RC-Struct: A Structure-Based Neural Network Approach for MIMO-OFDM Detection

Jiarui Xu , Zhou Zhou , Lianjun Li , Lizhong Zheng, *Fellow, IEEE*, and Lingjia Liu , *Student Member, IEEE*

Abstract—In this paper, we introduce a structure-based neural network architecture, namely RC-Struct, for MIMO-OFDM symbol detection. The RC-Struct exploits the temporal structure of the MIMO-OFDM signals through reservoir computing (RC). A binary classifier leverages the repetitive constellation structure in the system to perform multi-class detection. The incorporation of RC allows the RC-Struct to be learned in a purely online fashion with extremely limited pilot symbols in each OFDM subframe. The binary classifier enables the efficient utilization of the precious online training symbols and allows an easy extension to high-order modulations without a substantial increase in complexity. Experiments show that the introduced RC-Struct outperforms both the conventional model-based symbol detection approaches and the state-of-the-art learning-based strategies in terms of bit error rate (BER). The advantages of RC-Struct over existing methods become more significant when rank and link adaptation are adopted. The introduced RC-Struct sheds light on combining communication domain knowledge and learning-based receive processing for 5G/5G-Advanced and Beyond.

Index Terms—MIMO-OFDM receive processing, neural networks, structure knowledge, online learning, 5G, 5G-advanced and QAM constellation.

I. INTRODUCTION

MULTIPLE input multiple output with orthogonal frequency division multiplexing (MIMO-OFDM) is the dominant waveform in modern wireless systems, such as 4G (LTE-Advanced) and 5G New Radio (NR). The MIMO technology provides additional degrees of freedom in the spatial domain, which can be exploited through spatial multiplexing to increase the channel capacity. To realize MIMO capacity gain, symbol detection is a crucial stage to recover multiple transmitted data streams from multiple receive antennas.

MIMO-OFDM symbol detection approaches generally fall into two categories: conventional model-based strategies and learning-based methods. Conventional model-based symbol

detection techniques usually rely on explicit modeling of the underlying system from the transmitter to the receiver. However, as wireless systems become more and more complicated with non-linear device components (e.g., power amplifier, low-resolution analog-to-digital converters), it is difficult to analytically model such behaviors [2]. Furthermore, standard model-based approaches are usually built on top of the estimated channel state information (CSI) at the receiver. The inaccurate modeling and CSI estimation may degrade the performance of such approaches, especially in the low signal-to-interference-plus-noise (SINR) regime.

The recent advances of deep learning and machine learning (ML) have captured much attention on exploiting neural networks (NN) to MIMO-OFDM symbol detection [3]–[6]. These learning-based approaches do not require explicit system modeling, circumventing the above issues of model-based methods. On the other hand, learning-based strategies have their own challenges. One major challenge is that the available online over-the-air training set is extremely limited. For example, in 3GPP LTE/LTE-Advanced and 5G NR systems, the pilot overhead is constrained to be below 20% to secure the spectral-efficiency [7]. The limited training data may lead to the model overfitting problem for NNs and degrade the performance.

Another challenge is that NNs are difficult to be trained due to their large amount of model weights. On one hand, a large number of model weights require more training data to learn. On the other hand, the training process might be challenging. For example, recurrent neural network (RNN), owing to its ability to process temporal data, has been widely explored in the symbol detection task [8]–[10]. However, training RNNs suffer from high computational complexity and training instability, such as the gradient vanishing and exploding problem.

One way to address these issues is to use a special type of RNN called reservoir computing (RC). The RC network consists of a RNN-based reservoir with fixed weights and a trainable output layer. The RNN-based reservoir and light-weighted training equip RCs with the ability to process temporal sequences and can be easily trained online. Many recent works have been devoted to investigating the effective way to utilize RC in the MIMO-OFDM symbol detection task [11]–[16]. However, such RC-based NNs are still quite generic without incorporating all available inherent structures of communication systems to realize the full potential of RC.

In this work, we introduce a new NN-based symbol detection method, RC-Struct, to leverage the inherent structural knowledge available in MIMO-OFDM systems:

- the time domain convolution and superposition due to the wireless channel;
- the time-frequency structure of the OFDM waveform;
- the repetitive structure of the modulation constellation.

In the time domain, we adopt RC to process sequential input as previous RC-based approaches [11]–[16] allowing the underlying NN to capture the temporal dynamics embedded in the received signal and to decouple the multiple transmitted data streams for MIMO operation. The main novelty of this work comes from the incorporation of the other two structure knowledge. Specifically, we leverage the time-frequency structure of the OFDM waveform to make full use of the training data in these domains. An additional NN in the frequency domain is designed to conduct symbol detection based on the repetitive structure of the modulation constellation enabling RC-Struct to conduct multi-class detection with the basic binary classifier. Due to the incorporation of the repetitive modulation structure, fewer training samples are needed for RC-Struct as opposed to generically designed NNs, significantly improving training sample efficiency. Furthermore, RC-Struct is able to conduct symbol detection completely online in a subframe-by-subframe fashion. In the experiments, we demonstrate that RC-Struct can outperform existing symbol detection methods for MIMO-OFDM systems in various scenarios and can conduct receive processing in a subframe-by-subframe fashion under link and rank adaptation. The main contributions of the paper are summarized as follows:

- We introduce RC-Struct for symbol detection in MIMO-OFDM systems, which takes advantage of the inherent structure knowledge of MIMO-OFDM systems. Different from previous RC-based approaches that only utilize RC in the time domain to capture the spatiotemporal correlation of the MIMO-OFDM system, we leverage the time-frequency structure of the OFDM waveform to make full use of the available training data in different domains. To be specific, a separate NN is introduced in the frequency domain to conduct detection/classification for transmitted symbols.
- The frequency domain NN takes into consideration of the repetitive structure of the modulation constellation. Specifically, the multi-class classification of the high order modulation is converted into parallel binary classification tasks. The inherent modulation constellation structure not only allows the NN to be easily adapted to different modulation orders without re-training, but also significantly improves the efficiency of utilizing the limited number of training samples.
- RC-Struct has been designed to work in a completely online fashion under dynamic rank and link adaptation on a subframe basis for MIMO-OFDM systems. Extensive experiments have been conducted to demonstrate the effectiveness and advantage of our design that combines inherent structure knowledge from time, frequency, and constellation in realistic cellular environments. It provides

a promising NN-based receive processing technique for 5G/5G-Advanced and Beyond.

The remainder of this paper is organized as follows: Sec. II contains the related work and Sec. III briefly discusses the structure of MIMO-OFDM systems. Sec. IV presents the RC-Struct while Sec. V provides experimental evaluation. Sec. VI concludes the paper.

Notations: In this paper, as we discuss the signal in both the time domain and frequency domain, we use lowercase and uppercase letters of x and y to differentiate the time domain and frequency domain signals. Specifically, for x and y, we use the non-bold letter for scalar and the bold letter for both vector and matrix. For other letters, we follow the convention to use the non-bold letter for the scalar, the bold lowercase letter for the vector, and the bold uppercase letter for the matrix. Sets get calligraphic letters. Special sets get denoted by blackboard bold. For example, $\mathbb{R}$ represents the real number space and $\mathbb{C}$ denotes the complex number space.

II. RELATED WORK

A. Deep Neural Network

Deep neural network (DNN) methods have been recently applied to the symbol detection task in wireless systems. State-of-the-art DNN symbol detection algorithms can be roughly divided into three categories: (1) multi-layer perception (MLP) based methods, (2) long short-term memory (LSTM) based methods, and (3) generative adversarial network (GAN) based methods, (4) convolution neural network (CNN) based methods [17], [18]. The first MLP-based approach, as discussed in [3], adopted five fully-connected layers for symbol detection in OFDM systems. Recent advances, such as DetNet [4], OAMPNet [5], MMNet [6], and HyperMIMO [19], construct MLP networks by incorporating trainable parameters from conventional iterative algorithms. To capture the temporal information, LSTM has been utilized to learn time-memory characteristics within the data [8]–[10]. For GAN models, recent efforts have been devoted to treating the time-frequency channel matrix as a 2D image and estimating the channel matrix and the transmit signals jointly [20]. More recently, DeepRx [18] is introduced to directly predict log-likelihood ratios (LLRs) of the sent bits with ResNet in the frequency domain, which has been shown to match the performance of the traditional linear minimum mean square error (LMMSE) receiver with full channel knowledge. In [17], a complex-valued network, DCCN, is adopted to directly estimate transmitted bits from the time domain signal, achieving appealing performance in different channel models.

While these DNN-based approaches achieve promising performance, they usually require a large amount of training data, making them difficult to be utilized in practice especially for 5G/5G-Advanced systems where the training is extremely limited. In some works, such as MMNet, perfect CSI is required to train the network, which is difficult to be obtained in practice. Different from the aforementioned DNN-based approaches, we train the network on a subframe basis, instead of training on multiple subframes, and only utilize the limited number of pilot symbols in one subframe. Such an online learning scheme

