2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring) | 979-8-3503-1114-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Spring57618.2023.10200898

Block Sparse Channel Estimation based on Residual Difference and Deep Learning for Wideband MmWave Massive MIMO

Rongshun Tang, Chenhao Qi and Pengju Zhang
School of Information Science and Engineering, Southeast University, Nanjing, China
National Mobile Communications Research Laboratory, Southeast University, Nanjing, China
Email: qch@seu.edu.cn

***Abstract*—Time-domain channel estimation for wideband millimeter wave (mmWave) MIMO OFDM systems is considered. To mitigate the overfitting of the existing time-domain channel estimation exploiting block sparsity (TDCEBS) scheme, we propose a block sparse channel estimation exploiting residual difference (BSCERD) scheme, where we first compute the difference of the residual power for every two adjacent iterations, and then determine a threshold to indicate the convergence of the iterations. Moreover, to improve the global optimality and reduce the time overhead of compressive sensing, a block sparse channel estimation based on deep learning (BSCEDL) scheme is proposed to determine the indices of the nonzero blocks simultaneously. We exploit the QuaDRiGa to assess the efficacy of the schemes proposed. Simulation results show that both BSCERD and BSCEDL outperform TDCEBS, while BSCEDL is better than BSCERD in performance and can achieve much lower time overhead.**

***Index Terms*—Block sparse, channel estimation, deep learning, mmWave communications, sparse recovery**

I. Introduction

In order to fulfill high communication requirement of 5G and beyond, millimeter wave (mmWave) communication has received extensive attention due to its abundant spectral resources. Different from the early research on narrow-band mmWave channels [1]–[3], recent studies exhibit the wideband characteristics of mmWave channels in practice and orthogonal frequency-division multiplexing (OFDM) has been introduced to mmWave massive MIMO communication [4]–[6].

Channel estimate is required to get channel state information for mmWave massive MIMO [7]. Numerous studies have shown that mmWave channels are essentially sparse, and for the delay sparsity of wideband mmWave channels, time-domain channel estimation has been widely investigated. Utilizing the channel delay sparsity, [6] proposes a sparse Bayesian learning (SBL) based block sparse channel estimation scheme. In [8], based on the least squared (LS) method and the orthogonal matching pursuit (OMP), a time-domain channel estimation method with two steps is proposed. Based on the block sparsity of different spatial directions, [9] purposes a time-domain channel estimation exploiting block sparsity (TDCEBS) scheme and further improves channel estimation performance.

In practice, since the channel sparsity is unknown, sparse channel estimation methods may stop the iterations earlier or later than reaching the actual sparsity when iteratively performing the sparse recovery. In the former case, the time-domain channel is not fully recovered, which results in the underfitting of the channel estimation. In the latter case, the overfitting of channel estimation will probably happen, due to the severe noise effect [10]. In either case, there is some loss in sparse channel estimation performance, which calls for accurate estimation of the channel sparsity. Note that the TDCEBS scheme sets the termination condition of the iterations as reaching a predefined number probably a bit larger than the actual sparsity, which may result in the overfitting of the channel estimation.

Moreover, sparse channel estimation schemes based on compressive sensing (CS) estimate the time-domain channel in a sequential and greedy manner, which cannot ensure the global optimality [3]. In addition, compressed sensing algorithms require iterative calculations, which have high time overhead and may be difficult to adapt to large-scale real-time communication systems. Deep learning can typically train the neural network (NN) based on channel data before starting the channel estimation, which only needs a low time overhead by deploying the well-trained NN for channel prediction. The above motivates us to estimate the valid channel entries simultaneously rather than sequentially by deep learning.

The contribution of this paper is mainly summarized as follows.

1) To mitigate the overfitting of the TDCEBS scheme, we propose a block sparse channel estimation exploiting residual difference (BSCERD) scheme. For the BSCERD scheme, we first compute the difference of the residual power for every two adjacent iterations, and then determine a threshold to indicate the convergence of the iterations.
2) To avoid the greedy search and reduce the time overhead of CS algorithm, we propose a block sparse channel estimation based on deep learning (BSCEDL), where the indices of all nonzero blocks are estimated simultaneously by the deep learning.

Notations: $a, \boldsymbol{a}, \boldsymbol{A}$ denote a scalar, a vector, and a matrix, respectively, while $(*)^T, (*)^H, \|*\|_2, \|*\|_F$ signify the transpose, the conjugate transpose, the ℓ_2 norm, and the F-norm, respectively. The mth element of the vector $\boldsymbol{a}$ is represented as $[\boldsymbol{a}]_m$. $[\boldsymbol{A}]_{m,n}$ signifies the item in the mth row and nth column of the matrix $\boldsymbol{A}$, and $\boldsymbol{I}_K$ denotes an identity matrix with K-by-K dimensions. $\mathbb{R}$, $\mathbb{C}$ and $\emptyset$ represent the set of real-valued numbers, the set of complex-valued numbers and the empty set, respectively. $\cup$ and $\setminus$ denotes the union of sets and operation of set exclusion.

