RCNet: Incorporating Structural Information Into Deep RNN for Online MIMO-OFDM Symbol Detection With Limited Training

Zhou Zhou, Lingjia Liu, *Senior Member, IEEE*, Shashank Jere, Jianzhong Zhang, *Fellow, IEEE*, and Yang Yi, *Senior Member, IEEE*

***Abstract*— In this paper, we investigate online learning-based MIMO-OFDM symbol detection strategies focusing on a special recurrent neural network (RNN) – reservoir computing (RC). We first introduce the Time-Frequency RC to take advantage of the structural information inherent in OFDM signals. Using the time domain RC and the time-frequency RC as building blocks, we provide two extensions of the shallow RC to RCNet: 1) Stacking multiple time domain RCs; 2) Stacking multiple time-frequency RCs into a deep structure. The combination of RNN dynamics, the time-frequency structure of MIMO-OFDM signals, and the deep network enables RCNet to handle the interference and nonlinear distortion of MIMO-OFDM signals to outperform existing methods. Unlike most existing NN-based detection strategies, RCNet is also shown to provide a good generalization performance even with a limited online training set (i.e, similar amount of reference signals/training as standard model-based approaches). Numerical experiments demonstrate that the introduced RCNet can offer a faster learning convergence and as much as 20% gain in bit error rate over a shallow RC structure by compensating for the nonlinear distortion of the MIMO-OFDM signal, such as due to power amplifier compression in the transmitter or due to finite quantization resolution in the receiver.**

***Index Terms*— Deep learning, online learning, OFDM-MIMO, symbol detection, reservoir computing.**

I. Introduction

ARTIFICIAL-INTELLIGENCE (AI) enabled cellular networks are envisioned as the critical path for Beyond-5G networks [2], [3]. Among the various potential fields of communication systems where AI and its associated machine learning tools can contribute, symbol detection is a very important area in the physical layer. To be specific, symbol detection constitutes a key module within the signal processing chain of modern communication receivers. Assuming the availability of receiver channel state information (CSI), the optimal model-based strategy is to apply the maximum likelihood detector. However, the performance of model-based strategies is sensitive to model inaccuracies and CSI estimation errors. On the other hand, learning-based approaches can provide robust performance without relying on detailed underlying channel models. In this paper, we focus on the problem of symbol detection in MIMO-OFDM which is the major radio access technology for 4G/5G systems [4]. In current 4G/5G systems, symbol detection methods are based on modeling the underlying wireless link and applying associated model-based signal processing techniques [5]. However, in the presence of non-linearities, either due to the underlying wireless channels (e.g. mmWave and Terahertz channels for Beyond-5G) or device components (e.g. power amplifier), it becomes extremely difficult to analytically model such behavior in a tractable and accurate manner. Alternatively, recent work in [6], [7] demonstrated the effectiveness of using neural networks (NNs) for symbol detection under unknown environments. Along the same line of thinking, we consider exploiting the dynamic behavior of recurrent neural networks (RNNs) for the task of MIMO-OFDM symbol detection. The main motivation of adopting RNNs instead of other NN architectures is based on the fact that under fairly mild and general assumptions, RNNs are universal approximations of dynamic systems [8]. This is extremely important for wireless systems which are highly dynamic over time and frequency. On the other hand, to realize the full potential of RNNs, especially deep RNNs, new research challenges need to be addressed for MIMO-OFDM symbol detection:

- **Challenge 1**: From a system design perspective, training an NN-based symbol detector using over-the-air feedback—to update layer weights of the underlying NN based on the back-propagation algorithm—is likely prohibitively expensive in terms of the control overhead.[1]

Manuscript received January 22, 2020; revised July 2, 2020 and October 18, 2020; accepted January 4, 2021. Date of publication January 21, 2021; date of current version June 10, 2021. The work of Zhou Zhou, Lingjia Liu, Shashank Jere, and Yang Yi was supported in part by the U.S. National Science Foundation under Grant CCF-1937487 and Grant ECCS-1811497. This article was presented in part at the 34th AAAI Conference on Artificial Intelligence. The associate editor coordinating the review of this article and approving it for publication was J. Hoydis. *(Corresponding author: Lingjia Liu.)*

Zhou Zhou, Lingjia Liu, Shashank Jere, and Yang Yi are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24060 USA (e-mail: ljliu@ieee.org).