TABLE I
NOTATIONS APPEARING IN THE SYSTEM

Symbol	Definition
N_t	Number of transmitter antennas
N_r	Number of receiver antennas
N_{sc}	Number of OFDM sub-carriers
N_c	Number of channel realizations
N_{cp}	Length of Cyclic Prefix (CP)
N_p	Number of pilot symbols in one OFDM frame (training set)
N_d	Number of data symbols in one OFDM frame (testing set)
N	Total number of symbols in one OFDM frames ($N = N_p + N_d$)
$y_n^j(t)$	The tth sample of the nth OFDM symbol at the jth receive antenna in time domain
$x_n^i(t)$	The tth sample of the nth OFDM symbol at the ith transmit antenna in time domain
$\boldsymbol{y}_n^j \in \mathbb{C}^{(N_{\mathrm{sc}}+N_{\mathrm{cp}})\times 1}$	The nth OFDM symbol at the jth receive antenna in the time domain
$\boldsymbol{x}_n^i \in \mathbb{C}^{(N_{\mathrm{sc}}+N_{\mathrm{cp}})\times 1}$	The nth OFDM symbol at the ith transmit antenna in the time domain
$\boldsymbol{y}_n(t) \in \mathbb{C}^{N_r \times 1}$	The received tth sample of the nth OFDM symbol in time domain
$\boldsymbol{x}_n(t) \in \mathbb{C}^{N_t \times 1}$	The transmitted tth sample of the nth OFDM symbol in time domain
$\boldsymbol{y}_n \in \mathbb{C}^{N_r \times (N_{\mathrm{sc}}+N_{\mathrm{cp}})}$	The received nth OFDM symbol in time domain
$\boldsymbol{x}_n \in \mathbb{C}^{N_t \times (N_{\mathrm{sc}}+N_{\mathrm{cp}})}$	The transmitted nth OFDM symbol in time domain
$Y_n^j(k)$	The nth OFDM symbol at the jth receive antenna and kth subcarrier in frequency domain
$X_n^i(k)$	The nth OFDM symbol at the ith transmit antenna and kth subcarrier in frequency domain
$\boldsymbol{Y}_n^j \in \mathbb{C}^{N_{\mathrm{sc}} \times 1}$	The nth OFDM symbol at the jth receive antenna in frequency domain
$\boldsymbol{X}_n^i \in \mathbb{C}^{N_{\mathrm{sc}} \times 1}$	The nth OFDM symbol at the ith transmit antenna in frequency domain
$\boldsymbol{Y}_n(k) \in \mathbb{C}^{N_r \times 1}$	The received nth OFDM symbol at subcarrier k in frequency domain
$\boldsymbol{X}_n(k) \in \mathbb{C}^{N_t \times 1}$	The transmitted nth OFDM symbol at subcarrier k in frequency domain
$\boldsymbol{Y}_n \in \mathbb{C}^{N_r \times N_{\mathrm{sc}}}$	The received nth OFDM symbol in frequency domain
$\boldsymbol{X}_n \in \mathbb{C}^{N_t \times N_{\mathrm{sc}}}$	The transmitted nth OFDM symbol in frequency domain

makes RC-Struct a promising approach in 5G/5G-Advanced and Beyond with dynamic transmission modes.

B. Reservoir Computing in MIMO-OFDM Symbol Detection

RC is a special type of RNN that suits temporal data processing. Different from standard RNNs, RC can be learned with limited training data and less computational effort. An RC consists of a reservoir part and an output layer, where the reservoir remains fixed during training and only the output layer is updated, as shown in the time domain part of Fig. 2. This simple and effective training approach equips RC with the ability to efficiently learn with limited training data for symbol detection [11]–[16]. In [11], RC has been demonstrated to achieve compelling performance in online MIMO-OFDM symbol detection with limited training data. The follow-up RC-based works show that adding a sliding window to the RC [12] and using a cascaded deep RC [13], [14] further improve the performance. The most recent work has focused on tracking the channel change between OFDM symbols with scattered pilots for Wi-Fi systems and the associated hardware implementation [15].

Our work shares similar spirits with these efforts as we also build the NN based on RC to exploit the structural knowledge of convolution and superposition operation of the wireless channel in the time domain. However, the key difference lies in the fact that we further leverage the time-frequency structure of OFDM and the repetitive structure of the modulation constellation in the frequency domain to construct the underlying deep NN.

III. MIMO-OFDM SYSTEMS

In this section, we introduce the MIMO-OFDM system architecture where the notations are summarized in Tab. I.
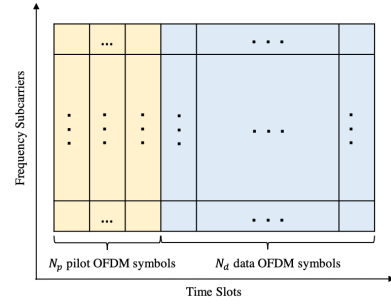

Fig. 1. OFDM subframe structure and pilot patterns.

In 4G/5G MIMO-OFDM systems, information is transmitted subframe by subframe where each subframe lasts for one millisecond. For simplicity, we assume each subframe contains N_p pilot symbols and N_d data symbols with $N = N_p + N_d$ symbols in total, as shown in Fig. 1. Consider a MIMO-OFDM system with N_t transmit antennas, N_r receive antennas, and N_{sc} subcarriers. The nth OFDM symbol ($n = 0, 1, \ldots, N - 1$) transmitted by the ith transmit antenna ($i = 0, 1, \ldots, N_t - 1$) in the *frequency domain* can be written as $\boldsymbol{X}_n^i \triangleq \left[X_n^i(0), X_n^i(1), \ldots, X_n^i(N_{\mathrm{sc}} - 1) \right]^T$, where $\boldsymbol{X}_n^i \in \mathbb{C}^{N_{\mathrm{sc}} \times 1}$ and $X_n^i(k)$ is the symbol modulated by quadrature amplitude modulation (QAM) at the kth subcarrier.

At the transmitter side, an inverse fast Fourier transform (IFFT) and cyclic prefix (CP) addition are applied to obtain the *time domain* transmission signal $\boldsymbol{x}_n^i \triangleq \left[x_n^i(0), x_n^i(1), \ldots, x_n^i(N_{\mathrm{sc}} + N_{\mathrm{cp}} - 1) \right]^T$, where $\boldsymbol{x}_n^i \in \mathbb{C}^{(N_{\mathrm{sc}}+N_{\mathrm{cp}})\times 1}$, N_{cp} is the length of the CP, and $x_n^i(t)$ is the tth sample of the nth OFDM symbol at the ith transmit antenna in the time domain with $t = 0, 1, \ldots, N_{\mathrm{sc}} + N_{\mathrm{cp}} - 1$.

The received signal at the jth receive antenna ($j = 0, 1, \ldots, N_r$) in the *time domain* can be expressed as

$$\boldsymbol{y}_n^j = \sum_{i=0}^{N_t-1} \boldsymbol{h}_n^{j,i} \circledast \phi(\boldsymbol{x}_n^i) + \boldsymbol{n}_n^j, \qquad (1)$$

where $\boldsymbol{y}_n^j \triangleq \left[y_n^j(0), y_n^j(1), \ldots, y_n^j(N_{sc} + N_{cp} - 1) \right]^T \in \mathbb{C}^{(N_{sc}+N_{cp}) \times 1}$, $\boldsymbol{h}_n^{j,i} \in \mathbb{C}^{L_c}$ is the channel impulse response between jth receive antenna and ith transmit antenna with L_c total number of delays; $\boldsymbol{n}_n^j$ stands for the the additive white gaussian noise (AWGN) at receiver j with zero mean and noise variance σ^2; $\circledast$ represents the circular convolution operation; $\phi(\cdot)$ is the non-linear operation such as power amplifier (PA).

The corresponding received signal $\boldsymbol{Y}_n^j$ in the *frequency domain* can be obtained by removing the CP and performing a fast Fourier transform (FFT), which can be denoted as $\boldsymbol{Y}_n^j \triangleq \left[Y_n^j(0), Y_n^j(1), \ldots, Y_n^j(N_{sc} - 1) \right]^T \in \mathbb{C}^{N_{sc} \times 1}$, where $Y_n^j(k)$ is the nth received symbol at the jth receiver and the kth subcarrier.

For training the RC-Struct network, we need the pilot symbols both in the time domain and frequency domain. For ease of discussion, we denote

$$\boldsymbol{y}_n(t) \triangleq \left[y_n^0(t), y_n^1(t), \ldots, y_n^{N_r-1}(t) \right]^T \in \mathbb{C}^{N_r \times 1},$$
$$\boldsymbol{x}_n(t) \triangleq \left[x_n^0(t), x_n^1(t), \ldots, x_n^{N_t-1}(t) \right]^T \in \mathbb{C}^{N_t \times 1},$$
$$\boldsymbol{Y}_n(k) \triangleq \left[Y_n^0(k), Y_n^1(k), \ldots, Y_n^{N_r-1}(k) \right]^T \in \mathbb{C}^{N_r \times 1},$$
$$\boldsymbol{X}_n(k) \triangleq \left[X_n^0(k), X_n^1(k), \ldots, X_n^{N_t-1}(k) \right]^T \in \mathbb{C}^{N_t \times 1},$$

where $\boldsymbol{x}_n(t) \in \mathbb{C}^{N_t \times 1}$ and $\boldsymbol{y}_n(t) \in \mathbb{C}^{N_r \times 1}$ are the transmitted and received tth sample of the nth OFDM symbol in the *time domain*; $\boldsymbol{X}_n(k) \in \mathbb{C}^{N_t \times 1}$ and $\boldsymbol{Y}_n(k) \in \mathbb{C}^{N_r \times 1}$ are the transmitted and received nth OFDM symbol at the kth subcarrier in the *frequency domain*. Then the corresponding matrix forms are

$$\boldsymbol{y}_n \triangleq [\boldsymbol{y}_n(0), \boldsymbol{y}_n(1), \ldots, \boldsymbol{y}_n(N_{sc}+N_{cp}-1)] \in \mathbb{C}^{N_r \times (N_{sc}+N_{cp})},$$
$$\boldsymbol{x}_n \triangleq [\boldsymbol{x}_n(0), \boldsymbol{x}_n(1), \ldots, \boldsymbol{x}_n(N_{sc}+N_{cp}-1)] \in \mathbb{C}^{N_t \times (N_{sc}+N_{cp})},$$
$$\boldsymbol{Y}_n \triangleq [\boldsymbol{Y}_n(0), \boldsymbol{Y}_n(1), \ldots, \boldsymbol{Y}_n(N_{sc}-1)] \in \mathbb{C}^{N_r \times N_{sc}},$$
$$\boldsymbol{X}_n \triangleq [\boldsymbol{X}_n(0), \boldsymbol{X}_n(1), \ldots, \boldsymbol{X}_n(N_{sc}-1)] \in \mathbb{C}^{N_t \times N_{sc}},$$

where $\boldsymbol{x}_n \in \mathbb{C}^{N_t \times (N_{sc}+N_{cp})}$ and $\boldsymbol{y}_n \in \mathbb{C}^{N_r \times (N_{sc}+N_{cp})}$ stand for the transmitted and received nth OFDM symbol in the time domain, and $\boldsymbol{X}_n \in \mathbb{C}^{N_t \times N_{sc}}$ and $\boldsymbol{Y}_n \in \mathbb{C}^{N_r \times N_{sc}}$ represent the nth OFDM symbol in the frequency domain.

