [26]. An example of the wideband mmWave massive MIMO channel is given in Fig. 2, from which two main sparsity structures are observed:

- **Circular Block Sparsity Structure**: Due to the clustered distribution of channel paths, the angular domain channel exhibits block sparsity such that nearby elements tend to have close levels of modulus, as shown in the shaded areas circled in Fig. 2. Then, the power leakage effect\(^3\) further makes the block sparsity circular, i.e., elements on the top and bottom (or the left and right) edges also have close levels of modulus. For instance, the actual AoAs of subpaths in one of the clusters of the given exemplary channel are around $90^\circ$, which correspond to the red circle on the bottom edge in Fig. 2(a), while elements in another red circle on the top edge are also significantly non-zero.

- **Shifted Common Sparsity Structure**: The common sparsity structure among different subchannels has been widely exploited in the existing works [7], [32], which, however, becomes shifted with the beam squint effect. Comparing Fig. 2(a) and Fig. 2(b), we can see that the locations of circles corresponding to the same cluster shift between different subchannels due

\(^2\)With large bandwidth, the propagation delay across the large antenna array becomes comparable to the time-domain sample period, a.k.a. the spatial-wideband effect. In frequency domain, it turns into the beam squint effect such that different subcarriers will “see” different angles of a same channel path. Please refer to [13] and [36] for detailed derivation.

\(^3\)The real channel angles do not necessarily lie on the predefined angular grids, and a channel path whose angle is not equal to any of the grids will have significantly non-zero responses on multiple nearby grids.
Therefore, Equation (5) can be written in the standard CS setting introduced in Section VI.

D. Problem Formulation

Stacking the columns of \( Y^k, H^k, X^k, \) and \( \tilde{N}^k \) yield \( y^k = \text{vec}(Y^k) \in \mathbb{C}^{M_k M_T K}, \) \( h^k = \text{vec}(H^k) \in \mathbb{C}^{N_k N_R K}, \) \( x^k = \text{vec}(X^k) \in \mathbb{C}^{G^2 K}, \) and \( \tilde{n}^k = \text{vec}(\tilde{N}^k) \in \mathbb{C}^{M_k M_T K}, \) respectively. Then, exploiting the property of Kronecker product that \( \text{vec}(ABC) = (C^T \otimes A)\text{vec}(B), \) we have \( h^k = (A^T_r(\phi) \otimes A_R(\phi))x^k \) and \( y^k = (E^T \otimes W^H)h^k + \tilde{n}^k. \) Therefore, Equation (5) can be written in the standard CS form as

\[
y^k = \Phi x^k + \tilde{n}^k, \quad (10)
\]

\[
\Phi \triangleq (E^T \otimes W^H)(A^T_r(\phi) \otimes A_R(\phi)) \in \mathbb{C}^{M_k M_T \times G^2}, \quad (11)
\]

where \( \Phi \) is called the measurement matrix and the covariance matrix of \( \tilde{n}^k \) is \( \tilde{R}_n = I_{M_T} \otimes \tilde{R}_{n_k}. \)

Combining the estimation process of all \( K \) subchannels together, we have

\[
Y = \Phi X + \tilde{N}, \quad (12)
\]

where \( Y = [y^1, \cdots, y^K] \in \mathbb{C}^{M_k M_T \times K}, \) \( X = [x^1, \cdots, x^K] \in \mathbb{C}^{G^2 \times K}, \) and \( \tilde{N} = [\tilde{n}^1, \cdots, \tilde{n}^K] \in \mathbb{C}^{M_k M_T \times K}. \) For uplink channel estimation, we aim to design a matrix \( \Phi \) for efficient channel measurement and an estimation algorithm \( f(\cdot) \) to accurately recover \( H = [h^1, \cdots, h^K] \in \mathbb{C}^{N_k N_R \times K} \) based on \( Y, \Phi, \) and \( \tilde{R}_n. \) Besides, beam elements are restricted to have constant modulus due to the phase-shifter-based implementation. The optimization problem is as follows:

\[
P1 : \min_{\Phi, f(\cdot)} \mathbb{E}\{ \|f(\Phi, Y, \tilde{R}_n) - H\|^2 \} \quad \text{s.t.} \quad \|W\|_{i,j} = \frac{1}{\sqrt{N_R}}, \forall i, j, \quad \|F\|_{i,j} = \frac{1}{\sqrt{N_T}}, \forall i, j. \quad (13)
\]

The original noise variance \( \sigma^2 \) can be calculated offline based on noise power spectrum density and bandwidth, or be estimated online periodically. Then, the effective noise covariance matrix can be constructed based on \( \sigma^2 \) and the designed \( \Phi. \)

III. CONVENTIONAL SBL-BASED CHANNEL ESTIMATION APPROACHES

Before diving into the proposed DL-based approach, in this section, we will first clarify the superiorities of the SBL algorithm in the considered problem. Then, the underlying principles of conventional SBL-based channel estimation approaches will be introduced. For the considered problem, SBL is particularly suitable due to the following reasons. First, SBL has better performance than \( l_1 \) penalty-based CS algorithms such as lasso and basis pursuit since the most sparse solution can be obtained [37]. Intuitively, SBL is shown to be equivalent to an iterative reweighted \( l_1 \) minimization algorithm where the \( l_1 \) algorithm is only the first step, which makes its performance superiority easy to understand. Besides, SBL can flexibly exploit various channel sparsity structures thanks to the adopted parameterized Gaussian prior distribution. Specifically, by configuring the parameter sharing mechanism beforehand or incorporating information from other elements during parameter update, sparsity structures can be naturally embedded or gradually reinforced through iterations in SBL. Last but not least, SBL is robust to general measurement matrices such as the highly structured one in Equation (11) since it can be seen as implicitly utilizing a complicated matrix-dependent penalty function [38]. In contrast, most other CS algorithms would suffer from performance deterioration when the measurement matrix does not satisfy good properties like RIP.