II. System Model

As illustrated in Fig. 1, a downlink wideband mmWave massive MIMO OFDM system is taken into consideration. The base station (BS) is outfitted with a uniform linear array (ULA) of N_{BS} antennas, while the user is given access to a single antenna. The duration of each OFDM sample is

$$T_{\mathrm{s}} = \frac{1}{N_{\mathrm{c}} \Delta f}, \tag{1}$$

where N_{c} is the number of OFDM subcarriers, Δf is the subcarriers spacing. The length of the cyclic prefix (CP) placed at the head of each OFDM symbol to eliminate inter-carrier interference is denoted by N_{cp}.

A line-of-sight (LoS) scenario is considered for the system, which includes a LoS path and $L-1$ non-line-of-sight (NLoS) paths. According to the extended Saleh-Valenzuela channel model, the wideband mmWave MIMO channel can be expressed as

$$\boldsymbol{h}(t) = \sqrt{\frac{N_{\mathrm{BS}}}{L}} \sum_{l=1}^{L} \gamma_l p(t-\tau_l) \boldsymbol{\alpha}^H(N_{\mathrm{BS}}, \phi_l), \tag{2}$$

where ϕ_l, τ_l and γ_l, for $l = 1, 2, \ldots, L$ denote the angle-of-departure (AoD), delay and the channel gain of each channel path respectively, $p(t)$ is the pulse shaping function with duration T_{s}, and $\boldsymbol{\alpha}(N_{\mathrm{BS}}, \phi_l)$ is expressed as

$$\boldsymbol{\alpha}(N_{\mathrm{BS}}, \phi_l) = \frac{1}{\sqrt{N_{\mathrm{BS}}}} [1, e^{j\pi\phi_l}, \ldots, e^{j(N_{\mathrm{BS}}-1)\pi\phi_l}]^T. \tag{3}$$

For convenience, we set the maximum channel time extension equal to the length of CP. After channel sampling with N_{cp} taps, (2) can be written as

$$\boldsymbol{h}(n) = \sqrt{\frac{N_{\mathrm{BS}}}{L}} \sum_{l=1}^{L} \gamma_l p(nT_{\mathrm{s}} - \tau_l) \boldsymbol{\alpha}^H(N_{\mathrm{BS}}, \phi_l), \tag{4}$$

for $n = 1, 2 \ldots, N_{\mathrm{cp}}$. The wideband channels are stacked together to obtain the time-domain channel matrix $\boldsymbol{H} \in \mathbb{C}^{N_{\mathrm{cp}} \times N_{\mathrm{BS}}}$, expressed as

$$\boldsymbol{H} \triangleq \left[\boldsymbol{h}(1)^T, \boldsymbol{h}(2)^T, \ldots, \boldsymbol{h}(N_{\mathrm{cp}})^T\right]^T. \tag{5}$$

Frequency-domain pilot symbols of length $K (K \leq N_{\mathrm{c}})$ are transmitted in order to estimate $\boldsymbol{H}$, where $x(k)$, $k = 1, 2, \ldots, K$ denotes the kth transmitted pilot symbol and $y(k)$, $k = 1, 2, \ldots, K$ stands for the corresponding received pilot symbol. Then the received pilot vector can be defined as

$$\boldsymbol{y} \triangleq [y(1), y(2), \ldots, y(K)]^T \in \mathbb{C}^K, \tag{6}$$

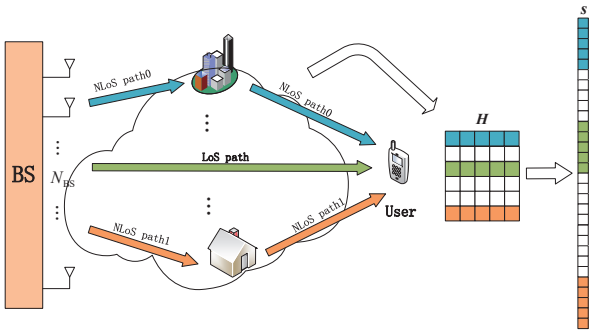

Fig. 1. Wideband mmWave MIMO system with block-sparse channels.

and we have

$$\boldsymbol{y} = \boldsymbol{XDHf} + \boldsymbol{\eta}, \tag{7}$$

where $\boldsymbol{X} = \mathrm{diag}\{x(1), x(2), \ldots, x(K)\} \in \mathbb{C}^{K \times K}$ is a diagonal matrix, and $\boldsymbol{D} \in \mathbb{C}^{K \times N_{\mathrm{cp}}}$ is constructed by extracting the first N_{cp} columns and picking the K rows that correspond to the pilot subcarriers from the standard discrete Fourier transform (DFT) matrix $\boldsymbol{G} \in \mathbb{C}^{N_{\mathrm{c}} \times N_{\mathrm{c}}}$. $\boldsymbol{f} \in \mathbb{C}^{N_{\mathrm{BS}}}$ is a beamforming vector and $\boldsymbol{\eta} \in \mathbb{C}^K$ represents an additive white Gaussian noise (AWGN) vector with $\boldsymbol{\eta} \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_K)$.