The training dataset $\{\mathcal{D}_n\}_{n=0}^{N_p-1}$ can be represented as

$$\mathcal{D}_n \triangleq \left(\{\boldsymbol{y}_n(t)\}_{t=0}^{N_{sc}+N_{cp}-1}, \{\boldsymbol{x}_n(t)\}_{t=0}^{N_{sc}+N_{cp}-1}, \{\boldsymbol{X}_n(k)\}_{k=0}^{N_{sc}-1} \right)$$
$$= (\boldsymbol{y}_n, \boldsymbol{x}_n, \boldsymbol{X}_n), \qquad (2)$$

where $\{\boldsymbol{y}_n(t)\}_{t=0}^{N_{sc}+N_{cp}-1}$ will be the input to the network; $\{\boldsymbol{x}_n(t)\}_{t=0}^{N_{sc}+N_{cp}-1}$ and $\{\boldsymbol{X}_n(k)\}_{k=0}^{N_{sc}-1}$ will be the target output in the time domain and frequency domain respectively; $\boldsymbol{y}_n$, and $\boldsymbol{x}_n$, $\boldsymbol{X}_n$ are the matrix form input and target.

IV. THE INTRODUCED APPROACH — RC-STRUCT

We introduce the RC-Struct method to exploit the properties of OFDM signals in both the time and frequency domain

for symbol detection. The introduced method is composed of two parts: RC-based time domain data-stream decoupling and equalization as well as the structure-based frequency domain NN classification. The received signal is first decoupled and equalized by RC in the time domain and then classified by the NN in the frequency domain. The architecture of the network is shown in Fig. 2. We will first discuss the time domain RC and then focus on our structured NN in the frequency domain.

A. Time Domain: Reservoir Computing

1) Introduction of Reservoir Computing: RNNs have been broadly applied in temporal data processing, for their special feedback and skip connections for generating history-dependent features. However, training RNNs is inherently difficult and time-consuming. RC is an alternative RNN-based framework that has fast-learning capability on a variety of temporal recognition tasks. The Echo State Network (ESN) [21], [22], as a specific type of RC model, consists of the input layer, reservoir unit, and the output layer, as shown in Fig. 3. Unlike conventional RNNs, the training of ESN is simple and fast. Specifically, during training, only the weights of the output layer are learned and updated, while the weights for the input layer and the reservoir are randomly initialized and fixed. To ensure the ESN works, the reservoir should be appropriately designed. The RNN-based reservoir should satisfy the echo state property so that the network can asymptotically eliminate any information from the initial condition [21], [23]. In [24], it is shown that the echo state property can be satisfied if the spectral radius of the reservoir transition matrix is smaller than unity. Thanks to the fast and simple training process, RC now has achieved promising performance in many tasks, such as speech recognition [25], [26], image detection [27], [28], wireless communication [11]–[15], etc.

2) Reservoir Computing in RC-Struct: RC-Struct adopts ESN in the time domain to decouple the received data streams and equalize the received signals. The input to the reservoir is the received signal $\boldsymbol{y}_n(t)$ and the estimated output is $\hat{\boldsymbol{x}}_n(t)$, which corresponds to the transmit signal $\boldsymbol{x}_n(t)$. The training dataset for RC in matrix form can be represented as

$$D_{rc} \triangleq \left(\left[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_p-1} \right], \left[\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_p-1} \right] \right). \quad (3)$$

For an ESN with N_n neurons in the reservoir, the states and the output are expressed by the following equations:

$$\boldsymbol{s}_n(t) = f(\boldsymbol{W} \boldsymbol{s}_n(t-1) + \boldsymbol{W}_{in} \boldsymbol{y}_n(t)), \qquad (4)$$
$$\hat{\boldsymbol{x}}_n(t) = f_{out}(\boldsymbol{W}_{out} \boldsymbol{z}_n(t)), \qquad (5)$$

where $\boldsymbol{s}_n(t) \in \mathbb{C}^{N_n \times 1}$ is the internal state estimated by the reservoir with $\boldsymbol{s}_n(-1)$ initialized as a zero vector, $\boldsymbol{z}_n(t) = \left[\boldsymbol{s}_n(t)^T, \boldsymbol{y}_n(t)^T \right]^T \in \mathbb{C}^{(N_n+N_r) \times 1}$ is the concatenation of the internal state and the input, $\boldsymbol{W}_{in} \in \mathbb{C}^{N_n \times N_r}$ is the input weight matrix, $\boldsymbol{W} \in \mathbb{C}^{N_n \times N_n}$ is the reservoir weight matrix, $\boldsymbol{W}_{out} \in \mathbb{C}^{N_t \times (N_n+N_r)}$ is the output weight matrix. The $f(\cdot)$ and $f_{out}(\cdot)$ are the activation functions for the internal units and the output units respectively. In general, identity transformation is used for the output activation function.
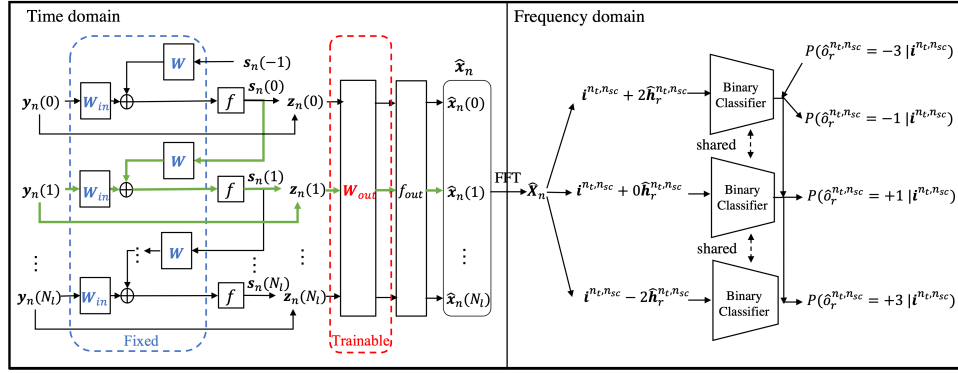
Fig. 2. The architecture of RC-Struct network. For better visualization, we denote $N_l = N_{sc} + N_{cp} - 1$ in the figure. In the time domain, the fixed weights are marked as blue and the trainable weights are colored with red. Note that all the $\boldsymbol{W}_{in}$'s share the same weights and all the $\boldsymbol{W}$'s share the same weights. The arrows for processing $\boldsymbol{y}_n(1)$ are highlighted in green to show how a specific $\boldsymbol{y}_n(t)$ is processed.
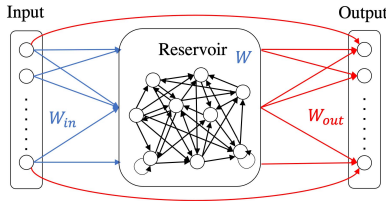

Fig. 3. Echo State Network. The fixed weights are highlighted in blue. The trainable weights are highlighted in red.

In conventional RNNs, the input weights $\boldsymbol{W}_{in}$, the reservoir weights $\boldsymbol{W}$, and the output weights $\boldsymbol{W}_{out}$ are all learned through the backpropagation algorithm. Different from the conventional RNNs, in RC, only the output weight matrix $\boldsymbol{W}_{out}$ is updated and all the other weights are fixed. More importantly, the output weight matrix is learned through a closed-form least-square solution. Specifically, RC is designed to minimize the mean square error between the network output signal $\hat{\boldsymbol{x}}_n(t)$ and the desired transmit signal $\boldsymbol{x}_n(t)$. The regression procedure can be described as

$$\hat{\boldsymbol{W}}_{\text{out}} = \arg \min_{\boldsymbol{W}_{\text{out}}} \sum_{n=0}^{N_p-1} \sum_{t=0}^{N_{cp}+N_{sc}-1} \|\hat{\boldsymbol{x}}_n(t) - \boldsymbol{x}_n(t)\|_2^2$$

$$= \arg \min_{\boldsymbol{W}_{\text{out}}} \sum_{n=0}^{N_p-1} \|\hat{\boldsymbol{x}}_n - \boldsymbol{x}_n\|_F^2 = \arg \min_{\boldsymbol{W}_{\text{out}}} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_F^2, \quad (6)$$

where $\hat{\boldsymbol{x}}_n = [\hat{\boldsymbol{x}}_n(0), \hat{\boldsymbol{x}}_n(1), \ldots, \hat{\boldsymbol{x}}_n(N_{sc} + N_{cp} - 1)] \in \mathbb{C}^{N_t \times (N_{sc}+N_{cp})}$ is the matrix form of the output signal, $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_0, \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{N_p-1}] \in \mathbb{C}^{N_t \times N_p(N_{sc}+N_{cp})}$ and $\boldsymbol{x} = [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_p-1}] \in \mathbb{C}^{N_t \times N_p(N_{sc}+N_{cp})}$ are the concatenation of the output signal matrices and target transmit signal matrices, respectively. In the training period, the concatenated vector $\boldsymbol{z}_n(t)$ is recorded and forms the following matrix:

$$\boldsymbol{Z}_n = [\boldsymbol{z}_n(0), \boldsymbol{z}_n(1), \ldots, \boldsymbol{z}_n(N_{cp} + N_{sc} - 1)] \in \mathbb{C}^{(N_n+N_r) \times (N_{cp}+N_{sc})},$$
$$(7)$$

$$\boldsymbol{Z} = [\boldsymbol{Z}_0, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{N_p-1}] \in \mathbb{C}^{(N_n+N_r) \times N_p(N_{cp}+N_{sc})}. \quad (8)$$

If $f_{\text{out}}(\cdot)$ is set as the identity function, the weights are updated by the least square solution

$$\hat{\boldsymbol{W}}_{\text{out}} = \boldsymbol{x} \boldsymbol{Z}^\dagger, \quad (9)$$

where $\boldsymbol{Z}^\dagger$ is the Moore-Penrose pseudo inverse of $\boldsymbol{Z}$.

At the training time, a delay parameter d will be learned to compensate for the lag-effect caused by the feedback nature of RC. The training dataset with delay d can be re-written as

$$\mathcal{D}_{\text{rc}}^{(d)} \triangleq \left(\left[\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{N_p-1}, \boldsymbol{0}_{N_r \times d} \right], \left[\boldsymbol{0}_{N_t \times d}, \boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_p-1} \right] \right),$$
$$(10)$$

where $\boldsymbol{0}$ denotes the zero matrix. The objective function can then be written as

$$\hat{d}, \hat{\boldsymbol{W}}_{\text{out}} = \arg \min_d \arg \min_{\boldsymbol{W}_{\text{out}}} \|\hat{\boldsymbol{x}}' - \boldsymbol{x}'\|_F^2, \quad (11)$$

where $\boldsymbol{x}'$ is the target transmit signal with zero inserted, and $\hat{\boldsymbol{x}}'$ stands for the RC output when reading in received signal with zero inserted.