Consider the CS problem in Equation (10), to estimate the \( k \)-th subchannel with SBL, the angular channel vector is modelled as following the complex Gaussian distribution with a diagonal covariance matrix:

\[
x^k \sim \mathcal{CN}(0, \tilde{R}_{x^k}), \quad (14)
\]

\[
\tilde{R}_{x^k} = \text{diag}(\gamma^k) \in \mathbb{R}^{G^2 \times G^2}, \quad (15)
\]

where \( \gamma^k = [\gamma_1^k, \cdots, \gamma_G^k] \) denotes the variance parameters of the elements of \( x^k. \) To promote sparsity in the estimated vector, the inverse of each variance parameter is assumed to follow a Gamma distribution with shape parameter \( a \) and inverse scale parameter \( b, \) such that:

\[
p(\alpha^k) = \prod_{i=1}^{G^2} \Gamma(a)\Gamma(b) = \Gamma(a+b)\Gamma(a+b-k)\frac{\lambda^k}{\Gamma(b)^a}, \quad (16)
\]

where \( \Gamma(\cdot) \) denotes the Gamma function and \( \lambda^k = 1/\gamma^k, \forall k, i. \) It can be shown that the prior distribution of \( \alpha^k \) with respect to \( a \) and \( b \) is a Student-t distribution, which is known to be sparsity promoting with sharp peak at zero when \( a \) and \( b \) are small, e.g., \( 10^{-4} \) [39]. To find an estimate of \( \alpha^k, \) a lower bound of the posterior density \( p(\alpha^k | y^k) \) is maximized using the expectation-maximization (EM) algorithm, which leads to the following update steps in each iteration [37]:

- **E-step:** Evaluate the posterior mean and the posterior covariance:

\[
\mu_{x^k} = \tilde{R}_{x^k} \Phi^H (\Phi \tilde{R}_n \Phi^H + \tilde{R}_{x^k})^{-1} y^k, \quad (17)
\]

\[
\Omega_{x^k} = \tilde{R}_{x^k} - \mu_{x^k} \Phi^H (\Phi \tilde{R}_n \Phi^H + \tilde{R}_{x^k})^{-1} \Phi \mu_{x^k}. \quad (18)
\]
- **M-step**: Update variance parameters:
  \[
  \gamma^k = |\mu_{x^k}|^2 + \text{diag}(\Omega_{x^k}).
  \] (19)

After initializing \( R_{x^k} \) as \( I_{G^2} \), the above steps are executed iteratively until convergence, and the eventual posterior mean is regarded as the estimation of \( x^k \). However, the original SBL algorithm is designed for sparse vectors with independent elements. To exploit various channel sparsity structures, the following variants of SBL can be used for performance improvement:

- **M-SBL**: To exploit the common sparsity structure in the frequency domain, instead of running SBL for \( K \) times to estimate different subchannels separately, M-SBL estimates all subchannels together, where “M” stands for MMV [8]. In M-SBL, \( K \) subchannel vectors \( x^1, \ldots, x^K \) share common \( \gamma \) and \( \Omega_x \). Therefore, Equation (19) is modified to
  \[
  \gamma = \frac{\sum_{k=1}^K |\mu_{x^k}|^2}{K} + \text{diag}(\Omega_x). \] (20)

- **PC-SBL**: To exploit the block sparsity structure in the angular domain, variance parameters of nearby elements in PC-SBL are designed to be entangled with each other, where “PC” stands for pattern-coupled [40]. First, \( \gamma^k, |\mu_{x^k}|^2 \), and \( \text{diag}(\Omega_{x^k}) \) are arranged into \((G, G)\)-dimensional matrices \( \Gamma^k_1, \Gamma^k_2 \), and \( \Gamma^k_3 \) in the order of angular grids’ sine values. Introducing the \((G, G)\)-dimensional auxiliary variable matrix \( A^k \), we have
  \[
  [\Gamma^k_{ij}] = \left( \frac{[A^k]_{ij} + \beta([A^k]_{i-1,j} + [A^k]_{i+1,j} + [A^k]_{i,j-1} + [A^k]_{i,j+1})}{\beta} \right)^{-1}, \forall i, j,
  \] where \( \beta \) denotes the coupling parameter. To update \( A^k \), we have \([A^k]_{ij} = \frac{0}{\omega_{i,j} + \beta}, \forall i, j \)
  \[
  \omega_{i,j} = |F^1_{i,j} + F^2_{i,j} + \beta([F^1_{i,j}]_{i-1,j} + [F^2_{i,j}]_{i-1,j} + [F^1_{i,j}]_{i+1,j} + [F^2_{i,j}]_{i+1,j} + [F^1_{i,j}]_{i,j-1} + [F^2_{i,j}]_{i,j-1} + [F^1_{i,j}]_{i,j+1} + [F^2_{i,j}]_{i,j+1}).
  \] (21)

Notice that, when the indices are out of bounds (\( i \) or \( j \) equals 0 or \( G + 1 \)) in all \( A^k, F^1_k, F^2_k \), the corresponding elements are set to 0 [40]. When \( \beta = 0, a = 0.5, b = 0 \), PC-SBL reduces to SBL.

- **M-PC-SBL**: To exploit both sparsity structures simultaneously, we can naturally combine M-SBL and PC-SBL. Similar to M-SBL, common \( \gamma \) and \( \Omega_x \) are shared by \( K \) subchannels, and \( F^1_k \) in Equation (21) is replaced by \( \frac{\sum_{k=1}^K F^k_{1i}}{K} \).

Nevertheless, the above SBL variants have limitations in the considered problem. Specifically, in PC-SBL, hyperparameters need to be carefully selected and the variance update rule is suboptimal [10]. Besides, the circularity of block sparsity is not considered. For M-SBL, its gain will be partially or even entirely offset by beam squint since the common sparsity among subchannels becomes shifted. For the same reason, M-PC-SBL may have worse performance than PC-SBL since averaging over all subchannels may enlarge the equivalent noise and make the local entanglement information more inaccurate.

**IV. DL-BASED CHANNEL ESTIMATION APPROACH**

To overcome the shortcomings of conventional SBL-based approaches, we propose a DL-based channel estimation approach in this section. Next, the overall framework, network architecture design, training details, and complexity analysis will be elaborated sequentially.

**A. Framework of SBL Unfolding**

In general, the SBL algorithm with multiple iterations is unfolded into a DNN with cascaded layers, called the “SBL-CE-Net”, and the update rule of variance parameters in each SBL layer of the SBL-CE-Net is realized by a dedicated DNN. To facilitate the exploitation of the shifted common sparsity structure in the frequency domain, a unique variance parameter is assigned to each element of \( X \) unlike M-SBL that uses shared variance parameters among different subchannels. Denote the variance parameter matrix in each SBL layer as \( \Gamma = [\gamma_1, \ldots, \gamma_K] \in \mathbb{R}^{G^2 \times K} \). As illustrated in Fig. 3, in an \( L \)-layer SBL-CE-Net, the input of the \( l \)-th SBL layer is \( Y, \Phi, R_{\mathbf{a}}, \) and \( \Gamma^{l-1} \), and the output is the updated \( \Gamma^l \). The all-one \( \Gamma^0 \) is input to the first SBL layer as the initialization of variance parameters, and the angular domain channel estimation \( X \) can be readily obtained according to Equation (17) after \( \Gamma^L \) is output by the last SBL layer.