The N_{BS} distinct beamforming vectors are used to scan the whole space, and make up a codebook matrix

$$\boldsymbol{F} \triangleq [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_{N_{\mathrm{BS}}}] \in \mathbb{C}^{N_{\mathrm{BS}} \times N_{\mathrm{BS}}}. \tag{8}$$

Based on (7) and (8), we have

$$\boldsymbol{Y} = \boldsymbol{XDHF} + \boldsymbol{\Psi}, \tag{9}$$

where $\boldsymbol{Y} \triangleq [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{N_{\mathrm{BS}}}] \in \mathbb{C}^{K \times N_{\mathrm{BS}}}$ is the received pilot matrix, and $\boldsymbol{\Psi} \triangleq [\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_{N_{\mathrm{BS}}}] \in \mathbb{C}^{K \times N_{\mathrm{BS}}}$ is the AWGN matrix.

To estimate $\boldsymbol{H}$, (9) is firstly multiplied on both sides by the right pseudo-inverse of $\boldsymbol{F}$, expressed as

$$\boldsymbol{YF}^H(\boldsymbol{FF}^H)^{-1} = \boldsymbol{XDH} + \boldsymbol{\Psi F}^H(\boldsymbol{FF}^H)^{-1}. \tag{10}$$

For simplicity, we rewrite (10) as

$$\boldsymbol{Z} = \boldsymbol{AH} + \boldsymbol{N}, \tag{11}$$

where,

$$\boldsymbol{A} \triangleq \boldsymbol{XD} \in \mathbb{C}^{K \times N_{\mathrm{cp}}}, \tag{12}$$

$$\boldsymbol{Z} \triangleq \boldsymbol{YF}^H(\boldsymbol{FF}^H)^{-1} \in \mathbb{C}^{K \times N_{\mathrm{BS}}}, \tag{13}$$

$$\boldsymbol{N} \triangleq \boldsymbol{\Psi F}^H(\boldsymbol{FF}^H)^{-1} \in \mathbb{C}^{K \times N_{\mathrm{BS}}}. \tag{14}$$

Then, $\boldsymbol{H}$ can be estimated by the LS method, expressed as

$$\widehat{\boldsymbol{H}}_{LS} = (\boldsymbol{A}^H \boldsymbol{A})^{-1} \boldsymbol{A}^H \boldsymbol{Z}. \tag{15}$$

It is worth noting that the foundation of (15) is that $\boldsymbol{A}$ is full rank of column. We cannot ensure that $\boldsymbol{A}$ is always full rank of column even if $K \geq N_{\mathrm{cp}}$ because the rank of $\boldsymbol{A}$

is fundamentally decided by $\boldsymbol{D}$ and $\boldsymbol{G}$. $\boldsymbol{A}$ may be a matrix with low rank of column if we decrease the quantity of pilot symbols to improve spectral efficiency, which results in $K < N_{\mathrm{cp}}$. So we cannot use (15) to estimate $\boldsymbol{H}$ directly.

In practice, because of the typically sparsity of mmWave channels, there are far less channel paths than the maximum channel time extension i.e., $L \ll N_{\mathrm{cp}}$, causing the majority of the rows in $\boldsymbol{H}$ are zero. As shown in Fig. 1, only some rows of $\boldsymbol{H}$ storing channel paths are nonzero and columns share the same sparsity. The time-domain channel matrix exhibits block-sparse features of rows.

Consistent with [9], to take use of the block sparsity of $\boldsymbol{H}$, we vectorize $\boldsymbol{H}$,$\boldsymbol{Z}$ and $\boldsymbol{N}$ by stringing each row together as follows

$$\boldsymbol{s} \triangleq \left[\boldsymbol{h}(1), \boldsymbol{h}(2), \ldots, \boldsymbol{h}(N_{\mathrm{cp}})\right]^T \in \mathbb{C}^{N_{\mathrm{cp}} N_{\mathrm{BS}}}, \tag{16}$$

$$\boldsymbol{q} \triangleq [\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_K]^T \in \mathbb{C}^{K N_{\mathrm{BS}}}, \tag{17}$$

$$\boldsymbol{v} \triangleq [\boldsymbol{n}_1, \boldsymbol{n}_2, \ldots, \boldsymbol{n}_K]^T \in \mathbb{C}^{K N_{\mathrm{BS}}}, \tag{18}$$

where $\boldsymbol{z}_k$ and $\boldsymbol{n}_k$ represent the kth row of $\boldsymbol{Z}$ and $\boldsymbol{N}$.

Then we define a stacked measurement matrix

$$\boldsymbol{B} \triangleq \begin{pmatrix} \boldsymbol{B}_{1,1} & \boldsymbol{B}_{1,2} & \cdots & \boldsymbol{B}_{1,N_{\mathrm{cp}}} \\ \boldsymbol{B}_{2,1} & \boldsymbol{B}_{2,2} & \cdots & \boldsymbol{B}_{2,N_{\mathrm{cp}}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{B}_{K,1} & \boldsymbol{B}_{K,2} & \cdots & \boldsymbol{B}_{K,N_{\mathrm{cp}}} \end{pmatrix} \in \mathbb{C}^{K N_{\mathrm{BS}} \times N_{\mathrm{cp}} N_{\mathrm{BS}}}, \tag{19}$$

where $\boldsymbol{B}_{k,m}$ is the (k,m)th block of $\boldsymbol{B}$, denoted as

$$\boldsymbol{B}_{k,m} \triangleq [\boldsymbol{A}]_{k,m} \boldsymbol{I}_{N_{\mathrm{BS}}} \in \mathbb{C}^{N_{\mathrm{BS}} \times N_{\mathrm{BS}}}, \tag{20}$$

for $k = 1, 2, \ldots, K$ and $m = 1, 2, \ldots, N_{\mathrm{cp}}$.