Following the training procedure in [14], the RC is trained with different values of d in the range of $[0, N_{cp}]$ with a step size of p. The optimal delay $\hat{d}$ is determined by finding the value d that generates the minimal object value defined in eq. (6). As discussed in [12], utilizing a sliding window when processing the input signal can further increase the short-term memory capacity of ESN. Following this work, we adopt a sliding window with length N_w to process the input data. Specifically, the input to the ESN includes both the current tth sample and also the previous samples from $t - N_w + 1$ to $t - 1$. More detailed discussions have been provided in our previous work [12], [14].

In summary, as shown in our previous work on symbol detection for MIMO-OFDM systems, RC can effectively decouple corresponding data streams and equalize the received signals in the time domain [11]–[15].

B. Frequency Domain: Structure-Based Neural Network

The novel ingredient of RC-Struct comes in the frequency domain. We introduce the structure-based NN, which leverages the time-frequency structure of the OFDM waveform as well as the repetitive structure of the modulation constellation.

The RC in the time domain and the classifier in the frequency domain are trained separately. The output weights of RC are first learned through the closed-form least square solution. After the training of RC, we obtain the RC output $\hat{\boldsymbol{x}}_n$ by processing the time domain received signal $\boldsymbol{y}_n$ with the trained RC. The input to the frequency domain classifier is $\hat{\boldsymbol{X}}_n \in \mathbb{C}^{N_t \times N_{\text{sc}}}$, which is the FFT of the RC output $\hat{\boldsymbol{x}}_n$. The training data of the frequency domain network is the RC output in the frequency domain $\hat{\boldsymbol{X}}_n$ and the transmitted symbols in the frequency domain $\boldsymbol{X}_n$.

As the introduced approach works the same for all the transmitted symbols, we drop the index n in the following analysis for ease of discussion. Since RC has decoupled the data streams, the underlying relationship between the output of RC and the transmitted symbols in the frequency domain can be represented as

$$\hat{x}^{n_t,n_{\text{sc}}} = h^{n_t,n_{\text{sc}}} x^{n_t,n_{\text{sc}}} + g^{n_t,n_{\text{sc}}}, \qquad (12)$$

where $\hat{x}^{n_t,n_{\text{sc}}}$ and $x^{n_t,n_{\text{sc}}}$ are the $(n_t, n_{\text{sc}})^{\text{th}}$ entry of the $\hat{\boldsymbol{X}}_n$ and $\boldsymbol{X}_n$, respectively; $h^{n_t,n_{\text{sc}}}$ represents the effective channel coefficients after RC-based time decoupling and equalization; $g^{n_t,n_{\text{sc}}}$ is the additive noise after the processing of RC. By transforming the complex values into real values, we have

$$\boldsymbol{i}^{n_t,n_{\text{sc}}} = \tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}} o_r^{n_t,n_{\text{sc}}} + \tilde{\boldsymbol{h}}_{\text{im}}^{n_t,n_{\text{sc}}} o_{\text{im}}^{n_t,n_{\text{sc}}} + \tilde{\boldsymbol{g}}^{n_t,n_{\text{sc}}}, \quad (13)$$

where

$$\boldsymbol{i}^{n_t,n_{\text{sc}}} = \begin{bmatrix} \Re\{\hat{x}^{n_t,n_{\text{sc}}}\} \\ \Im\{\hat{x}^{n_t,n_{\text{sc}}}\} \end{bmatrix}, o_r^{n_t,n_{\text{sc}}} = \Re\{x^{n_t,n_{\text{sc}}}\}, o_{\text{im}}^{n_t,n_{\text{sc}}} = \Im\{x^{n_t,n_{\text{sc}}}\},$$

and

$$\tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}} = \begin{bmatrix} \Re\{h^{n_t,n_{\text{sc}}}\} \\ \Im\{h^{n_t,n_{\text{sc}}}\} \end{bmatrix}, \quad \tilde{\boldsymbol{h}}_{\text{im}}^{n_t,n_{\text{sc}}} = \begin{bmatrix} -\Im\{h^{n_t,n_{\text{sc}}}\} \\ \Re\{h^{n_t,n_{\text{sc}}}\} \end{bmatrix}.$$

Note that the value of $o_r^{n_t,n_{\text{sc}}}$ and $o_{\text{im}}^{n_t,n_{\text{sc}}}$ are in the set of $\mathcal{C} = \{-2K - 1, -2K + 1, \ldots, +2K - 1, +2K + 1\}$ for M-QAM ($M \in \{4, 16, 64, \ldots\}$), where $K = \frac{\sqrt{M}-2}{2}$. Specifically, for QPSK, the class set is $\{-1, +1\}$. The symbol detection task can thus be formulated as a classification problem in the frequency domain. For simplicity, we focus on discussing the real-value part. But the same conclusion holds for the imaginary part by replacing $\hat{o}_r^{n_t,n_{\text{sc}}}$ with $\hat{o}_{\text{im}}^{n_t,n_{\text{sc}}}$ and $\tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}$ with $\tilde{\boldsymbol{h}}_{\text{im}}^{n_t,n_{\text{sc}}}$.

1) Network Architecture: Due to the repetitive structure of the modulation constellation, we observe that

$$
\begin{aligned}
&P\{\hat{o}_r^{n_t,n_{\text{sc}}} = o_r^{n_t,n_{\text{sc}}} | \boldsymbol{i}^{n_t,n_{\text{sc}}}\} \\
&= P\{\hat{o}_r^{n_t,n_{\text{sc}}} = +1 | \boldsymbol{i}^{n_t,n_{\text{sc}}} + (-o_r^{n_t,n_{\text{sc}}} + 1) \\
&\qquad \cdot \tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}\}, \qquad\qquad\qquad\qquad (14) \\
&P\{\hat{o}_r^{n_t,n_{\text{sc}}} = o_r^{n_t,n_{\text{sc}}} | \boldsymbol{i}^{n_t,n_{\text{sc}}}\} \\
&= P\{\hat{o}_r^{n_t,n_{\text{sc}}} = -1 | \boldsymbol{i}^{n_t,n_{\text{sc}}} + (-o_r^{n_t,n_{\text{sc}}} - 1) \\
&\qquad \cdot \tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}\}, \qquad\qquad\qquad\qquad (15)
\end{aligned}
$$

where $\hat{o}_r^{n_t,n_{\text{sc}}}$ is the estimated real-value part of the transmitted symbol. The observation indicates that the multi-class detection can be transformed into a binary classification between $+1$ and -1 by shifting $\boldsymbol{i}^{n_t,n_{\text{sc}}}$ with either $(-o_r^{n_t,n_{\text{sc}}} + 1) \cdot \tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}$ or $(-o_r^{n_t,n_{\text{sc}}} - 1) \cdot \tilde{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}$. We name the $-o_r^{n_t,n_{\text{sc}}} + 1$ and $-o_r^{n_t,n_{\text{sc}}} - 1$ as the shifting parameter.

We leverage these properties to construct the network in the frequency domain. Due to the repetitive structure of the modulation constellation points, the multi-class classification for M-QAM can be divided into several binary classification processes through a shifting process. Thus, the network only consists of a binary classifier. The shifting process is conducted by utilizing the effective CSI between the RC processed signal and transmit signal. Since the perfect CSI is unknown, the effective channel is estimated through LMMSE, which is denoted as $\hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}$. In Fig. 2, we show an example when the network is detecting the real-value part of the transmitted symbols modulated in 16-QAM.

2) Training Process: Denote the shifting parameter as $s_r^{n_t,n_{\text{sc}}}$. At the training time, the input to the classifier is $\boldsymbol{i}^{n_t,n_{\text{sc}}} + s_r^{n_t,n_{\text{sc}}} \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}$, where $s_r^{n_t,n_{\text{sc}}}$ is either $-o_r^{n_t,n_{\text{sc}}} + 1$ or $-o_r^{n_t,n_{\text{sc}}} - 1$. When $s_r^{n_t,n_{\text{sc}}} = -o_r^{n_t,n_{\text{sc}}} + 1$, the binary label for the input is $b_r^{n_t,n_{\text{sc}}} = +1$. When $s_r^{n_t,n_{\text{sc}}} = -o_r^{n_t,n_{\text{sc}}} - 1$, the binary label for the input is $b_r^{n_t,n_{\text{sc}}} = -1$. Due to this unique process of generating training labels, one pair of data $(\boldsymbol{i}^{n_t,n_{\text{sc}}}, o_r^{n_t,n_{\text{sc}}})$ can be used to construct two binary training samples, making the network exploit training data more efficiently.

The binary classifier consists of two linear layers and a non-linear function between the two layers. If the estimated binary label is represented as $\hat{b}_r^{n_t,n_{\text{sc}}}$, then the function approximated by the binary classifier can be written as $f_{n_t,n_{\text{sc}}}(\hat{b}_r^{n_t,n_{\text{sc}}}; \boldsymbol{i}^{n_t,n_{\text{sc}}} + s_r^{n_t,n_{\text{sc}}} \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}})$.