Stack all subchannels’ posterior mean vectors’ squared modulus and posterior covariance matrices’ diagonal vectors to obtain two feature matrices \( F_1 = [|\mu_{x^1}|^2, \ldots, |\mu_{x^K}|^2] \in \mathbb{R}^{G^2 \times K} \) and \( F_2 = [\text{diag}(\Omega_{x^1}), \ldots, \text{diag}(\Omega_{x^K})] \in \mathbb{R}^{G \times K} \).

Then, the optimal variance parameter update rule in each SBL layer, although may not be mathematically derivable under complex circumstances, should be a mapping function in the form of \( \Gamma_{\text{updated}} = g(F_1, F_2) \), which can be parameterized by a DNN\(^5\) as \( g_{\text{opt}}(\cdot; \Theta_{\text{opt}}) \) and the optimal network parameter set \( \Theta_{\text{opt}} \) can be learned through training. Notice that, the update rules used in M-SBL and PC-SBL can be regarded as specific examples of \( g(\cdot) \).

As for the measurement matrix, instead of using hand-crafted matrices for \( W \) and \( F \) such as those consisting of random phases or Zadoff-Chu sequences [11], we jointly optimize them with the channel estimator. Specifically, the phases of phase shifters at transceivers are modelled as trainable parameters. Denote a certain phase as \( \theta \), the corresponding complex element of \( W \) or \( F \) can be obtained by \( e^{j2\pi\theta/\sqrt{N_R}} \).

\(^5\)In this paper, we train different parameters for DNNs in different SBL layers to achieve high performance. Nevertheless, DNNs in all SBL layers can also share the same parameters to improve parameter efficiency at the cost of slight performance degradation.
Channel sparsity structures are adopted in the proposed network architecture to exploit SBL layer’s feature computation. The following key designs or Fig. 4. Network architecture of the DNN in the l-th SBL layer.

or $e^{j2\pi \theta} / \sqrt{N_T}$. Thanks to the periodicity of function $e^{j2\pi \cdot (-)}$, $\theta$ can be unconstrained. Using this method, the constant modulus constraints of the elements of $W$ and $F$ are naturally satisfied. During training, the phases can be updated together with the weights of all SBL layers’ DNNs [32], [35]. After training, optimal $W$ and $F$ can be readily constructed based on the learned phases and deployed at the BS side and user side, respectively.

B. Network Architecture

Although DNNs are universal approximators theoretically, the specific architecture is critical to the performance of a DNN in practice. First of all, as illustrated in Fig. 2, the circular block sparsity exists in the two-dimensional AoA-AoD space. Therefore, similar reshaping as in PC-SBL is performed to facilitate the update of variance parameters. Specifically, in the l-th SBL layer, the $(G^2, K)$-dimensional feature matrices $F_l^{i-1}$ and $F_l^{j-1}$ are converted to $(G, G, K)$-dimensional tensors where AoA and AoD are two separate dimensions, and the $(G, G, K)$-dimensional tensor of variance parameters is updated by the DNN, which is then converted back to the $(G^2, K)$-dimensional matrix, i.e., $\Gamma_l^i$, for the $(l + 1)$-th SBL layer’s feature computation. The following key designs are adopted in the proposed network architecture to exploit channel sparsity structures:

- **3D Convolution**: Both the block sparsity in the angular domain and the shifted common sparsity in the frequency domain result in local correlation in the input feature tensor, which can be well captured by the convolution operation. We use 3D convolution that is often used to process video data [41], where filters slide over the input tensor along its three dimensions, namely AoA, AoD, and subcarrier, to obtain the output tensor. Intuitively, each subchannel can be viewed as a gray-scale frame, and the shift among subchannels is similar to the slight changes of consecutive frames of a video. Compared to the fully-connected (FC) layer used in [42], convolutional layer not only has better performance, but also has much lower complexity in large scale systems.

- **Circular Padding**: In CNNs, padding is a widely adopted technique to keep the dimensions of feature maps unchanged after convolution. In the existing CNN-based channel estimation works [23], [35], zero padding (ZP) is usually used around the feature map before convolution. To deal with the circularity of block sparsity in the angular domain, we propose to use circular padding (CP) that is often used to process panoramic images [43], where the top (and left) part is copied and concatenated to the bottom (and right) of the feature map, and vice versa. Notice that, ZP is still used in the subcarrier dimension.

- **Position Features Input**: One of the key features of CNN is that the convolution filters are shared and position-independent processing are executed when filters slide over the input feature map. However, unlike natural images, in the considered problem, the local correlation patterns in the feature tensor are naturally position-dependent. For instance, the squint directions in different AoA-AoD sections are different, as illustrated by the arrows in Fig. 2. To realize position-dependent dynamic processing, we explicitly inform convolution filters of their current position [44] by incorporating the sine values of the covered angular grids’ AoAs and AoDs as additional features. Denote the $(G, G, K)$-dimensional position features as $F_3$, we have $[F_3]_{i,j,k,0} = \sin(\phi_i)$ and $[F_3]_{i,j,k,1} = \sin(\phi_j), \forall i,j,k$.

The detailed architecture of the DNN in the l-th SBL layer is illustrated in Fig. 4. The $(G, G, K, 4)$-dimensional input feature map is obtained by stacking $F_1, F_2, F_3$ along the last dimension. We use two 3D convolutional layers with CP (C-Conv3D), where the first layer has $N_F$ filters to extract intermediate features and the second layer has only one filter to output the updated $\Gamma_l^i$. The filter size is $F_S$. ReLU activation is used after both layers, where the first ReLU introduces nonlinearity and the second ReLU outputs non-negative variance predictions.

C. Network Training

For network training, we generate 10,000 channel samples in total based on the channel model in (6), and split them into a training set, a validation set, and a testing set with a ratio of 8:1:1. Adam is used as the optimizer. The mean-squared error (MSE) is used as the loss function, i.e., $\text{Loss} = \frac{1}{N_{\text{batch}}N_TN_R} \| H_s - H_s^\prime \|^2$, where the number of samples in a mini-batch, $S$, is set to 16 in experiments. A multi-stage training strategy is used to improve the speed and stability of training, as well as the converged performance. Specifically, in the first stage, the weights of convolution filters are initialized using Xavier [45] and trained until convergence. For different channel samples, different $W$ and $F$ whose elements’ phases are sampled from $U(0, 2\pi]$ are applied so...
that the trained channel estimator is applicable to next stage measurement matrices and will not fail in the next stage when the measurement matrix is optimized. Then, in the second stage, the convolution filters are frozen, i.e., the channel estimator is fixed and only the phases of $W$ and $F$ are trained to obtain the optimal measurement matrix. In the first two stages, the learning rate is initialized as $10^{-5}$ and decays with a factor of 10 and a patience of 2 epochs to accelerate training, and early stopping with a patience of 3 epochs is used to avoid overfitting. In the third stage, the entire network is fine-tuned with a small learning rate $10^{-5}$, where all trainable parameters are trained together. Again, early stopping with a patience of 3 epochs is used.