Finally, (11) can be rewritten as

$$\boldsymbol{q} = \boldsymbol{B}\boldsymbol{s} + \boldsymbol{v}. \tag{21}$$

We transform the row-sparse features of $\boldsymbol{H}$ into block-sparse features of $\boldsymbol{s}$, which can be estimated by the block sparse compressive recovery algorithms.

III. BSCERD Time-domain Channel Estimation

It is worth noting that the channel delays of different paths are generally different, but it is likely that there are some paths with close channel delays. Due to the limited resolution of tap sampling, the paths with close delays are collected in the same delay tap and superimposed in the same row of $\boldsymbol{H}$. This shows that the number of non-zero rows of $\boldsymbol{H}$ is not necessarily equal to L, but usually less than L and changes dynamically. Therefore, it is necessary to design an effective termination condition for the block sparse channel estimation of $\boldsymbol{H}$. TDCEBS only sets a predefined number $N = L$ of iterations or a fixed threshold as the termination condition, which may lead to overfitting.

Different from TDCEBS, we find the stability of the difference of the residual power for two adjacent iterations, and accurately estimate the channel sparsity by computing an adaptive residual difference thresholds.

Firstly, a residual vector is defined as $\boldsymbol{r} \in \mathbb{C}^{N_{\mathrm{q}}}$, and initialized by $\boldsymbol{r} \leftarrow \boldsymbol{q}$. Then, the iteration count m, and the maximum iteration number $M > L+1$ are defined to control the entire iterative process.

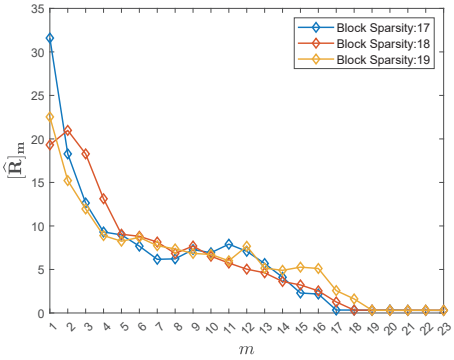

Fig. 2. Convergence of $\widehat{\boldsymbol{R}}$ with increasing iterations.

At each iteration, the optimal non-zero block index of $\boldsymbol{H}$ can be obtained [9] by

$$I = \arg \max_{i \in \{1,2,\ldots,N_{\mathrm{cp}}\} \backslash \boldsymbol{\Gamma}} \left\| \boldsymbol{P}_i^H \boldsymbol{r} \right\|_2, \tag{22}$$

where $\boldsymbol{P}_i \in \mathbb{C}^{K N_{\mathrm{BS}} \times N_{\mathrm{BS}}}$ is the ith column block of $\boldsymbol{B}$ defined as

$$\boldsymbol{P}_i \triangleq \left[\boldsymbol{B}_{1,i}^T, \boldsymbol{B}_{2,i}^T, \ldots, \boldsymbol{B}_{K,i}^T\right]^T. \tag{23}$$

We define a vector $\boldsymbol{\Gamma} \in \mathbb{R}^M$ to store I picked out at each iteration. For the m-th iteration, $[\boldsymbol{\Gamma}]_m$ can be denoted as

$$[\boldsymbol{\Gamma}]_m \leftarrow I. \tag{24}$$

Then the residue vector is updated as

$$\boldsymbol{r} \leftarrow \boldsymbol{q} - \boldsymbol{P}_{\boldsymbol{\Gamma}} (\boldsymbol{P}_{\boldsymbol{\Gamma}}^H \boldsymbol{P}_{\boldsymbol{\Gamma}})^{-1} \boldsymbol{P}_{\boldsymbol{\Gamma}}^H \boldsymbol{q}, \tag{25}$$

where

$$\boldsymbol{P}_{\boldsymbol{\Gamma}} \triangleq \{\boldsymbol{P}_i,\ i \in \boldsymbol{\Gamma}\}. \tag{26}$$

We define a vector $\boldsymbol{R} \in \mathbb{R}^{M+1}$ to store the power $\|\boldsymbol{r}\|_2$ at each iteration, where $[\boldsymbol{R}]_0$ is initialized as $[\boldsymbol{R}]_0 \leftarrow \|\boldsymbol{q}\|_2$ when the iteration is not started, and $[\boldsymbol{R}]_m$ at the mth iteration can be expressed as

$$[\boldsymbol{R}]_m \leftarrow \|\boldsymbol{r}\|_2. \tag{27}$$

In addition, we define a vector $\widehat{\boldsymbol{R}} \in \mathbb{R}^M$ to store the difference of the residual power for two adjacent iterations. $[\widehat{\boldsymbol{R}}]_m$ at the mth iteration can be expressed as

$$[\widehat{\boldsymbol{R}}]_m \leftarrow [\boldsymbol{R}]_{m-1} - [\boldsymbol{R}]_m. \tag{28}$$

We iteratively run (22), (24), (25), (27) and (28) until the iteration number is equal to the maximum iteration number, i.e., $m = M$.