Jianzhong Zhang is with the Standards and Mobility Innovation Laboratory, Samsung Research America, Mountain View, CA 94043 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3051317.

Digital Object Identifier 10.1109/TWC.2021.3051317

[1] For example, in 3GPP LTE/LTE-Advanced systems, the reference signal overhead is specified and is usually fixed for different MIMO configurations [9]: The training set (demodulation reference signals) for SISO-OFDM is around 5% of all the resource elements.On the other hand, for a 2×2 MIMO-OFDM system, the overhead for reference signals is around 10%. In 5G, more flexible reference signal design is designed to reduce the reference signal overhead [10]

Therefore, over-fitting could occur when the selected NN model is too complicated.

- **Challenge 2**: NNs, especially RNNs, are mainly designed to process time-domain data. Since data is transmitted in both time and frequency domains in contemporary cellular systems, it is critical to combine RNNs with domain knowledge in an organic way to offer reliable and robust performance gains over current communication strategies.
- **Challenge 3**: The underlying wireless environment changes dynamically over time and frequency. This would be especially true for Beyond-5G systems which would mainly use mmWave and Terahertz channels. Accordingly, the underlying NN model needs to be sophisticated enough to capture the time-frequency variation of the channel. Otherwise, under-fitting can result in poor symbol detection performance.

Furthermore, a general challenge of neural networks to wireless systems is "uncertainty in generalization" [2]—it is often unclear whether the dataset used for training the underlying neural network is general enough to capture the distribution of data encountered in reality. This is especially true for 5G and Beyond 5G networks that will provide reliable service under vastly different scenarios and environments. In 4G/5G MIMO-OFDM systems, there exist many MIMO modes with link adaptation, rank adaptation, and scheduling operating on a subframe basis [11]. Therefore, it is challenging, if not impossible, to adopt a complete offline training-based approach. Rather, it is critical to design an online neural-network-based approach to conduct symbol detection in each subframe only using the limited training symbols that are contained in that particular subframe. In this way, the online-learning-based approach can be adaptive and robust to the change of operation modes, channel distributions, and environments.

A. Our Contributions

To address these challenges, our approach is based on reservoir computing (RC) [12] which is a special category of RNNs. RC is capable of avoiding the issues of vanishing and exploding gradients which occur during training of conventional RNNs using back-propagation through time (BPTT) [13]. Furthermore, the training of RC is only conducted on the output layer while the hidden layers and the input layers are fixed according to a certain distribution. In this way, the amount of training needed for MIMO-OFDM symbol detection can be significantly reduced making it an ***operationally feasible*** solution to address *Challenge 1*. This benefit can be clearly seen in Section V-D where we show a quantitative comparison of the training overhead for various learning-based strategies. The RC architecture makes applying NN techniques in the physical layer of cellular networks possible and feasible. It is also shown in [14] that RC-based symbol detection can significantly improve the underlying energy-efficiency of the system.

In this work, rather than directly applying the shallow RC structure, we attempt to address *Challenge 2* and *Challenge 3* by introducing *RCNet* through the following extensions to facilitate ***deep*** RC-based symbol detection methods that further improve detection performance by merely using over-the-air training, that is, **online training**:

- Extend the output layer of a shallow RC structure to a multiple-layer network promoting joint time-frequency processing. This method can effectively resolve *Challenge 2* by incorporating the structural information (time-frequency structure) of MIMO-OFDM signals into the output layer design of RCNet.
- Stack shallow RCs together into a "deep" RC to improve the processing capability of RCNet. In this way, *Challenge 3* can be addressed owing to the boosting mechanism of NNs.

The first extension on deepening the output layer is achieved by replacing the original single layer output of the shallow RC with a three-layer structure: a time-domain layer, a Fourier transform layer, and a frequency-domain layer, namely the "time-frequency RC". The time-domain layer attempts to reconstruct the transmitted time-domain signal. The Fourier transform layer is used to transform the time-domain signal to the frequency-domain. The frequency-domain layer attempts to extract frequency-domain features to further improve the detection performance. The second extension is achieved by concatenating multiple "time-frequency RCs" sequentially. Note that the output weights of each RC layer are trained in a consecutive fashion.