### D. Complexity Analysis

The number of real floating operations (FLOPs) is calculated to compare the complexity of different algorithms. For algorithms in the SBL family, the FLOPs per iteration of complexity-dominating operations is $16K G^2 (M R M F)^2$. As for the proposed SBL-CE-Net, the extra real FLOPs per iteration introduced by the DNN is $(5N_F + 2)F_B^3 K G^2$. We only demonstrate the online prediction complexity here since the offline training does not happen very often and the BS is assumed to have sufficient memory and computation resources. Using an NVIDIA GeForce RTX 3090 GPU, the typical network training time is several hours. With typical system and network parameters, the increase of complexity per iteration of SBL-CE-Net is marginal while the number of iterations can be dramatically reduced compared to SBL, as will be shown later in simulation. Therefore, the overall complexity of SBL-CE-Net is much lower than SBL.

### V. EXTENSION TO THE MULTI-BLOCK CASE

So far we have been discussing channel estimation in a single block. In this section, we further extend the proposed approach to the multi-block case and exploit channel correlation in the time domain for further performance improvement.

#### A. Time-Selective Channel Model

Since the angles and delays of channel paths change relatively slow, it is reasonable to assume that they can be modeled for several consecutive blocks [8], [23]. The $k$-th subchannel in the $n$-th block can be expressed as

$$H^{k}[n] = \frac{\sum_{i=1}^{N_c} \sum_{j=1}^{N_T} \alpha_{i,j}[n] e^{-j2\pi f s k} a_R(\theta_{i,j}, k)}{\sum_{i=1}^{N_c} \sum_{j=1}^{N_T} \alpha_{i,j}[n] e^{-j2\pi f s k}} a_T(\theta_{i,j}, k)^H,$$

(22)

where the evolution of path gains in time follows the first order autoregressive process as

$$\alpha_{i,j}[n] = \rho \alpha_{i,j}[n-1] + \sqrt{1-\rho^2} w_{i,j}[n], \forall i,j,$$

(23)

and the temporal correlation coefficient can be computed according to the Jakes’ model as $\rho = J_0(2\pi f_D \Delta t)$, where $J_0$ is the zeroth-order Bessel function of the first kind, $\Delta t$ is the block length, and $f_D = v f_c / c$ is the maximum Doppler frequency with $v$ and $c$ denoting the speed of user and light, respectively [8]. Besides, $w_{i,j}[n] \sim \mathcal{CN}(0,1)$ denotes the innovation noise. The received signal in the $n$-th block can be expressed as

$$Y[n] = \Phi[n] X[n] + \tilde{N}[n],$$

(24)

where the measurement matrix here can also be time-varying rather than long-term fixed.

#### B. Problem Reformulation

To exploit channel correlation in the time domain, one method is to estimate the channels of multiple blocks together by using algorithms like M-SBL [8]. Nevertheless, in this way, the channel estimations of previous blocks are not available until the last block’s pilots arrive, and the complexity increases with the number of consecutive blocks. Therefore, a more proper way is to estimate online and exploit previous blocks’ estimation results to aid current block’s estimation [8]. In this paper, we only exploit one previous block for simplicity while exploiting several previous blocks is totally feasible. We use the modulus of angular domain channel estimation, $|\hat{X}[n-1]|$, as the time features since they reflect the short-term channel sparsity pattern. Then, the time features are exploited in two aspects, namely facilitating the update of variance parameters and designing the adaptive measurement matrix. The former is based on that consecutive blocks’ angular domain channels have similar positions of non-zero elements while the latter is based on that the measurement efficiency of the measurement matrices can be improved by focusing energy on angular grids with high confidence [35]. The optimization problem is as follows:

$$\begin{align*}
P2 : \min_{\Phi[n]} \mathbb{E}\left\{ \frac{||f'(\Phi[n], Y[n], R_n[n], \hat{X}[n-1]) - H[n]||^2}{||H[n]||^2} \right\} \\
\text{s.t.} \quad \Phi[n] = q(\hat{X}[n-1]).
\end{align*}$$

(25)

where $f'(\cdot)$ and $q(\cdot)$ denote the function of channel estimation and the function of adaptive measurement matrix prediction, respectively.

#### C. Modifications to the Network Architecture

To realize the multi-block channel estimation, the network architecture illustrated in Fig. 5 is adopted. Compared to Fig. 4, the first modification is to incorporate the $(G, G, K, 1)$-dimensional time features $|\hat{X}[n-1]|$, i.e., $F_4$, as an extra part of input of the DNN in each SBL layer. The second modification is that another dedicated DNN is used for measurement matrix prediction. Notice that only the receive combining matrix $W[n]$ is adaptively predicted and the transmit beamforming matrix $F$ is still fixed since it is deployed at the user side while previous blocks’ estimation results are only available at the BS in TDD systems. Specifically, a two-dimensional global average pooling (GAP2D) layer is used first to average out the dimensions of subcarrier and AoD in $F_4$ since all subcarriers share a common measurement.

---

6Matrix $F$ can also be predicted and deployed at the cost of extra feedback overhead, which is not considered in this paper.
parameters are the same as those introduced in Section IV-C, and early stopping are still used. The specific values of the adopted training rate. Strategies including learning rate decay and early stopping are ultimately, the entire network is fine-tuned with a small learning rate.

After convergence, the convolution filters are frozen and the previously optimized $F$ in the single-block case, $\Phi[n]$ can be readily computed.

D. Network Training and Complexity Analysis

In the multi-block case, the modified SBL-CE-Net is trained alone first. For the initialization of weights of convolution filters in two C-Conv3D layers, those connecting to the newly added $F_1$ are initialized as zero while the rest are copied from the previously trained model in the single-block case, so that the function of the modified SBL-CE-Net is exactly the same as the original one at the beginning of training. After convergence, the convolution filters are frozen and the dedicated DNN for matrix prediction is introduced and trained to convergence, whose weights use Xavier initialization. Eventually, the entire network is fine-tuned with a small learning rate. Strategies including learning rate decay and early stopping are still used. The specific values of the adopted training parameters are the same as those introduced in Section IV-C, including the numbers of datasets, the learning rates, and the patience epochs.

Compared to the single-block approach, the extra complexity is only marginal while decent performance gain can be achieved, as will be shown later in simulation.

VI. SIMULATION RESULTS

In this section, extensive simulation results are provided to validate the superiority of the proposed approaches in terms of performance, complexity, and robustness. The source code is available at https://github.com/EricGJB/SBL_unfolding_CE.