3) Testing Process: At the testing time, the input $\boldsymbol{i}^{n_t,n_{\text{sc}}}$ is tested with all the possible shifting parameters in the set $\mathcal{S} = \{-2K, -2K + 2, \ldots, 2K - 2, 2K\}$. When transmitted with QPSK, $K = 0$ and the set is $S = \{0\}$. Following eq. (14) and eq. (15), the estimated pairwise likelihood ratio for classes in set $\mathcal{C}$ can be obtained by

$$
\begin{aligned}
&\frac{P_{n_t,n_{\text{sc}}}\{\hat{o}_r^{n_t,n_{\text{sc}}} = -2k + 1 | \boldsymbol{i}^{n_t,n_{\text{sc}}}\}}{P_{n_t,n_{\text{sc}}}\{\hat{o}_r^{n_t,n_{\text{sc}}} = -2k - 1 | \boldsymbol{i}^{n_t,n_{\text{sc}}}\}} \\
&= \frac{f_{n_t,n_{\text{sc}}}(\hat{b}_r^{n_t,n_{\text{sc}}} = +1; \boldsymbol{i}^{n_t,n_{\text{sc}}} + 2k \cdot \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}})}{f_{n_t,n_{\text{sc}}}(\hat{b}_r^{n_t,n_{\text{sc}}} = -1; \boldsymbol{i}^{n_t,n_{\text{sc}}} + 2k \cdot \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}})}, \quad (16)
\end{aligned}
$$

where $k = -K, -K + 1, \ldots, +K$ is the index of each shifting parameter. For ease of discussion, we denote the likelihood ratio as

$$
\begin{aligned}
&\mathcal{L}_{+-}(\boldsymbol{i}^{n_t,n_{\text{sc}}} + 2k \cdot \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}) \\
&:= \frac{f_{n_t,n_{\text{sc}}}(\hat{o}_r^{n_t,n_{\text{sc}}} = +1; \boldsymbol{i}^{n_t,n_{\text{sc}}} + 2k \cdot \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}})}{f_{n_t,n_{\text{sc}}}(\hat{o}_r^{n_t,n_{\text{sc}}} = -1; \boldsymbol{i}^{n_t,n_{\text{sc}}} + 2k \cdot \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}})}. \quad (17)
\end{aligned}
$$

Then the eq. (16) can be expressed as

$$
\begin{aligned}
&\frac{P_{n_t,n_{\text{sc}}}\{\hat{o}_r^{n_t,n_{\text{sc}}} = -2k + 1 | \boldsymbol{i}^{n_t,n_{\text{sc}}}\}}{P_{n_t,n_{\text{sc}}}\{\hat{o}_r^{n_t,n_{\text{sc}}} = -2k - 1 | \boldsymbol{i}^{n_t,n_{\text{sc}}}\}} \\
&\qquad\qquad = \mathcal{L}_{+-}(\boldsymbol{i}^{n_t,n_{\text{sc}}} + 2k \cdot \hat{\boldsymbol{h}}_r^{n_t,n_{\text{sc}}}).
\end{aligned}
$$

TABLE II

COMPLEXITY COMPARISON

Method	Train Complexity	Test Complexity
RCNet	$\mathcal{O}(V(N_n + N_{\text{train}} + N_t)(N_n + N_r)N_{\text{train}})$	$\mathcal{O}(V(N_n + N_r)(N_n N_{\text{test}} + N_t))$
RC-Struct	$\mathcal{O}(V(N_n + N_{\text{train}} + N_t)(N_n + N_r)N_{\text{train}} + 8N_h N_t N_{\text{sc}} N_p N_{\text{ep}})$	$\mathcal{O}(V(N_n + N_r)(N_n N_{\text{test}} + N_t)) + 4N_h N_t N_{\text{sc}} N_d$

By collecting all the pairwise likelihood ratios, the posterior marginal estimation of each class can be written as

$$
\begin{aligned}
P\{\hat{o}_r^{n_t, n_{\text{sc}}} &= -2k+1 | \boldsymbol{i}^{n_t, n_{\text{sc}}}\} \\
&= P\{\hat{o}_r^{n_t, n_{\text{sc}}} = -2K - 1 | \boldsymbol{i}^{n_t, n_{\text{sc}}}\} \\
&\times \prod_{k'=k}^{K} \mathcal{L}_{+-}(\boldsymbol{i}^{n_t, n_{\text{sc}}} + 2k' \cdot \hat{\boldsymbol{h}}_r^{n_t, n_{\text{sc}}}),
\end{aligned} \tag{18}
$$

where we assume constant probability $P\{\hat{o}_r^{n_t, n_{\text{sc}}} = -2K - 1 | \boldsymbol{i}^{n_t, n_{\text{sc}}}\}$. The final class is chosen as the class that has the maximum probability.

C. Complexity Analysis

In our previous work [12], [14], [29], we have shown that our RC-based approaches have less computation complexity than the LMMSE method when the number of subcarriers is large. As the complexity comparison with conventional methods has been detailed discussed in our previous work, we focus on the complexity comparison between RCNet [14] and the introduced RC-Struct. Since the costs for matrix addition and element-wise operation are negligible compared with the matrix multiplication and pseudo-inverse operation, we mainly consider the computation cost of the matrix multiplication and pseudo-inverse operation. Note that the complexity for the multiplication of a $m \times n$ matrix and a $n \times k$ matrix is $\mathcal{O}(mnk)$, and the complexity for pseudo-inverse of a $m \times n$ matrix ($m \le n$) is $\mathcal{O}(mn^2)$ when it is implemented by single value decomposition.

Denote the number of training samples in the time domain as $N_{\text{train}} = (N_{\text{cp}} + N_{\text{sc}})N_p$ and the number of testing samples in the time domain as $N_{\text{test}} = (N_{\text{cp}} + N_{\text{sc}})N_d$. For simplicity, the complexity of the delay learning process and the sliding window process are ignored here, as they do not change the order of magnitude of the complexity. The training complexity for RC is the sum of the state update complexity $\mathcal{O}((N_n + N_r)N_n N_{\text{train}})$ and $\hat{\boldsymbol{W}}_{\text{out}}$ estimation complexity $\mathcal{O}(N_{\text{train}}^2(N_n + N_r) + N_t N_{\text{train}}(N_n + N_r))$. Therefore, the training complexity for RC is $\mathcal{O}((N_n + N_{\text{train}} + N_t)(N_n + N_r)N_{\text{train}})$. At testing time, the forward pass includes state update process in eq. (4) and the output estimation process in eq. (5). Thus, the testing complexity for RC is $\mathcal{O}((N_n + N_r)(N_n N_{\text{test}} + N_t))$. In RCNet, as V layers of RC are cascaded, the total training and testing complexity are $\mathcal{O}(V(N_n + N_{\text{train}} + N_t)(N_n + N_r)N_{\text{train}})$ and $\mathcal{O}(V(N_n + N_r)(N_n N_{\text{test}} + N_t))$.

As the RC-Struct builds on top of RCNet, it shares the same complexity as RCNet in the time domain. In the frequency domain, the binary classifier with two linear layers is adopted for classification. Suppose the number of neurons in the first layer is N_h. As the input size to the first layer is 2, the complexity for passing the first layer is $\mathcal{O}(2N_h)$ per input sample. As the classifier only has two classes, the number of neurons in the second layer is 2. Then the complexity for passing the second layer is also $\mathcal{O}(2N_h)$ per input sample. Therefore, the complexity for inferring the binary classifier is $\mathcal{O}(4N_h)$ per input sample. At training time, the number of training samples is $2N_t N_{\text{sc}} N_p$, as one pair of data can construct two binary training samples. If the number of training epochs is set as N_{ep}, then the training complexity is $\mathcal{O}(8N_h N_t N_{\text{sc}} N_p N_{\text{ep}})$. The number of testing samples is $N_t N_{\text{sc}} N_d$. The testing complexity is $\mathcal{O}(4N_h N_t N_{\text{sc}} N_d)$. The total complexity of RC-Struct is calculated by the sum of the complexity in the time domain and frequency domain.

In Tab. II, we summarized the training and testing complexities for both methods. As shown in the table, RC-Struct has higher training and testing complexity than RCNet due to its extra network in the frequency domain. However, the complexity of RC-Struct and RCNet are still in the same order of magnitude.

V. PERFORMANCE EVALUATION

A. Experimental Setting

This section conducts the performance evaluation of the introduced RC-Struct for symbol detection in MIMO-OFDM systems. A typical MIMO-OFDM system with $N_t = 4$ transmit antennas and $N_r = 4$ receive antennas will be investigated. Without adaptation in the transmission mode, the system will conduct a fixed rank 4 transmission with a fixed modulation of 16-QAM. Gray coding is adopted for constellation mapping. This is the typical evaluation scenario that has been conducted in the majority of NN-based symbol detection papers. However, it is important to note that in reality, 5G/5G-Advanced systems adopt a much more dynamic transmission operation where link adaptation and rank adaptation are done on a millisecond basis [30]. Other system parameters in the evaluation are configured as $N_{\text{sc}} = 1024, N_{\text{cp}} = 160, N_p = 4, N_d = 16, N = 20$. The training overhead of the underlying system is 20% as opposed to other learning-based methods that require a prohibitively large training set. Meanwhile, it also complies with the pilot occupancy requirement specified in the 3GPP LTE/LTE-Advanced and 5G NR systems [7]. For simplicity, the first N_p OFDM symbols of each subframe are set to be pilot symbols, while the rest N_d OFDM symbols are set to be data symbols, as shown in Fig. 1. The pilot symbols are randomly chosen from the QAM

constellation. It is important to note that similar to our previous work [12], RC-Struct can be readily extended to scattered pilot patterns where the training overhead can be significantly reduced. The channel coefficients are generated using the QuaDRiGa channel simulator [31] following the 3GPP 3D MIMO model defined in [32]. In this paper, we focus on the scenario when users are in low mobilities. To be specific, the user speed for generating the channel is set as 5 km/h. Our work on high mobility users can be found in [33].

The RC used in the experiments has $N_n = 16$ neurons and the input window length is $N_w = 128$. The step size for searching the optimal delay parameter is set as $p = 5$. Following RCNet, two cascaded RC are adopted in the time domain. In the frequency domain, the binary classifier consists of two linear layers, each of which has 128 neurons. The two layers are connected with hyperbolic tangent non-linear function. The weights of the hidden layers are initialized with Xavier weight initialization [34]. The network is updated with SGD methods with the learning rate of 0.01 and the momentum of 0.001. The underlying network is trained for 800 epochs. During training, seven resource block groups (RBGs), each of which consists of 12 subcarriers, are combined together to train a single classification network in order to reduce the computation complexity.