Since the block sparsity of $\boldsymbol{H}$ is unknown, after all M iterations, $\boldsymbol{\Gamma}$ stores redundant indices, so $\boldsymbol{\Gamma}$ need to be further filtered. We consider filtering $\boldsymbol{\Gamma}$ exploiting the residual difference vector $\widehat{\boldsymbol{R}}$. As shown in Fig. 2, in the iterative process, as the nonzero blocks of $\boldsymbol{H}$ are projected according to the primary and secondary, $\widehat{\boldsymbol{R}}$ reduces and converges with increasing iterations. When the current iteration number m is equal to the actual block sparsity, which means that all of the

Algorithm 1 Block Sparse Channel Estimation Exploiting Residual Differential (BSCERD)

Input: $\boldsymbol{A}$, $\boldsymbol{Z}$, σ, M.
Output: $\widehat{\boldsymbol{H}}_{\boldsymbol{\Omega}}$.

Initialization: $m \leftarrow 1$, $\boldsymbol{r} \leftarrow \boldsymbol{q}$, $\boldsymbol{\Gamma} \leftarrow \mathbf{0}^M$, $\boldsymbol{R} \leftarrow \mathbf{0}^{M+1}$, $\widehat{\boldsymbol{R}} \leftarrow \mathbf{0}^M$, $\boldsymbol{\Omega} \leftarrow \emptyset$.
$[\boldsymbol{R}]_0 \leftarrow \|\boldsymbol{q}\|_2$.
while $m \leq M$ **do**
  Obtain the optimal nonzero block index via (22).
  Update the index vector $\boldsymbol{\Gamma}$ via (24).
  Update the residual vector $\boldsymbol{r}$ via (25).
  Update the residual power vector $\boldsymbol{R}$ via (27).
  Update the residual difference vector $\widehat{\boldsymbol{R}}$ via (28).
  $m \leftarrow m + 1$.
end while
for $i = 1$ to $M - 1$ **do**
  if $[\widehat{\boldsymbol{R}}]_i \geq [\widehat{\boldsymbol{R}}]_M + \sigma$ **then**
    $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} \cup \{[\boldsymbol{\Gamma}]_m\}$
  end if
end for
Estimate the nonzero rows of $\boldsymbol{H}$ via (31).

nonzero blocks have been projected, the residual difference $[\widehat{\boldsymbol{R}}]_{m+1}, \ldots, [\widehat{\boldsymbol{R}}]_M$ will fluctuate slightly and then converge. By setting the residual difference threshold, the actual block sparsity of the channel can be accurately estimated to filter $\boldsymbol{\Gamma}$.

We define the set $\boldsymbol{\Omega}$ to store the filtered results, and initialize $\boldsymbol{\Omega} \leftarrow \emptyset$. When all M iterations are over, set $[\widehat{\boldsymbol{R}}]_M + \sigma$ as the adaptive filtering threshold, where σ is used to eliminate jitter. If $[\widehat{\boldsymbol{R}}]_i$ is greater than $[\widehat{\boldsymbol{R}}]_M + \sigma$, $[\boldsymbol{\Gamma}]_i$ is stored in $\boldsymbol{\Omega}$, which can be expressed as

$$\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} \cup \{[\boldsymbol{\Gamma}]_i\}, \ i = 1, 2, \ldots, M-1, \tag{29a}$$

$$\text{s.t.} \ \ [\widehat{\boldsymbol{R}}]_i > [\widehat{\boldsymbol{R}}]_M + \sigma. \tag{29b}$$

Then the columns from $\boldsymbol{A}$ corresponding to the indices in $\boldsymbol{\Omega}$ are picked out to form a submatrix

$$\boldsymbol{A}_{\boldsymbol{\Omega}} \triangleq \{\boldsymbol{A}_i, \ i \in \boldsymbol{\Omega}\} \in \mathbb{C}^{K \times J}. \tag{30}$$

Finally, LS estimation method is adopted to calculate the J nonzero rows $\boldsymbol{H}_{\boldsymbol{\Omega}}$ of $\boldsymbol{H}$.

$$\widehat{\boldsymbol{H}}_{\boldsymbol{\Omega}} = (\boldsymbol{A}_{\boldsymbol{\Omega}}^H \boldsymbol{A}_{\boldsymbol{\Omega}})^{-1} \boldsymbol{A}_{\boldsymbol{\Omega}}^H \boldsymbol{Z}. \tag{31}$$

The overview of the specific steps of the BSCERD channel estimation scheme is shown in **Algorithm 1**

IV. BSCEDL Time-domain Channel Estimation

The BSCERD and TDCEBS schemes take a sequential and greedy manner to find the optimal nonzero block indexes, which cannot guarantee the global optimality and lead to high time overhead.

In this section, we propose a block sparse channel estimation based on deep learning (BSCEDL) scheme to efficiently estimate the valid of all blocks of $\boldsymbol{s}$ in parallel. The scheme mainly includes time-domain channel nonzero block indices estimation and channel reconstruction. The block diagram of BSCEDL is shown in Fig. 3.