A. Configuration

Unless specified, the following commonly adopted system and channel parameters [12], [30] will be used as the default setting in simulation: $N_T = 32$, $N_R = 32$, $N_{RF} = 4$, $N_c = 3$, $N_p = 10$, $\Delta t = 5^\circ$, $f_c = 28$ GHz, $f_s = 4$ GHz, $K = 8$, SNR = 20 dB, $v = 1$ m/s, and $\Delta t = 1$ ms so that the corresponding $\rho = 0.916$. Normalized mean-squared error (NMSE) defined as $E[||H - \hat{H}||^2 / ||H||^2]$ is utilized to measure the channel estimation performance obtained by averaging over 300 Monte Carlo channel and noise realizations. The following algorithms are selected as baselines for comparison:

- **The SBL family** [8], [10]: SBL and PC-SBL are executed for $K$ times to estimate different subchannels separately while M-SBL and M-PC-SBL with the best searched hyperparameters are executed once to estimate all subchannels together. Iterations are executed until convergence.
- **LS with sufficient beams**: Use $M_R = N_R$ and $M_T = N_T$ beams to obtain the LS estimation by $\hat{H}_{LS} = (W^H)^{-1}Y_k F^{-1}$, which needs $N_T N_R$ times overhead of all the other algorithms.
- **Data-driven DL** [25]: Obtain coarse estimation with a FC layer first and use CNN for further refinement. The measurement matrix is jointly optimized as well. In the considered problem, this method performs better than the other two data-driven methods in [23] and [26].

**Fig. 5.** Network architecture adopted in the multi-block case.
channel sparsity structures simultaneously using conventional SBL variants. and PC-SBL, which indicates that beam squint makes it hard to exploit both of ZP in the angular domain, channel paths near the edges of averaging over all subchannels.

Notice that, some other greedy CS algorithms such as SOMP, group-LASSO, and MMV-AMP are not included as baselines for neatness, since results in [8], [12], and [32] have already demonstrated significant performance superiority of the above baselines to them.

B. Determination of the Network Architecture

Eventually, \( L = 3, N_F = 8, F_S = 5 \) are determined through cross validation. As an example, the impact of the number of SBL layers \( L \) is illustrated in Fig. 6, where \( M_T = 16, M_R = 16, G = 64 \), and other parameters are default. We can see the network performance saturates with 3 SBL layers and more layers only improves performance slightly but leads to higher complexity.

The effectiveness of each specific design adopted in the proposed network architecture is validated in Table I, where \( M_T = 16, M_R = 8, G = 64 \), and other parameters are default. As we can see, the network performance gradually improves from the vanilla version with more designs used, and eventually exceeds all conventional SBL-based algorithms. Specifically, the performance of Conv2D improves dramatically with the input of position features and becomes better than PC-SBL, which demonstrates the effective exploitation of the block sparsity structure in the angular domain. Then, Conv3D outperforms Conv2D due to further exploitation of the channel sparsity structure in the frequency domain. In contrast, M-PC-SBL has no performance gain over PC-SBL with best hyperparameters searched being \( \beta = 0, a = 0.5, b = 0 \) since beam squint damages the common sparsity structure and makes the local entanglement information less accurate after averaging over all subchannels.\(^7\) At last, by using CP instead of ZP in the angular domain, channel paths near the edges of

\(^7\)Notice that, without beam squint, M-PC-SBL does outperform M-SBL and PC-SBL, which indicates that beam squint makes it hard to exploit both channel sparsity structures simultaneously using conventional SBL variants.

C. Performance Analysis With a Single Block

In this subsection, we analyze the performance of different algorithms in the single-block case.

1) Impacts of Key System Parameters: First, we consider that the network is trained with fixed system and channel parameters, while its generalization ability will be investigated later. Different random \( W \) and \( F \) are applied to different channel samples, which corresponds to the first training stage in SBL-CE-Net. As illustrated in Fig. 7(a), the performance of all algorithms improves with the growth of \( G \) at the beginning since dictionary matrices with higher angular resolutions can alleviate power leakage and enhance the sparsity of the angular domain channel [15]. Then, when \( G \) is large enough, the performance saturates. Similarly, the impact of the number of receive beams is illustrated in Fig. 7(b). With more information brought by more receive beams, the performance gets better at the cost of higher estimation overhead and complexity. Same trends can be observed when changing the number of transmit beams. From Fig. 7, we can see that the proposed SBL-CE-Net consistently outperforms conventional SBL algorithms with various \( G \) and \( M_R \), demonstrating its superiority and generality. To balance performance, complexity, and overhead, \( G = 64, M_T = 16, M_R = 16 \) will be used afterwards. The impact of measurement matrix is illustrated in Fig. 8(a). With the jointly optimized measurement matrix (solid curves), in various SNR regimes, all SBL-based methods achieve decent performance gain compared to their counterparts with the widely used random measurement matrices (dashed curves) thanks to higher measurement efficiency and better matrix properties for sparse signal recovery. Then, the performance of different algorithms with the optimized measurement matrix is shown in Fig. 8(b). First of all, the poor performance of data-driven DL confirms the necessity of using model-driven approaches. Aided by domain knowledge, MMV-LAMP outperforms LS and SBL in low SNR regimes. However, it’s performance is still limited by the selected AMP model and the ignorance or overly ideal assumption about channel sparsity structures. Similar problem occurs in M-SBL and PC-SBL. In contrast, SBL-CE-Net is able to learn arbitrary sparsity structures and achieves much more significant performance gain.

2) Generalization Ability: Strong generalization ability is critical to make a DL-based method practical. First of all, the proposed approach can naturally generalize to different system scales including \( N_T, N_R, M_T, M_R \) since convolution only happens in the angular domain and frequency domain. Furthermore, even if with different \( G \) and \( K \), the same network can be re-used thanks to the parameter sharing mechanism of CNN. On the other hand, parameters influencing the channel distribution could be time-varying in practice, such as SNR, the number of clusters, and the angular spread, whose impact needs to be investigated.

Among several common methods to deal with SNR generalization [26], [35], [46], we choose to test with the model
TABLE I
EFFECTIVENESS OF THE PROPOSED NETWORK ARCHITECTURE. CONV2D ONLY PERFORMS CONVOLUTION IN THE AOA-AoD SPACE AND DIFFERENT SUBCHANNELS ARE ESTIMATED SEPARATELY

| Type                          | Approach                        | NMSE  |
|-------------------------------|---------------------------------|-------|
| The SBL family                | SBL                             | 0.1203|
|                               | PC-SBL                          | 0.1015|
|                               | M-SBL                           | 0.0803|
|                               | M-PC-SBL                        | 0.0803|
| The SBL unfolding family      | Conv2D + ZP                     | 0.3127|
|                               | Conv2D + ZP + Position features | 0.0710|
|                               | Conv3D + ZP + Position features | 0.0429|
|                               | Conv3D + CP + Position features | 0.0333|

Fig. 7. Impacts of the number of grids $G$ and the number of receive beams $M_R$.