In our experiments, we compare our RC-Struct with state-of-the-art learning algorithms: RCNet and MMNet. For conventional model-based signal processing strategies, we focus on the popular LMMSE-based and sphere decoding-based algorithms. Overall, the compared schemes include: (1) *LMMSE+LMMSE-CSI*: The linear decoder that exploits the LMMSE-based symbol detector with the LMMSE-based estimated CSI; (2) *SD+LMMSE-CSI*: The sphere decoding-based method for symbol detection using the LMMSE-based estimated CSI [35]; (3) *MMNet*: The MMNet network designed for symbol detection under arbitrary channel matrices using the LMMSE-based estimated CSI [6]. To achieve the best performance, we adopt the offline setting in [6] and train the network for each subcarrier with 500 iterations. Note that the network for each subcarrier has trained for 1000 iterations in [6]. However, as we have 1024 subcarriers, it is time-consuming to train the network for each subcarrier with 1000 iterations. Moreover, we observe that the training BER does not change anymore after 500 iterations and thus choose to train for 500 iterations to avoid possible overfitting; (4) *RCNet*: The RC-based approach with 2 cascaded RC in time domain [14]. All the parameters are set the same as the time domain RC in RC-Struct; (5) *RC-Struct*: The introduced method with LMMSE-based estimated shifting parameters. We notice that initializing MMNet with pre-trained offline weights can lead to better performance than training from scratch. However, when dynamic link adaptation and/or rank adaptation are adopted, it is almost impossible to initialize the network with pre-trained weights due to the underlying model mismatch. Since this paper focuses on realistic 5G/5G-Advanced transmission operations with dynamic rank and link adaptation, we do not incorporate the results for MMNet with pre-trained weights.
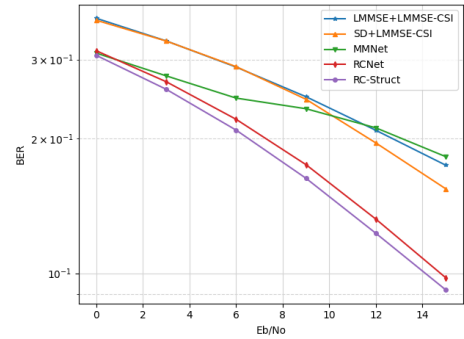

Fig. 4. Comparison of BER in the linear region.

B. Key Performance Indicators (KPI) / Performance Metrics

In this paper, we use BER as the key performance indicator (KPI) / evaluation metric when no adaptation is applied. When link and rank adaptations are adopted, BER along is not suitable as the KPI since it neglects the order of different modulation as well as the rank of the transmission. Accordingly, we adopt the RawBER as the KPI / metric to evaluate the underlying system performance [36] so that data streams with different modulation orders will be factored in the performance evaluation. To be specific, the RawBER is defined as

$$RawBER = \frac{\sum_{j=1}^{N_{ds}} b_j BER_j}{\sum_{j=1}^{N_{ds}} b_j}, \quad (19)$$

where N_{ds} is the number of data streams, b_j is the number of bits per symbol and BER_j is the BER of the detector on the jth data stream. Note that RawBER is equivalent to BER if all streams are modulated with the same modulation order. The performance evaluation is discussed with the plot of BER or RawBER versus signal to noise ratio in terms of E_b/N_o in the dB scale, where each BER point or RawBER point is tested with 100 subframes.

C. Comparison With the State-of-Art Detection Strategies

To evaluate the performance under the task of symbol detection, we first compare RC-Struct with different approaches in terms of the BER without transmission adaptation.

In Fig. 4, we show the BER as a function of E_b/N_o for various symbol detection methods. As indicated from the results, both RC-based methods achieve better symbol detection performance compared with conventional model-based methods, which highlights the effectiveness of RC-based schemes in the linear region of the channel. As the inaccurate LMMSE channel estimation in the low E_b/N_o regime affects the performance of SD, the SD method is shown to have close performance with the LMMSE detection approach. In addition, RC-Struct is shown to outperform RCNet across all evaluated E_b/N_o values. This is because the equalization after RC works as the nearest neighbor detector. RC-Struct, instead, works as a non-linear detector, making it better to classify the data. We will discuss this with more details in Section V-E.
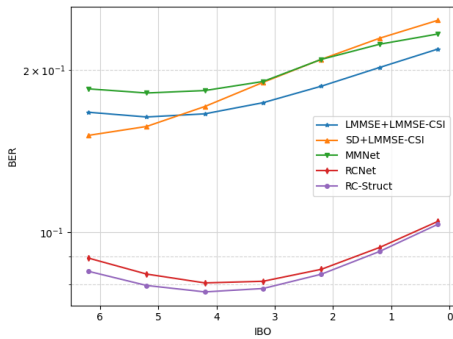
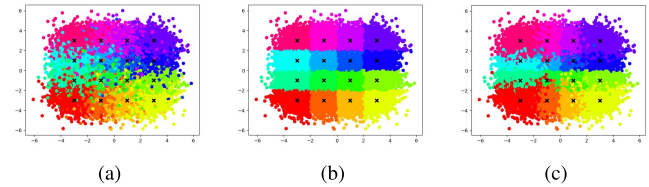Fig. 5.   Comparison of BER in the non-linear region.

Fig. 6.   Constellation classification for the RC processed received symbols at $E_b/N_o = 15$ dB with 16-QAM modulation. (a) Colored with ground truth labels (b) Colored with RC equalized labels (c) Colored with RC-Struct predicted labels.

While MMNet has shown impressive performance in [6], it is noteworthy that 500 pilot symbols with perfect CSIs are utilized for the underlying training, which is different from the online setup used in this paper. In this paper, we conduct the detection subframe by subframe, which is an online over-the-air scenario. The number of pilot symbols is set as 4 per subframe, which is far less than the setting in [6]. Note that the total number of trainable parameters in MMNet is around 40K, while the number of trainable parameters in RC-Struct is around 7K. As MMNet has more learnable weights, it is shown to be less effective than RC-Struct when learned with only a limited number of training data, as shown in Fig. 4. In addition, the BER of MMNet does not improve much as E_b/N_o increases. As a large number of weights need to be learned, the MMNet is more likely to suffer from overfitting in the high E_b/N_o regime when the training data is limited.

D. Comparison of Strategies Under System Non-Linearity

To investigate the performance of the approaches under system non-linearity, we incorporate the following PA model in the evaluation [37]:

$$\phi(x) = \frac{x}{\left[1 + \left(\frac{|x|}{x_{\text{sat}}}\right)^{2\rho}\right]^{0.5\rho}}, \tag{20}$$

where x is the input transmission signal, ρ represents the smoothing parameter, and x_{sat} measures the saturation level. The function indicates that when $|x| \ll x_{\text{sat}}$, the $\phi(x)$ is approximately equal to x, which forms the linear region without distortion. When it comes to the region where x approximates x_{sat}, the function becomes non-linear and the input signal is distorted. In a nutshell, the distortion occurs when the peak-to-average-power-ratio (PAPR) of the input signal is higher than the input back-off (IBO), where the IBO is the ratio between PA's saturation power to the input power. In this paper, the parameters are set as $x_{\text{sat}} = 1$ and $\rho = 3$. As PAPR is controlled in the range of 6 dB to 9 dB, the non-linear region is chosen by setting IBO smaller than 6.5 dB.

In Fig. 5, we compare BER for different approaches when PA non-linear distortion occurs. Fig. 5 demonstrates that the RC-based approaches can effectively combat the non-linearity of the PA. The RC-Struct continues showing advantages over all other methods. In the low IBO regime, the performance

gap between RC-Struct and RCNet becomes smaller. This is because the low IBO makes the transmitted signal severely distorted, which affects the LMMSE-based shifting parameter and results in the poor performance of RC-Struct. The MMNet collapses as it models system without non-linearity and is affected by inaccurate LMMSE-based CSI in non-linear region.

E. Effectiveness of Structure-Based Neural Network

In this section, we discuss the effectiveness of the classification network in the frequency domain and analyze the performance comparison with more details. For ease of discussion, we choose to analyze the classification results with 16-QAM modulation at 15 dB and 5 dB E_b/N_o for the 4×4 MIMO-OFDM system without rank and link adaptation. Note that RCNet processes the received signals with the RC-based time domain data-stream decoupling and equalization, following with a frequency domain nearest-neighbor classification. RC-Struct replaces the nearest-neighbor classification with a frequency domain structure-based NN.

In Fig. 6 (a), we show the signal constellation of the frequency domain received symbols after RC-based time domain data-stream decoupling and equalization. Each color in the plot represents a particular constellation class (16 constellation classes in total) and the cross-marks represent the original position of the transmitted signal constellation point. Fig. 6 (b) shows the frequency domain decision boundary of RCNet after RC-based decoupling and equalization in the time domain and nearest-neighbor classifier in the frequency domain. The nearest-neighbor method classifies the received signal constellation to the closest transmitted signal constellation. Comparing Fig. 6 (a) and (b), we can see that the ground truth labels for the received symbols are not necessarily scattered in the nearest region of the corresponding transmitted constellation point and there are still noise within the received symbols. By applying the frequency domain NN, RC-Struct introduces non-linear decision boundaries in the classification processes, as shown in Fig. 6 (c). Although the decision boundaries are still not perfectly aligned with the ground truth decision boundaries, introducing more non-linearity in the classifier has made the decision boundaries closer to the ground truth ones.

When the E_b/N_o decreases to 5 dB, the equalized received signals are noisier as shown in Fig. 7 (a). It is clear that received symbols cannot be well separated by the linear boundaries, leading to a poor performance of RCNet where
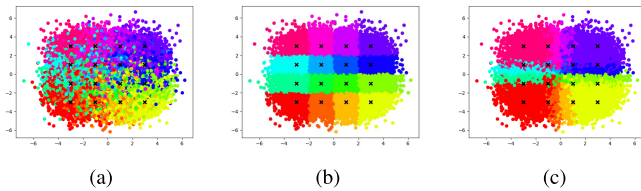
Fig. 7.   Constellation classification for the RC processed received symbols at $E_b/N_o = 5$ dB with 16-QAM modulation. (a) Colored with ground truth labels (b) Colored with RC equalized labels (c) Colored with RC-Struct predicted labels.
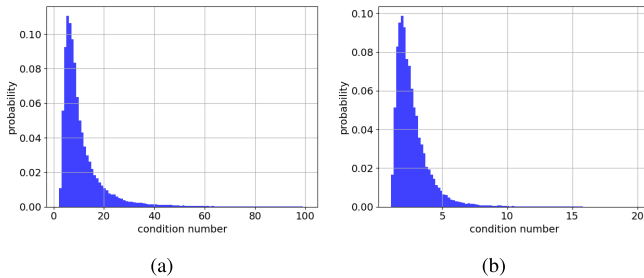

Fig. 8.   Histogram of channel condition number. (a) before rank adaptation. (b) after rank adaptation.

the decision boundary between classes is linear. As exhibited in Fig. 7 (c), RC-Struct learns more noise-tolerable boundary lines to separate data and is shown to be more effective than the linear separation of RCNet in Fig. 4. However, both methods are affected by the noise in the low E_b/N_o regime, resulting in high BER performance.