Through extensive experiments, we show that this deep structure of RCNet demonstrates appealing and robust performance when non-linear effects exist in the end-to-end wireless system. In other words, it outperforms conventional MIMO symbol detection strategies under channel non-linearities caused, for e.g., by the power amplifier (PA) at the transmitter (Tx) or the quantization error due to the low resolution of analog-to-digital converters (ADCs) at the receiver (Rx). The results suggest that the introduced MIMO-OFDM symbol detection framework can be a very promising enabling technology for Beyond-5G cellular systems where high-frequency spectrum and low-resolution ADCs would be frequently used. To the best of our knowledge, this is the first recurrent neural network-based MIMO-OFDM symbol detector in the literature that can provide good symbol detection performance using a limited online training set under relevant channel environments (e.g., the WINNER II channel) in the presence of practical real-world constraints such as transmit-side non-linearity due to the PA and limited quantization resolution in the receiver.

B. Related Work

1) Deep RNN: Deep neural networks (DNN) can extract sophisticated features thereby providing improved classification performance over shallow NNs [15]. Hierarchical RNNs are shown to be capable of learning long-term dependencies of signals [16]. In [17], the deep long short-term memory network was introduced by consecutively stacking the hidden layers of multiple RNNs. This deep structure is shown to significantly improve performance in the task of speech recognition. The methods of extending a shallow RNN to deep RNNs have been summarized in [18]: multi-layer RNNs can be constructed by increasing input layers, hidden layers and output layers, as well as stacking multiple shallow RNNs into a deep form. Concatenating echo state networks (ESNs) into a chain by learning readout layers connecting to each ESN layer is introduced in [19]. The follow up work in [20], [21] extended this structure into a deep version which is demonstrated to be able to achieve a higher memory compared to the shallow one. However, these extensions are very general and do not consider specific domain knowledge and structural information for targeted applications.

2) NN-Based Symbol Detection: The work in [7] shows that an unfolding NN from the projected gradient descent algorithm can be trained to achieve state-of-the-art performance for MIMO symbol detection tasks. In [22], an online and offline combined approach is introduced for OFDM symbol detection. In [23], a soft-thresholding based neural network structure is applied for MIMO symbol detection in a spatial correlated channel. Note that [7], [22] combine online and offline training for symbol detection tasks, where the online training overheads can be reduced through leveraging the same statistical features from the offline training dataset. However, the symbol detection performance will deteriorate when the training dataset is statistically different from the testing ones. In contrast, we consider a purely online learning-based approach that only utilizes the limited training dataset available within each subframe for MIMO-OFDM symbol detection to mitigate the issue of "uncertainty in generalization" for robust and adaptive communications.

Our previous work [6], [24], showed that shallow RCs can achieve good performance for online MIMO-OFDM symbol detection even with a very limited training set. In [6], an ESN, which is a special variant of RC, is introduced as an OFDM symbol detector without relying on obtaining explicit CSI. This scheme is evaluated under relevant scenarios for cellular networks where the training/reference signal overhead is comparable to that of current cellular networks. In our follow-up work, the windowed ESN (WESN) is introduced by adding a sliding window to the input of ESN to enhance the short-term memory (STM) [24]. Experimental results show that WESN can provide good performance over standard ESNs using the training/reference signal set adopted in 4G LTE-Advanced [4]. Furthermore, ESN-based symbol detectors can effectively compensate for the distortion caused by non-linear components in wireless transmission.

The organization of this paper is as follows. In Sec. II, we briefly describe the transceiver architecture. In Sec. III and IV, we introduce the two extensions incorporated in our proposed RCNet structure and its associated learning algorithms. Sec. V evaluates the performance of RCNet as opposed to existing symbol detection strategies for MIMO-OFDM systems. The conclusion and future work is contained in Sec. VI.

II. OFDM Transceiver Architecture and Structural Information

In this section, we briefly introduce the transceiver architecture of a MIMO-OFDM system and illustrate the structural information we are going to utilize for the design of RCNet. The OFDM resource grid can be seen most clearly in Fig. 1. The total system bandwidth is divided into N_{sc} subcarriers. In each sub-frame, the first Q OFDM symbols are the reference signals (the training set) and the rest N_d OFDM symbols are used to carry data (the testing set). Therefore, the reference signal overhead (training overhead) is defined as $\eta = Q/(Q + N_d)$. In 3GPP LTE/LTE-Advanced and 5G systems, $Q+N_d = 14$ for normal sub-frames and the overhead is typically below 20% to improve spectral-efficiency [9], [10]. For a single MIMO-OFDM symbol, we denote the modulation symbol as $\{\boldsymbol{z}(n)\}_{n=0}^{N_{sc}-1}$, where $\boldsymbol{z}(n) \in \mathbb{C}^{N_t \times 1}$ represents the modulation symbols at the nth sub-carrier of the underlying MIMO-OFDM symbol in the **frequency domain**; N_t is the number of transmit antennas in the system. Each element of $\boldsymbol{z}(n)$ is modulated using quadrature amplitude modulation (QAM). A single MIMO-OFDM symbol in the frequency domain also can be lumped as a matrix,