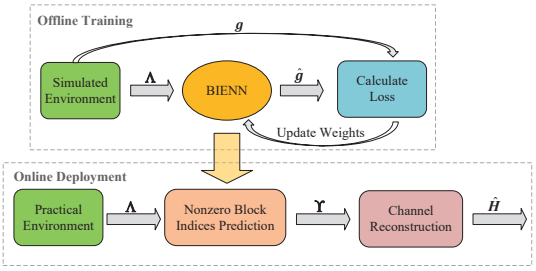

Fig. 3. Block diagram of BSCEDL: offline training and online deployment.

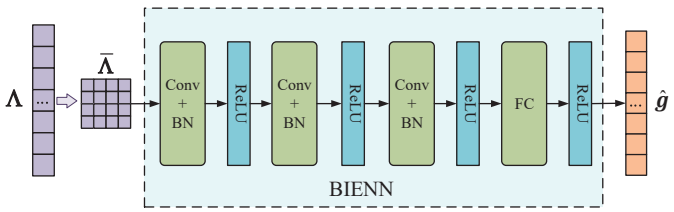

Fig. 4. Illustration of the BIENN.

A. Nonzero Block Indices Estimation

We first define $\boldsymbol{\Lambda} \in \mathbb{R}^{N_{\text{cp}}}$ to store the projected value of each column block in $\boldsymbol{B}$ on $\boldsymbol{q}$, where the ith element of $\boldsymbol{\Lambda}$ is expressed as

$$[\boldsymbol{\Lambda}]_i = \left\| \boldsymbol{P}_i^H \boldsymbol{q} \right\|_2. \tag{32}$$

As shown in Fig. 3, the time-domain channel nonzero block indices estimation is made up of two stages: the offline training of the nonzero block indices estimation neural network (BIENN) and its online deployment. After BIENN has been trained offline, the time-domain channel sparse block gain estimate uses the BIENN as the kernel. As shown in Fig. 4, the input of the BIENN is $\bar{\boldsymbol{\Lambda}}$, which is obtained by reshaping $\boldsymbol{\Lambda}$ into the size of $\sqrt{N_{\text{cp}}} \times \sqrt{N_{\text{cp}}}$.

The ultimate goal of BIENN is to find the indices of nonzero blocks in $\boldsymbol{H}$. We define $\boldsymbol{g} \in \mathbb{R}^{N_{\text{cp}}}$ to store the label of each block in $\boldsymbol{H}$, where the label of the ith block in $\boldsymbol{H}$ can be expressed as

$$[\boldsymbol{g}]_i = \begin{cases} 1, & \|\boldsymbol{h}(i)\|_2 \neq 0 \\ 0, & \|\boldsymbol{h}(i)\|_2 = 0 \end{cases}, \ i = 1, 2, \ldots, N_{cp}. \tag{33}$$

Therefore, we use $\boldsymbol{g}$ as the training label of BIENN, and BIENN outputs the prediction $\widehat{\boldsymbol{g}}$ of $\boldsymbol{g}$ by learning the characteristics of $\bar{\boldsymbol{\Lambda}}$. The optimization goal of the whole training process is to make $\widehat{\boldsymbol{g}}$ as close to $\boldsymbol{g}$ as possible, so the loss function trained for BIENN is defined as the mean square error function, expressed as

$$f_{\text{loss}}(\boldsymbol{g}, \widehat{\boldsymbol{g}}) = \frac{1}{N_{\text{cp}}} \sum_{n=1}^{N_{\text{cp}}} ([\boldsymbol{g}]_n - [\widehat{\boldsymbol{g}}]_n)^2. \tag{34}$$

As illustrated in Fig. 4, three hidden layers and an output layer make up the BIENN model used in the BSCEDL scheme.

Algorithm 2 Block Sparse Channel Estimation based on Deep Learning (BSCEDL)

Input: $\boldsymbol{A}$, $\boldsymbol{Z}$, $\boldsymbol{q}$, ϵ.
Output: $\widehat{\boldsymbol{H}}_{\boldsymbol{\Upsilon}}$.

```
Initialization: Υ ← ∅.
Obtain Λ and Λ̄ via (32).
Input Λ̄ to the offline-trained BIENN to get ĝ.
for i = 1 to N_cp do
    if [ĝ]_i > ε then
        Υ ← Υ ∪ {i}
    end if
end for
Estimate the nonzero rows of H via (36).
```

Each hidden layer is made up of a convolutional (Conv) layer and a batch normalization (BN) layer. These three Conv layers each include 8, 16, and 32 filters, respectively, and the size and stride of filter in each Conv layer are set to be 3×3 and 1, respectively. In addition, the ReLU function, which can be expressed as $f_{\mathrm{Re}} = \max(0, x)$, is the activation function used in each layer. The input of BIENN is $\bar{\boldsymbol{\Lambda}} \in \mathbb{R}^{\sqrt{N_{\mathrm{cp}}} \times \sqrt{N_{\mathrm{cp}}}}$, the output is $\widehat{\boldsymbol{g}}$.

In the simulation environment in Fig. 3, we generate a dataset with 2×10^4 samples exploiting QuaDRiGa whose parameters are described in **Section** V and divide it into training set and verification set in a ratio of $8:2$ for offline training of BIENN. The adaptive moment estimation (Adam) optimizer is adopted to train BIENN by the deep learning toolbox of Matlab. The learning rate is set to an exponential decay function, where the initial value of the learning rate is 0.001 decaying to 80% after every 10 training epochs.