F. Performance Comparison With Transmission Adaptation

Due to the channel variation and user mobility, the condition number of the underlying MIMO channel in a mobile broadband network such as 4G and 5G/5G-Advanced is also changing rapidly. Transmitting with full rank and fixed modulation constellations will result in poor system performance. In Fig. 8, we show the channel condition numbers before and after rank adaptation for a duration of 100 milliseconds of a 3GPP 3D MIMO channel [32]. Without rank adaptation, the condition number is calculated as the ratio of the largest singular value of the MIMO channel in the frequency domain to its smallest singular value. With rank adaptation, the condition number is calculated based on the effective MIMO channel after precoding. As shown in Fig. 8, without rank adaptation, most of the MIMO channels have a relatively high condition number. However, after rank adaptation, the underlying effective MIMO channels have much smaller condition numbers. This is the reason why all detection strategies presented in Fig. 4 are shown to have higher BERs, when MIMO transmission adaptation is not adopted. In fact, this is also why link adaptation and rank adaptation are introduced in 4G and 5G/5G-Advanced networks as important features to support MIMO communications [30].

In this section, we analyze the performance of various symbol detection methods under link and rank adaptation. It is important to note that this is a critical step to evaluate

the promise of NN-based symbol detection in realistic and meaningful mobile broadband networks. To demonstrate the effectiveness of link adaptation and rank adaptation, we provide performance analysis when link adaptation and rank adaptation are utilized separately. We can show that our RC-based approaches preserve their advantages in all cases as they do not rely on explicit system modeling. The procedures of link and rank adaptation are detailed in the Appendix.

In the experiments, we follow current 4G/5G standards to conduct the corresponding link and rank adaptation. Specifically, the link adaptation will modulate the data streams using either QPSK, 16-QAM, or 64-QAM following 4G/5G standards [7]. The modulation order is adjusted in a wideband fashion and the reference SINR is referred to [38]. For the rank adaptation, the precoding matrix is obtained by conducting the singular value decomposition (SVD) of the LMMSE estimated channel matrix. The rank of the MIMO transmission is adjusted across all the subcarriers complying with the wideband-based rank indicator (RI) feedback [39], and the precoding matrix is applied in a group of subcarriers following the procedure of the subband-based precoding matrix indicator (PMI) feedback [39]. The sub-band size is set as 84 subcarriers, which is 7 RBGs. The detailed procedure is described in the Appendix.

When rank adaptation is conducted, the RawBER for all the methods decreases compared with the case without utilizing adaptation, as shown in Fig. 9 (a). Note that in this case, RawBER is equivalent to BER since link adaptation is not utilized. RC-Struct continues demonstrating its advantage over all other methods. These results clearly underline the effectiveness of RC-Struct in both good and bad channel conditions. For LMMSE, SD, and MMNet, the RawBER at 3 dB E_b/N_o is lower than 6 dB E_b/N_o. As illustrated in Fig. 10, the channels at 0 dB and 3 dB E_b/N_o are all adapted to rank 2, while those at 6 dB have a high percentage to be adapted to rank 3. Therefore, the RawBER curve for these three methods from 0 dB to 3 dB E_b/N_o is decreasing since the adapted rank is the same, while the curve increases from 3 dB to 6 dB E_b/N_o as the channels have a higher percentage to be adapted to higher ranks. When link adaptation is applied, RC-Struct preserves its advantage over the other methods, as shown in Fig. 9 (b).

When employing both link and rank adaptation in the system, the performance of all the methods has been improved, as shown in Fig. 9 (c). The performance improvement is because the transmission rank and modulation order are dynamically adapted, as presented in Fig. 10. The dynamically adapted rank and modulation order, on the other hand, leads to the zigzag pattern of the performance curve, as lower transmission rank and modulation order may be adopted in lower E_b/N_o regimes. RC-Struct still achieves the best performance among all the compared methods and demonstrates its ability to be applied with dynamic transmission modes. Furthermore, we conducted an experiment when LDPC coding is adopted with link and rank adaptation at 15 dB Eb/N_o. In 3GPP 5G NR [40], the code rate ranges from 0.0762 to 0.9258. In our experiment, the code rate for LDPC coding is set as 0.3125. Tab. III shows that the BER is 1.84% and BLER is 6.45% after LDPC. This result demonstrates that
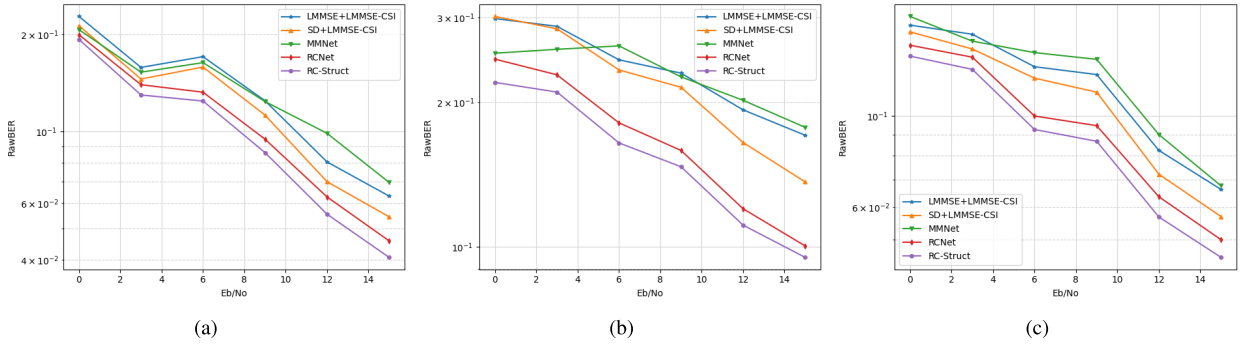
Fig. 9. Comparison of RawBER with adaptation. (a) rank adaptation only. (b) link adaptation only. (c) rank and link adaptation.
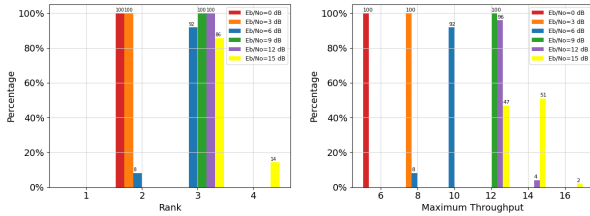

Fig. 10. Percentage of adapted rank and capacity for all the tested channels.

TABLE III

PERFORMANCE WHEN ADOPTED LDPC CHANNEL
CODING AT 15 DB E_b/N_o

	Coded BER	BLER
RC-Struct	1.84%	6.45%

RC-Struct can meet the target BLER of 10% as specified in 3GPP 5G NR [40].

Here are the summaries and takeaways for the experiments.

- When no adaptation is applied, RC-Struct outperforms all existing methods. It is shown to have lower BER than RCNet, due to its ability to handle non-linear boundaries in the frequency domain.

- For systems with unknown PA non-linearity, RC-Struct and RCNet shows relatively large performance gains over other methods. In the low IBO regime, the performance gap between RC-Struct and RCNet becomes smaller since the severely distorted signal negatively affects the estimation of the shifting parameter.

- When rank adaptation and link adaptation are conducted, RC-Struct and RCNet can effectively conduct online learning with extremely limited training symbols to adjust detection strategies on a subframe-by-subframe basis. Meanwhile, RC-Struct continues showing advantages over other methods including RCNet. When LDPC coding is adopted, RC-Struct has a BLER of 6.45%, which is lower than the target BLER specified in [40].

VI. CONCLUSION

In this paper, we introduced an NN-based symbol detector for MIMO-OFDM systems, which incorporates the structural knowledge of the systems, including the temporal dynamics within the data, the time-frequency structure of the OFDM waveform, and the repetitive structure of the modulation constellation. The temporal information is captured by the RC network in the time domain, while the constellation symmetry is leveraged by the classifier in the frequency domain. The network architecture allows the network to be learned with significantly reduced training overhead and can be applied with rank adaptation and link adaptation of the underlying MIMO transmissions. Experiments indicate the effectiveness of the introduced RC-Struct network and demonstrate the advantages of incorporating the structure information into the network under MIMO channels with time-dynamic transmission modes. Due to the ability to be applied in practical MIMO operations, RC-Struct provides a promising symbol detection approach for the MIMO-OFDM system in the 5G/5G-Advanced and Beyond networks.

APPENDIX A
RANK ADAPTATION AND LINK ADAPTATION

A. Rank Adaptation

A key premise of 5G/5G-Advanced system is its ability to dynamically switch between different ranks to efficiently exploit the bandwidth for a given channel condition and improve the performance of the system, since there is no single mode that works best in all channel conditions [41]. The number of transmitted data streams is referred to as the transmission rank. We focus on the capacity based rank adaptation to adjust the rank of the wireless channel.

The precoding process for L data streams transmission can be represented as $\boldsymbol{X}_n(k) = \boldsymbol{Q}_n(k)\boldsymbol{S}_n(k)$, where $\boldsymbol{Q}_n(k) \in \mathbb{C}^{N_t \times L}$ is the unitary precoding matrix and $\boldsymbol{S}_n(k) \in \mathbb{C}^{L \times 1}$ is the effective transmitted symbols. Suppose the precoding matrix is $\boldsymbol{Q}_n(k) = \boldsymbol{V}_n^L(k)$, where $\boldsymbol{V}_n^L(k)$ is first L columns from the unitary matrix $\boldsymbol{V}_n(k)$ in the SVD $\boldsymbol{H}_n(k) = \boldsymbol{U}_n(k)\boldsymbol{\Lambda}_n(k)\boldsymbol{V}_n(k)^H$. The throughput of L data streams transmission can be written as [42]

$$C_L = \sum_{l=1}^{L} \log_2\left(1 + \frac{P_t}{L\sigma^2}\lambda_l^2\right), \qquad (21)$$

where P_t is the total transmit power and λ_l is the lth singular value in the SVD of $\boldsymbol{H}_n(k)$ with the order of

$\lambda_{max} = \lambda_1 \geq \cdots \geq \lambda_{min} \geq 0$. The rank is adapted to take the maximum throughput among all the possible L values ($L \leq N_t$).

B. Link Adaptation

In 5G wireless communication network, multiple modulation and coding schemes (MCS) are enabled to transmit with higher data rates and reliability. The transmitter is expected to transmit data with a proper MCS according to the channel conditions. Link adaptation techniques select MCS for wireless transmission based on the channel quality indicator (CQI) feedback. In a simple case, the base station (BS) transmits pilot signals to user equipment (UE) at given OFDM subcarrier positions. The UE measures the SINR at each OFDM subcarrier and calculates an effective SINR using techniques such as effective exponential SNR mapping (EESM) [43].