Fig. 8. Impacts of the measurement matrix and SNR.

trained by a moderate SNR point [35] since it is simple and can already achieve satisfactory generalization performance. As illustrated in Fig. 9(a), the performance of testing with the network trained by only 10 dB data is almost the same as testing with individual networks trained by accurate SNR points, except for slight performance degradation when $\text{SNR} = 0$ dB. The trends are similar when the optimized measurement matrix is used.

The generalization to different numbers of clusters and angular spreads is illustrated in Fig. 9(b). As we can see, the network trained with 3-cluster channel has no performance loss when testing 4-cluster channel and vice versa. Regarding the appearance of a cluster as a pattern, no matter how many clusters there are in the channel image, they can be captured when the convolution filters corresponding to this pattern slide over. As for angular spread, network trained with larger-angular-spread channel generalizes well to smaller-angular-spread channel since the latter can be regarded as the special case of the former. However, there is obvious performance degradation in the contrary case, as indicated by the red curve. Therefore, we should train the network with large-angular-spread channel to ensure robustness. Thanks to the strong generalization ability, only a single network needs to be trained in practice to handle various situations,
Fig. 9. Generalization to key channel-related parameters. Random matrices are used. In the legends of the second subfigure, the two numbers before (or after) the arrow denote the number of clusters and the angular spread of the training set (or testing set), respectively.

Fig. 10. Performance gain of the proposed multi-block approach.

which dramatically enhances the practicality of the proposed approach.

D. Performance Analysis With Multiple Blocks

In this subsection, we demonstrate the performance gain in the multiple-block case. From Fig. 10, with the jointly optimized measurement matrix, the incorporation of time features in SBL-CE-Net facilitates the update of variance parameters and decreases the NMSE slightly in all SNR regimes. Then, the performance further improves decently with the predicted measurement matrix, which validates the effectiveness of the proposed designs in the multi-block approach.

The generalization ability in the multi-block case is also tested. On the one hand, when the trained network is tested with a different temporal correlation factor $\rho = 0.707$, no performance degradation is observed (NMSE keeps unchanged at 0.0019) since the most important information exploited in the proposed approach is the short-term locations of non-zero angular grids, which can always be reflected in time features regardless of the value of $\rho$. On the other hand, since it may be too strict to assume constant path angles in consecutive blocks, for each new block, a $\mathcal{U}[-3^\circ, 3^\circ]$ disturbance is added to the AoAs and AoDs of all subpaths in each cluster [47]. Again, no obvious performance degradation is observed (NMSE increases from 0.0019 to 0.0020).

To promote intuitive understanding of the superiority of the adaptively predicted measurement matrix, two examples are given in Fig. 11, where three curves are displayed in each subfigure. The curve with the circle marker demonstrates the average energy distribution of the angular domain channel, whose $g$-th point is obtained by $\frac{1}{KG}\sum_{k=1}^{K}\sum_{g'=1}^{G}|X_{k}^{g}|^2$. The other two curves are the average measurement energy distributions of different measurement matrices, whose $g$-th point is obtained by $||[W]^HAR(\Phi)_{,g}||$ or $||[W_{n}]^HAR(\Phi)_{,g}||$. Notice that, since only $W$ deployed at the user side for AoA sensing is adaptively predicted, other dimensions, including the AoD dimension and the subcarrier dimension, are averaged out to highlight the difference in the AoA dimension when computing these energy distributions. Besides, the curves are scaled properly for the best visual effect. Intuitively, the above two types of energy distributions can be seen as the short-term channel sparsity pattern and the measurement energy allocation, respectively. For a good measurement matrix, they should match well so that more energy can be allocated to measuring angles with rich channel information and less energy is wasted on measuring angles with little channel information. As we can see, in both examples, the jointly optimized matrix in the single-block case has dispersive measurement energy throughout AoA grids since it is long-term fixed and has to consider all possible path angles fairly. In contrast, the adaptively predicted matrix has focused measurement energy on grids that are highly possible to have incoming paths, which are inferred by the DNN in Fig. 5 from the time features. The perfect match with the short-term channel sparsity pattern well explains the performance gain of the adaptively predicted matrix. Furthermore, we remark that although lots of angles without incoming channel paths still correspond to moderate measurement energy in the predicted matrix due to limited degree of optimization freedom, brand new paths reflected from scatters that appear in the current block can still be measured accurately with sufficient energy and it is actually beneficial to enhance the matrix’s robustness to non-stationary channel.

E. Complexity Comparison

The specific numbers of FLOPs and the average running time on the same CPU of different approaches under different system settings are shown in Table II. Although the
per-iteration complexity of the proposed DL-based approach is slightly higher than the original SBL algorithm with the extra DNN part, its overall complexity is much lower than SBL thanks to much fewer layers, i.e., iterations, required for convergence. Besides, the additional complexity in the multi-block approach is only marginal. Consistently, the average running time of DL-based approaches is much shorter than SBL. Under the default system setting, the multi-block DL-based approach takes less than 1 second while SBL takes near 28 seconds. Therefore, the superiority of the proposed approach is demonstrated in both performance and complexity, making it a very promising solution for channel estimation in practical HAD massive MIMO systems.

VII. CONCLUSION

In this paper, we have proposed a DL-based approach for wideband HAD mmWave massive channel estimation. The SBL algorithm is deeply unfolded and a tailored DNN is used to update Gaussian variance parameters in each SBL layer. The extension to the multi-block case has also been made. With effective and efficient exploitation of channel sparsity structures in various domains and the optimized or predicted measurement matrix, the proposed approaches demonstrate superiority over existing approaches in terms of both performance and complexity, and the strong generalization ability further enhances the practicality. In the future, we will consider to exploit channel sparsity in the delay domain and develop computationally cheaper E-step for further performance improvement and complexity reduction of the proposed SBL unfolding framework.