During the online deployment stage of the BIENN, the received signal $\boldsymbol{Y}$ is obtained from the actual environment. Since $\boldsymbol{F}$, $\boldsymbol{X}$ and $\boldsymbol{D}$ are known to the base station, we can obtain $\boldsymbol{A}$ and $\boldsymbol{Z}$ based on (12) and (13). Then we construct $\boldsymbol{q}$ and $\boldsymbol{B}$ based on (17) and (19). Finally, we get $\bar{\boldsymbol{\Lambda}}$ based on (32) and feed it into BIENN to get the prediction $\widehat{\boldsymbol{g}}$ of $\boldsymbol{g}$.

B. Channel Reconstruction

We set a threshold of $\epsilon = 0.5$ to determined the validity of each time domain channel block with binary judgment. The estimated sparsity J is obtained by calculating the number of elements in $\widehat{\boldsymbol{g}}$ greater than ϵ, and store the index of the corresponding elements in $\boldsymbol{\Upsilon} \in \mathbb{R}^J$, expressed as

$$\boldsymbol{\Upsilon} \leftarrow \boldsymbol{\Upsilon} \cup \{i\}, \ i = 1, 2, \ldots, N_{cp}, \tag{35a}$$

$$\text{s.t.} \ \ [\widehat{\boldsymbol{g}}]_i > \epsilon. \tag{35b}$$

Finally, the LS estimation method is adopted to calculate the J nonzero rows $\boldsymbol{H}_{\boldsymbol{\Upsilon}}$ of $\boldsymbol{H}$.

$$\widehat{\boldsymbol{H}}_{\boldsymbol{\Upsilon}} = (\boldsymbol{A}_{\boldsymbol{\Upsilon}}^H \boldsymbol{A}_{\boldsymbol{\Upsilon}})^{-1} \boldsymbol{A}_{\boldsymbol{\Upsilon}}^H \boldsymbol{Z}. \tag{36}$$

The overview of the specific steps of the BSCEDL channel estimation scheme is shown in **Algorithm 2**

V. Simulation Results

We use the QuaDRiGa [11] to carry on the validation of performance for the wideband mmWave communication system, where a BS equipped with $N_{\mathrm{BS}} = 64$ antennas serves a single-antenna user. According to the 5G new radio (NR) high-frequency standard, specific simulation parameters for OFDM modulation are listed in Table I.

TABLE I
Parameters of QuaDRiGa.

Parameter	value
Number of BS antennas	$N_{\mathrm{BS}} = 64$
Number of subcarriers	$N_{\mathrm{c}} = 2048$
length of CP	$N_{\mathrm{cp}} = 144$
Number of paths	$L = 21$
Center frequency	28GHz
Subcarrier spacing	$\Delta f = 120$KHz
Distance between UE and BS	$25 - 250$m
Height of BS	25m
Height of UE	1.5m

The 132 resource blocks (RBs) that take up 1584 OFDM subcarriers around the center frequency are utilized for signal transmission in accordance with the 5G NR standard, while the remaining OFDM subcarriers at the two band edges are employed as blank subcarriers. Each RB transmits one of the $K = 132$ pilot symbols.

The proposed BSCERD and BSCEDL schemes are compared with existing TDCEBS [9], SBL [6] and OMP schemes. We set the maximum number of iterations $N = L = 21$ for TDCEBS and OMP, $M = 23$ and $\sigma = 0.1$ for BSCERD, and $\epsilon = 0.5$ for BSCEDL.

As shown in Fig. 5, we compare the channel estimation performance in terms of the normalized mean squared error (NMSE), which is expressed as

$$\mathrm{NMSE} = \frac{\|\widehat{\boldsymbol{H}} - \boldsymbol{H}\|_F^2}{\|\boldsymbol{H}\|_F^2}. \tag{37}$$

It is seen that the NMSE performances of the proposed BSCERD and BSCEDL schemems are better than that of other schemes and BSCEDL has the best performance. When $\mathrm{SNR} = 20$dB, BSCERD and BSCEDL respectively have 1.8dB and 2.8dB improvements compared with TDCEBS. Three schemes are better than OMP and SBL.

As shown in Fig. 6, we also compare the bit error rate (BER) performance of different schemes using the digital modulation method of quadrature phase shift keying (QPSK). When $\mathrm{SNR} = 20$dB, BSCERD and BSCEDL respectively have 0.5dB and 1dB BER performance improvements compared with TDCEBS.

The reasons for the performance differences are explained as follows. The SBL performs worse than the other schemes since it can only achieve approximately-sparse channel estimation which leads to power leakage on zero channel entries. OMP recovers each column vector of $\boldsymbol{H}$ independently without taking use of the block sparsity of $\boldsymbol{H}$. The other three schemes all take use of the block sparsity of $\boldsymbol{H}$ for sparse recovery, but

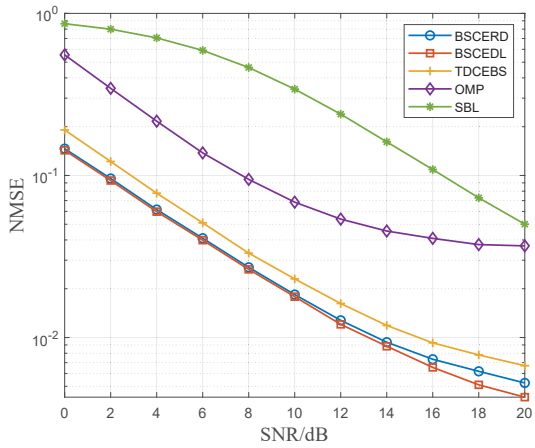

Fig. 5. Comparison of NMSE performance between different schemes.