The EESM approach has been widely applied to the OFDM link layers. It maps individual subcarrier SINRs to an effective SINR with the following equation:

$$SINR_{eff} = -\beta ln \left[\frac{1}{S} \sum_{n=1}^{S} e^{-\frac{SINR_n}{\beta}} \right], \qquad (22)$$

where $SINR_n$ is the nth subcarrier SINR and S represents the size of the subband. The parameter β is empirically obtained and is calibrated to fit the model for different MCS level. The effective SINR is compared with the reference SINR value and mapped to a CQI value, which indicates the highest modulation order and code rate for keeping a Packet Error Rate (PER) below 10% [38]. Once the CQIs are collected by the base station, it allocates resources for each user.

REFERENCES

[1] J. Xu, Z. Zhou, L. Li, L. Zheng, and L. Liu, "RC-Struct: Reservoir computing meets knowledge of structure in MIMO-OFDM," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2021, pp. 1–5.
[2] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. Zhang, "Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 212–217, Apr. 2020.
[3] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
[4] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.
[5] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 584–588.
[6] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020.
[7] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation*, Standard TS 36.211, Rev. 16.6.0, 3GPP, 2021.
[8] X. Lyu, W. Feng, and N. Ge, "Deep neural network-based symbol detection for highly dynamic channels," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
[9] N. Farsad and A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018.
[10] Y. Liao, N. Farsad, N. Shlezinger, Y. C. Eldar, and A. J. Goldsmith, "Deep neural network symbol detection for millimeter wave communications," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
[11] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets MIMO-OFDM," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4694–4708, Oct. 2018.
[12] Z. Zhou, L. Liu, and H.-H. Chang, "Learning for detection: MIMO-OFDM symbol detection through downlink pilots," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3712–3726, Jun. 2020.
[13] Z. Zhou, L. Liu, V. Chandrasekhar, J. Zhang, and Y. Yi, "Deep reservoir computing meets 5G MIMO-OFDM systems in symbol detection," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, Apr. 2020, pp. 1266–1273.
[14] Z. Zhou, L. Liu, S. Jere, J. Zhang, and Y. Yi, "RCNet: Incorporating structural information into deep RNN for online MIMO-OFDM symbol detection with limited training," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3524–3537, Jun. 2021.
[15] L. Li, L. Liu, J. C. Zhang, J. D. Ashdown, and Y. Yi, "Reservoir computing meets Wi-Fi in software radios: Neural network-based symbol detection using training sequences and pilots," in *Proc. 29th Wireless Opt. Commun. Conf. (WOCC)*, May 2020, pp. 1–6.
[16] Z. Zhou, S. Jere, L. Zheng, and L. Liu, "Learning with knowledge of structure: A neural network-based approach for MIMO-OFDM detection," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, Nov. 2020, pp. 22–26.
[17] Z. Zhao, M. C. Vuran, F. Guo, and S. D. Scott, "Deep-waveform: A learned OFDM receiver based on deep complex-valued convolutional networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2407–2420, Aug. 2021.
[18] M. Honkala, D. Korpi, and J. M. J. Huttunen, "DeepRx: Fully convolutional deep learning receiver," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3925–3940, Jun. 2021.
[19] M. Goutay, F. A. Aoudia, and J. Hoydis, "Deep hypernetwork-based MIMO detection," in *Proc. IEEE 21st Intl. Wkshps Signal Process. Adv. Wireless Commun. (SPAWC)*, May 2020, pp. 1–5.
[20] X. Yi and C. Zhong, "Deep learning for joint channel estimation and signal detection in OFDM systems," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2780–2784, Dec. 2020.
[21] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," German Nat. Res. Center Inf. Technol. GMD, Bonn, Germany, Tech. Rep., 2001, vol. 148, no. 34, p. 13.
[22] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004.
[23] G. Tanaka *et al.*, "Recent advances in physical reservoir computing: A review," *Neural Netw.*, vol. 115, pp. 100–123, Jul. 2019.
[24] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks Trade*. Berlin, Germany: Springer, 2012, pp. 659–686.
[25] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 2307–2315.
[26] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Reservoir-based techniques for speech recognition," in *Proc. IEEE Int. Joint Conf. Neural Netw. Proc.*, Jul. 2006, pp. 1050–1053.
[27] A. Jalalvand, G. Van Wallendael, and R. Van De Walle, "Real-time reservoir computing network-based systems for detection tasks on visual contents," in *Proc. 7th Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jun. 2015, pp. 146–151.
[28] Z. Tong and G. Tanaka, "Reservoir computing with untrained convolutional neural networks for image recognition," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1289–1294.
[29] R. Shafin *et al.*, "Realizing green symbol detection via reservoir computing: An energy-efficiency perspective," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[30] L. Liu, R. Chen, S. Geirhofer, K. Sayana, Z. Shi, and Y. Zhou, "Downlink MIMO in LTE-advanced: SU-MIMO vs. MU-MIMO," *IEEE Commun. Mag.*, vol. 50, no. 2, pp. 140–147, Feb. 2012.
[31] 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 Propag.*, vol. 62, no. 6, pp. 3242–3256, Jun. 2014.
[32] *Study on 3D Channel Model for LTE*, document TR 36.873., 3GPP, 2015.
[33] Z. Zhou, L. Liu, J. Xu, and R. Calderbank, "Learning to equalize OTFS," 2021, *arXiv:2107.08236*.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th intl. conf. Artif. intell. Statist.*, 2010, pp. 249–256.

[35] A. Ghasemmehdi and E. Agrell, "Faster recursions in sphere decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3530–3536, Jun. 2011.

[36] F. Peng, J. Zhang, and W. E. Ryan, "Adaptive modulation and coding for IEEE 802.11 n," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2007, pp. 656–661.

[37] C. Rapp, "Effects of HPA-nonlinearity on a 4-DPSK/OFDM-signal for a digital sound broadcasting signal," *ESA Special Publication*, vol. 332, pp. 179–184, Oct. 1991.

[38] *Physical Layer Aspects for Evolved Utra (Release 7)*, document TSG-RAN. TR 25.814, 3Gpp, 2006.

[39] *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures*, Standard TS 36.213, Rev. 16.6.0, 3GPP, 2021.

[40] *5G; NR; Physical Layer Procedures for Data*, document TS 38.214, Rev. 16.6.0, 3GPP, 2021.

[41] M. U. Sheikh, R. Jagusz, and J. Lempiäinen, "Performance evaluation of adaptive MIMO switching in long term evolution," in *Proc. 7th Intl. Wireless Commun. Mobile Comput. Conf.*, Jul. 2011, pp. 866–870.

[42] M. Zhang, M. Shafi, P. J. Smith, and P. A. Dmochowski, "Precoding performance with codebook feedback in a MIMO-OFDM system," in *Proc. IEEE Intl. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–6.

[43] R. Sandanalakshmi, T. G. Palanivelu, and K. Manivannan, "Effective SNR mapping for link error prediction in OFDM based systems," in *Proc. IET-U.K. Int. Conf. Inf. Commun. Technol. Electr. Sci. (ICTES)*, 2007, pp. 684–687.

Lianjun Li received the B.S. degree in telecommunications engineering from Zhejiang University, Hangzhou, China, and the M.S. degree in electrical engineering from The University of Texas at Dallas, Richardson, TX, USA. He is currently pursuing the Ph.D. degree with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA.

After receiving his B.S. degree, he joined Ericsson, China, as a Wireless Network Optimization Engineer for seven years. His research interests are applying reinforcement learning and deep learning-based techniques to wireless communication systems.

Lizhong Zheng (Fellow, IEEE) received the B.S. and M.S. degrees from Tsinghua University, Beijing, China, in 1994 and 1997, respectively, and the Ph.D. degree from the University of California, Berkeley, CA, USA, in 2002. Since 2002, he has been with the Department of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA, where he is currently a Professor of electrical engineering and computer sciences. His research interests include information theory, wireless communications, and statistical inference. He was a recipient of the Eli Jury Award from UC Berkeley in 2002, the IEEE Information Theory Society Paper Award in 2003, the NSF CAREER Award in 2004, and the AFOSR Young Investigator Award in 2007.

Jiarui Xu received the B.S. degree in telecommunication engineering from the Beijing Institute of Technology, Beijing, China, in 2017, and the M.S. degree in electrical and computer engineering from the University of Michigan, Ann Arbor, MI, USA, in 2019. She is currently pursuing the Ph.D. degree with the Bradley Department of Electrical and Computer Engineering (ECE), Virginia Tech, Blacksburg, VA, USA. Her research interests include deep learning, machine learning, and their applications in wireless communications.
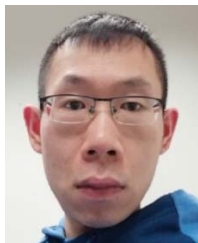
Zhou Zhou received the B.S. degree in communications engineering from the University of Electronic Science and Technology of China (UESTC) in 2011. Since 2018, he has been with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, as a Research Assistant. His current research interests are in the broad area of neural networks, machine intelligence, and wireless communications.

Lingjia Liu (Student Member, IEEE) received the B.S. degree in electronic engineering from Shanghai Jiao Tong University and the Ph.D. degree in electrical and computer engineering from Texas A&M University.

He is currently a Professor with the ECE Department, Virginia Tech (VT), and also the Associate Director of Wireless@VT. Prior to joining VT, he was an Associate Professor with the EECS Department, The University of Kansas (KU). He spent more than four years working at the Mitsubishi Electric Research Laboratory (MERL) and the Standards and Mobility Innovation Laboratory, Samsung Research America (SRA), where he received the Global Samsung Best Paper Award in 2008 and 2010. He was leading Samsung's efforts on multiuser MIMO, CoMP, and HetNets in LTE/LTE-Advanced standards. His general research interests lie in emerging technologies for beyond 5G cellular networks, including machine learning for wireless networks, massive MIMO, massive MTC communications, and mmWave communications. He received the Air Force Summer Faculty Fellowship from 2013 to 2017, the Miller Scholarship at KU in 2014, the Miller Professional Development Award for Distinguished Research at KU in 2015, the 2016 IEEE GLOBECOM Best Paper Award, the 2018 IEEE ISQED Best Paper Award, the 2018 IEEE TAOS Best Paper Award, the 2018 IEEE TCGCC Best Conference Paper Award, the 2020 Charles K. Kao Best Paper Award, and the 2021 VT College of Engineering Dean's Award for Excellence in Research.