REFERENCES

[1] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, “An overview of massive MIMO: Benefits and challenges,” IEEE J. Sel. Topics Signal Process., vol. 8, no. 5, pp. 742–758, Oct. 2014.
[2] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter-wave cellular wireless networks: Potentials and challenges,” Proc. IEEE, vol. 102, no. 3, pp. 366–385, Mar. 2014.
[3] F. Sohrabi and W. Yu, “Hybrid digital and analog beamforming design for large-scale antenna arrays,” IEEE J. Sel. Topics Signal Process., vol. 10, no. 3, pp. 501–513, Apr. 2016.
[4] X. Xia, K. Xu, S. Zhao, and Y. Wang, “Learning the time-varying massive MIMO channels: Robust estimation and data-aided prediction,” IEEE Trans. Veh. Technol., vol. 69, no. 8, pp. 8080–8096, Aug. 2020.
[5] D. Fan et al., “Angle domain channel estimation in hybrid millimeter wave massive MIMO systems,” IEEE Trans. Wireless Commun., vol. 17, no. 12, pp. 8165–8179, Dec. 2018.
[6] K. Venugopal, A. Alkhateeb, N. González-Prelcic, and R. W. Heath Jr., “Channel estimation for hybrid architecture-based wideband millimeter wave systems,” IEEE J. Sel. Areas Commun., vol. 35, no. 9, pp. 1996–2009, Sep. 2017.
[7] 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.
[8] S. Srivastava, A. Mishra, A. Rajorjaya, A. K. Jagannatham, and G. Ascheid, "Quasi-static and time-selective channel estimation for block-sparse millimeter wave hybrid MIMO systems: Sparse Bayesian learning (SBL) based approaches," IEEE Trans. Signal Process., vol. 67, no. 5, pp. 1251–1266, Mar. 2019.
[9] S. Srivastava, C. S. K. Patro, A. K. Jagannatham, and L. Hanzo, “Sparse, group-sparse, and online Bayesian learning aided channel estimation for doubly-selective mmWave hybrid MIMO OFDM systems,” IEEE Trans. Commun., vol. 69, no. 9, pp. 5843–5858, Sep. 2021.
[10] P. Wang, M. Pajovic, P. V. Orlik, T. Koike-Akino, K. J. Kim, and J. Fang, “Sparse channel estimation in millimeter wave communications: Exploiting joint AoD-AoA angular spread,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2017, pp. 1–6.

The running time can be dramatically decreased with dedicated hardware like FPGA or on chips in practice [4].
H. Ye, L. Liang, G. Y. Li, and B.-H. F. Juang, “Deep learning based channel estimation and adaptive spatial interlation,” IEEE Trans. Veh. Technol., vol. 68, no. 11, pp. 10663–10678, Nov. 2019.

X. Lin, S. Wu, C. Jiang, L. Kuang, J. Yan, and L. Hanzo, “Estimation of broadband multiuser millimeter wave massive MIMO-OFDM channels by exploiting their sparse structure,” IEEE Trans. Wireless Commun., vol. 17, no. 6, pp. 3959–3973, Jun. 2018.

B. Wang, M. Jian, F. Gao, G. Y. Li, and H. Lin, “Beam squint and channel estimation for wideband mmwave massive MIMO-OFDM systems,” IEEE Trans. Signal Process., vol. 67, no. 23, pp. 5983–5908, Dec. 2019.

A. Abdallah, A. Celik, M. M. Mansour, and A. M. Eltawil, “Deep learning-based frequency-selective channel estimation for hybrid mmWave MIMO systems,” IEEE Trans. Wireless Commun., vol. 21, no. 6, pp. 3808–3821, Jun. 2022.

J. Dai, A. Liu, and V. K. N. Lau, “FDD massive MIMO channel estimation with arbitrary 2D-array geometry,” IEEE Trans. Signal Process., vol. 66, no. 10, pp. 2584–2599, May 2018.

W. Ma, C. Qi, and G. Y. Li, “High-resolution channel estimation for frequency-selective mmWave massive MIMO systems,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3517–3529, May 2020.

J. Tan and L. Dai, “Wideband channel estimation for THz massive MIMO,” China Commun., vol. 18, no. 5, pp. 66–80, May 2021.

Z. Qin, H. Ye, G. Y. Li, and B. H. F. Juang, “Deep learning in physical layer communications,” IEEE Wireless Commun., vol. 26, no. 2, pp. 93–99, Mar. 2019.

C.-K. Wen, W.-T. Shih, and S. Jin, “Deep learning for massive MIMO CSI feedback,” IEEE Wireless Commun. Lett., vol. 7, no. 5, pp. 748–751, Oct. 2018.

H. Song, M. Zhang, J. Gao, and C. Zhong, “Unsupervised learning-based joint active and passive beamforming design for reconfigurable intelligent surfaces aided wireless networks,” IEEE Commun. Lett., vol. 25, no. 3, pp. 892–896, Mar. 2021.

H. Ye, G. Y. Li, and B.-H. F. 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.

H. Ye, L. Liang, G. Y. Li, and B.-H. F. Juang, “Deep learning based end-to-end wireless communication systems with GAN as unknown channel,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 3133–3143, May 2020.

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.

S. Liu, Z. Gao, J. Zhang, M. D. Renzo, and M.-S. Alouini, “Deep denoising neural network assisted compressive channel estimation for mmWave intelligent reflecting surfaces,” IEEE Trans. Veh. Technol., vol. 69, no. 8, pp. 9223–9228, Aug. 2020.

X. Ma and Z. Gao, “Data-driven deep learning to design pilot and channel estimator for massive MIMO,” IEEE Veh. Technol. Mag., vol. 69, no. 5, pp. 5677–5682, May 2020.

J. Gao, M. Hu, C. Zhong, G. Y. Li, and Z. Zhang, “An attention-aided deep learning framework for massive MIMO channel estimation,” IEEE Trans. Wireless Commun., vol. 21, no. 3, pp. 1823–1835, Mar. 2022.

O. Wang, J. Gao, and G. Y. Li, “Learn to adapt to new environments from past experience and few pilot blocks,” IEEE Trans. Cognit. Commun. Netw., early access, Dec. 13, 2022, doi: 10.1109/TCCN.2022.3228536.

M. Wenyan, 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, Jan. 2020.

J. 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.

H. He, R. Wang, W. Jin, S. Jin, C.-K. Wen, and G. Y. Li, “Beamspace channel estimation for wideband millimeter-wave MIMO: A model-driven unsupervised learning approach,” IEEE Trans. Wireless Commun., vol. 22, no. 3, pp. 1808–1822, Mar. 2023.

W. Jin, H. He, C.-K. Wen, S. Jin, and G. Y. Li, “Adaptive channel estimation based on model-driven deep learning for wideband mmWave systems,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2021, pp. 1–6.

X. Ma, Z. Gao, F. Gao, and M. D. Renzo, “Model-driven deep learning based channel estimation and feedback for millimeter-wave massive hybrid MIMO systems,” IEEE J. Sel. Areas Commun., vol. 39, no. 8, pp. 2388–2406, Aug. 2021.