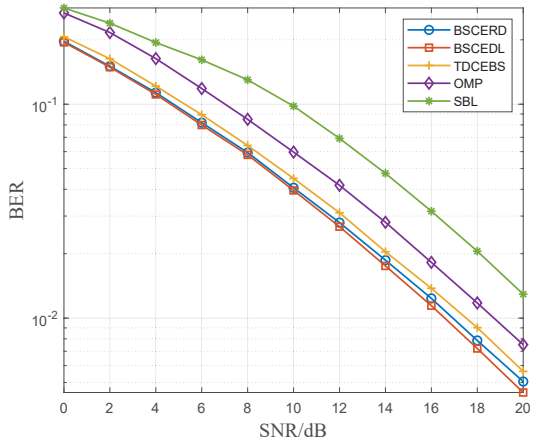

Fig. 6. Comparison of BER performance between different schemes.

TDCEBS only stops the iterations by the predefined number $N = L$, so the estimated results may include the indices of zero blocks, resulting in overfitting. BSCERD accurately filters the nonzero blocks of $\boldsymbol{H}$ exploiting the difference of the residual power for every two adjacent iterations. By learning the features of $\boldsymbol{\Lambda}$, BSCEDL estimates all the nonzero blocks of $\boldsymbol{H}$ simultaneously to improve the global optimality, which has the best performance.

Finally, we conduct a simulation of the time overhead performance of the schemes. All simulation code was executed in Matlab on a computer with a hexa-core processor, 16GB RAM and an NVIDIA GeForce GTX1660Ti graphics card. Table II compares the simulation time overhead of the four schemes. As shown in Table II, the runing time of BSCEDL is reduced by η_1, η_2 and η_3 respectively compared with BSCERD, TDCEBS and OMP. BSCEDL uses a trained neural network to predict all channel entries in parallel rather than iterative calculation, reducing time overhead greatly. The BSCERD and TDCEBS schemes need to exploit a stacked measurement matrix $\boldsymbol{B}$ and exploit the block sparsity of $\boldsymbol{H}$, so the time overhead will be greater than that of OMP.

TABLE II
TIME OVERHEAD COMPARISON

Schemes	Time overhead(s)	Time reduction ratio
BSCEDL	0.047	Baseline
BSCERD	8.384	$\eta_1 = 99.44\%$
TDCEBS	7.483	$\eta_2 = 99.37\%$
OMP	0.675	$\eta_3 = 93.04\%$

VI. CONCLUSIONS

In this paper, two block sparse time-domain channel estimation schemes have been proposed. BSCERD exploits residual difference to mitigate the overfitting of the existing TDCEBS scheme. BSCEDL improves the global optimality and reduces the time overhead by deep learning. The focus of our future study will be continued on deep learning based channel prediction and beamforming for mmWave massive MIMO.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China (NSFC) under Grant 62071116.

REFERENCES

[1] R. W. Heath, N. Gonzalez-Prelcic, S. Rangan, W. Roh, and A. Sayeed, "An overview of signal processing techniques for millimeter wave MIMO systems," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 3, pp. 436–453, Apr. 2016.

[2] X. Yang, M. Matthaiou, J. Yang, C.-K. Wen, F. Gao, and S. Jin, "Hardware-constrained millimeter-wave systems for 5G: Challenges, opportunities, and solutions," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 44–50, Jan. 2019.

[3] W. Ma, C. Qi, 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, May 2020.

[4] I.-S. Kim and J. Choi, "Spatial wideband channel estimation for mmWave massive MIMO systems with hybrid architectures and low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 4016–4029, June 2021.

[5] 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.

[6] S. Srivastava, A. Mishra, A. Rajoriya, 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.

[7] C. Qi, P. Dong, W. Ma, H. Zhang, Z. Zhang, and G. Y. Li, "Acquisition of channel state information for mmWave massive MIMO: Traditional and machine learning-based approaches," *Sci. China Inf. Sci.*, vol. 64, no. 8, Aug. 2021, Art. no. 181301.

[8] H. Kim, G.-T. Gil, and Y. H. Lee, "Two-step approach to time-domain channel estimation for wideband millimeter wave systems with hybrid architecture," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 5139–5152, July 2019.

[9] Y. Wang, C. Qi, P. Li, Z. Lu, and P. Lu, "Channel estimation for wideband mmWave MIMO OFDM system exploiting block sparsity," *IEEE Commun. Lett.*, vol. 26, no. 4, pp. 897–901, Apr. 2022.

[10] P. Boufounos, M. F. Duarte, and R. G. Baraniuk, "Sparse signal reconstruction from noisy compressive measurements using cross validation," in *Proc. IEEE/SP 14th Workshop Stat. Signal Process.*, Aug. 2007, pp. 299–303.

[11] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials," *IEEE Trans. Antennas Propagat.*, vol. 62, no. 6, pp. 3242–3256, Mar. 2014.