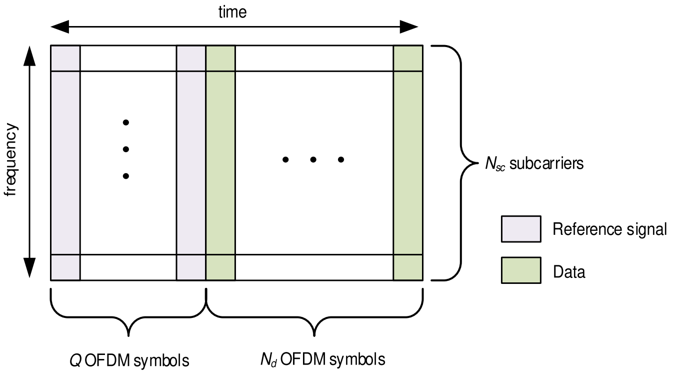


Fig. 1. OFDM Resource Grid.

$$\boldsymbol{Z} \triangleq [\boldsymbol{z}(0), \boldsymbol{z}(1), \cdots, \boldsymbol{z}(N_{sc}-1)]^T. \tag{1}$$

After the inverse fast Fourier transform (IFFT) operation and addition of the cyclic prefix (CP), we can obtain the **time domain** MIMO-OFDM symbol, denoted as $\tilde{\boldsymbol{X}}$. At the transmitter, the MIMO-OFDM symbol is distorted by a non-linear activation function $f(\cdot)$ due to the inherent non-linearity of transmitter-side radio circuits, such as the PA [25]. At the receiver, a single MIMO-OFDM symbol is expressed as

$$\boldsymbol{X} = q(h(f(\tilde{\boldsymbol{X}})) + \boldsymbol{N}), \tag{2}$$

where $\boldsymbol{X} \triangleq [\boldsymbol{x}(0), \boldsymbol{x}(1), \cdots, \boldsymbol{x}(N_{sc} + N_{cp} - 1)]^T \in \mathbb{C}^{(N_{sc}+N_{cp})\times N_r}$; $\boldsymbol{x}(t)$ represents the t-th **time domain** sample of the MIMO-OFDM symbol; N_r is the number of receive antennas; $\boldsymbol{N}$ is the additive noise; $h(\cdot)$ represents the multi-path channel, such as the 3GPP spatial channel model (SCM) [26] and $q(\cdot)$ represents the non-linearity at the receiver.

For RC-based MIMO-OFDM symbol detection, the objective is to recover the **frequency domain** modulation symbols $\{\boldsymbol{z}(n)\}_{n=0}^{N_{sc}-1}$. The inputs to the RC are the **time domain** samples $\{\boldsymbol{x}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$. This time-frequency structural information needs to be explored in the design of RCNet to improve its detection performance beyond that of the existing shallow RC-based symbol detection. In the supervised learning

TABLE I
NOTATIONS

Symbols	Definitions
N_r	Number of receiver antennas
N_t	Number of transmitter antennas
N_{sc}	Number of sub-carriers
N_{cp}	Length of Cyclic Prefix (CP)
Q	Number of MIMO-OFDM symbols in the training set (Number of batches in the training set)
N_d	Number of MIMO-OFDM symbols carrying data (testing set)
η	Reference signal (training) overhead
$\boldsymbol{X}$	One MIMO-OFDM symbol at Rx in the time domain (One batch of training input)
$\tilde{\boldsymbol{X}}$	One MIMO-OFDM symbol at Tx in the time domain (One batch of training target in the time domain)
$\boldsymbol{Z}$	One MIMO-OFDM symbol at Tx in the frequency domain (One batch of training target in the frequency domain)
t	Sample index in the time domain
n	Sub-carrier index (modulation symbol index) in the frequency domain
$\boldsymbol{z}(n)$	The nth modulation symbol of one MIMO-OFDM symbol at Tx in the frequency domain
$\tilde{\boldsymbol{x}}(t)$	The tth sample of one MIMO-OFDM symbol at Tx in the time domain
$\boldsymbol{x}(t)$	The tth sample of one MIMO-OFDM symbol at Rx in the time domain
$\boldsymbol{y}(t)$	The tth sample of one MIMO-OFDM symbol at the output of a time domain RC
$\tilde{\boldsymbol{y}}(n)$	The nth modulation symbol of one MIMO-OFDM symbol at the output of a time domain RC in the frequency domain
$\tilde{\boldsymbol{z}}(n)$	The nth modulation symbol of one MIMO-OFDM symbol at the output of a time-frequency domain RC