Q. Hu, S. Shi, Y. Cai, and G. Yu, “DDPG-driven deep-unfolding with adaptive depth for channel estimation with sparse Bayesian learning,” IEEE Trans. Signal Process., vol. 70, pp. 4666–4680, 2022.

P. Wu and J. Cheng, “Deep unfolding basis pursuit: Improving sparse channel reconstruction via data-driven measurement matrices,” IEEE Trans. Wireless Commun., vol. 21, no. 10, pp. 8090–8150, Oct. 2022.

J. Gao, C. Zhong, G. Y. Li, and Z. Zhang, “Deep learning-based channel estimation for massive MIMO with hybrid transceivers,” IEEE Trans. Wireless Commun., vol. 21, no. 7, pp. 5162–5174, Jul. 2022.

F. Gao, B. Wang, C. Xing, J. An, and G. Y. Li, “Wideband beamforming for hybrid massive MIMO terahertz communications,” IEEE J. Sel. Areas Commun., vol. 39, no. 6, pp. 1725–1740, Jun. 2021.

D. P. Wipf and B. D. Rao, “Sparse Bayesian learning for basis selection,” IEEE Trans. Signal Process., vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

D. P. Wipf, “Sparse estimation with structured dictionaries,” in Proc. Adv. Neural Inf. Process. Syst., 2011, pp. 2016–2024.

M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” J. Mach. Learn. Res., vol. 1, pp. 211–244, Jun. 2001.

J. Fang, Y. Shen, H. Li, and P. Wang, “Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals,” IEEE Trans. Signal Process., vol. 63, no. 2, pp. 360–372, Jan. 2015.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” 2014, arXiv:1412.0767.

R. J. Peter and C. R. Murthy, “Learned-SBL: A deep learning architecture for sparse signal recovery,” 2019, arXiv:1909.08185.

S. Schubert, P. Neubert, J. Poschmann, and P. Protzel, “Circu- lar convolutional neural networks for panoramic images and laser data,” in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2019, pp. 653–660.

R. Liu et al., “An intriguing failing of convolutional neural networks and theCoordConv solution,” 2018, arXiv:1807.03247.

X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” J. Mach. Learn. Res., vol. 9, pp. 249–258, May 2010.

J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, “Wireless image transmission using deep source channel coding with attention modules,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 4, pp. 2315–2328, Apr. 2022.

J. He, T. Kim, H. Ghauch, K. Liu, and G. Wang, “Millimeter wave MIMO channel tracking systems,” in Proc. IEEE Globecom Workshops (GC Wkshps), Dec. 2014, pp. 416–421.
Caijun Zhong (Senior Member, IEEE) received the B.S. degree in information engineering from Xi’an Jiaotong University, Xi’an, China, in 2004, and the M.S. degree in information security and the Ph.D. degree in telecommunications from University College London, London, U.K., in 2006 and 2010, respectively. From September 2009 to September 2011, he was a Research Fellow with the Institute for Electronics, Communications and Information Technology (ECTT), Queens University Belfast, Belfast, U.K. Since September 2011, he has been with Zhejiang University, Hangzhou, China, where he is currently a Professor. His research interests include reconfigurable intelligent surfaces-assisted communications and artificial intelligence based on wireless communications. He was an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE COMMUNICATIONS LETTERS. He is an Editor of Science China Information Science and China Communications. He was a recipient of the 2013 IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award. He and his coauthors was a recipient of the Best Paper Award from the IEEE GLOBECOM 2020, IEEE ICC 2019, and IEEE ICC’2022.

Geoffrey Ye Li (Fellow, IEEE) was a Professor at the Georgia Institute of Technology, Atlanta, GA, USA, for 20 years, and a Principal Technical Staff Member at AT&T Laboratories—Research, Middletown, NJ, USA, for five years. He is currently a Chair Professor with Imperial College London, U.K. He has published over 600 journals and conference papers in addition to over 40 granted patents and several books in his research areas. His publications have been cited over 50,000 times with an H-index over 100. He has been recognized as a Highly Cited Researcher, by Thomson Reuters, almost every year. His general research interests include statistical signal processing and machine learning for wireless communications. He was a recipient of several prestigious awards, including the Donald G. Fink Overview Paper Award from the IEEE Signal Processing Society in 2017; the James Evans Avant Garde Award in 2013 and the Jack Neubauer Memorial Award in 2014 from the IEEE Vehicular Technology Society; and the Stephen O. Rice Prize Paper Award in 2013, the Award for Advances in Communication in 2017, and the Edwin Howard Armstrong Achievement Award in 2019 from the IEEE Communications Society. He also received the 2015 Distinguished ECE Faculty Achievement Award from Georgia Tech. He has organized and chaired many international conferences, including the Technical Program Vice-Chair of the IEEE ICC’03 and the General Co-Chair of the IEEE GlobalSIP’14, the IEEE VTC’19, Fall, and the IEEE SPAWC’20. He has been involved in editorial activities for over 20 technical journals, including the Founding Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Special Series on ML in Communications and Networking.

Joseph B. Soriaña received the B.S. degree in electrical engineering from the California Institute of Technology (Caltech) and the Ph.D. degree in electrical and computer engineering from the University of California San Diego (UCSD) with a focus on coding theory, communications, and information theory. He is currently the Senior Director of Technology, and lead the Systems Team, Corporate Research and Development Department, Qualcomm AI Research, Qualcomm Technologies Inc. The team is broadly responsible for driving hardware and software platform innovations, advancing applications in areas, such as multimedia and wireless, and contributing to fundamental research in machine learning. Prior to this role, he led the 5G Core Physical Layer Research Team, Qualcomm Wireless Research and Development, and helped drive the standardization of 5G as both a 3GPP RAN1 delegate and a technical specifications editor of TS 38.202. He has also had previous experience as the Systems Team Lead for Qualcomm’s 3G EV-DO commercial cell-site modem (CSM). He has been granted over 280 U.S. Patents. He was awarded the IEEE Data Storage Best Paper in 2007 and the EURASIP Journal on Wireless Communications and Networking Best Paper in 2012.

Arash Behboodi (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2005 and 2007, respectively, the master’s degree in philosophy from Pantheon-Sorbonne University, Paris, France, in 2011, and the Ph.D. degree from Ecole superieure d’électricité (now Centrale-Supélec), Gif-sur-Yvette, France, in 2012. He was a Senior Researcher with the Technical University of Berlin from 2013 to 2015 and RWTH Aachen University from 2015 to 2019. He is currently a Machine Learning Scientist and a Principal Engineer with Qualcomm AI Research, Qualcomm Technologies Netherlands B.V. His research interests include machine learning design for wireless communication, learning theory, machine learning for inverse problems, information theory, compressed sensing, and mathematical signal processing.