framework, the training set is defined as $\{d_q\}_{q=0}^{Q-1}$:

$$\begin{aligned} d_q &\triangleq \left(\{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\boldsymbol{z}_q(n)\}_{n=0}^{N_{sc}-1}\right), \\ &\cong \left(\{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\tilde{\boldsymbol{x}}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}\right), \\ &\cong \left(\boldsymbol{X}_q, \boldsymbol{Z}_q\right) \cong \left(\boldsymbol{X}_q, \tilde{\boldsymbol{X}}_q\right). \end{aligned} \tag{3}$$

$\cong$ represents equivalently defined as; The subscript 'q' stands for the qth MIMO-OFDM symbol, i.e, the training set has Q batches in total. The notations used are summarized in Table I.

III. Time-Frequency RC–Incorporating Structural Information

To incorporate the time-frequency structural information inherent in the OFDM signal structure, we introduce in this section the new concept of "time-frequency RC" in addition to the shallow RC. For differentiation, the shallow RC is referred to as the "time domain RC" in this paper. To provide a systematic view of the RC-based design and to better articulate how the structural information is incorporated, we will first briefly discuss the "time domain RC" used in our previous work [6], [24] before describing the "time-frequency RC".

A. Time Domain RC–The Shallow RC

One realization of the time domain RC is illustrated in Fig. 2, namely ESN [27]. In the figure, the collection of neurons is denominated as a *reservoir*. The ESN drives the input signal into a high dimensional dynamic response through a **fixed** random projection, where the response signal is represented by the trajectory of hidden neuron states. Meanwhile, non-linear activation functions are applied to the neuron states transition. The neurons in the reservoir are sparsely connected with fixed weights to satisfy certain distributions under which the response signals are asymptotically uncorrelated to the initial neuron states [27]. One justification for using fixed hidden states transition is from an experimental fact that the dominant changes of an RNN's weights during training happen at the output layer [28]. Finally, the desired outputs are obtained by learning a combination of the non-linear response signals.

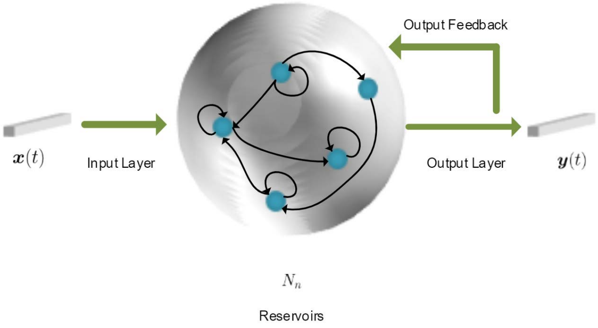


Fig. 2. RCNet: Time Domain RC-ESN Realization.

Given the training set $\{d_q\}_{q=0}^{Q-1}$, the states of the reservoir generated by the qth batch are

$$\begin{aligned} &\boldsymbol{s}_q^T(t+1) \\ &\quad = f\left(\boldsymbol{s}_q^T(t)\boldsymbol{W}_s + \boldsymbol{x}_q^T(t)\boldsymbol{W}_{in} + \tilde{\boldsymbol{x}}_q{}^T(t)\boldsymbol{W}_{fb} + \boldsymbol{n}^T(t)\right) \end{aligned} \tag{4}$$

where $\boldsymbol{s}_q(t) \in \mathbb{C}^{N_n \times 1}$ represents the neurons state vector, with N_n denoting the number of neurons in the reservoir; f is the activation function; $\boldsymbol{W}_s \in \mathbb{C}^{N_n \times N_n}$ denotes the state transition weights matrix; $\boldsymbol{W}_{in} \in \mathbb{C}^{N_r \times N_n}$ denotes the weights of the input layer; $\boldsymbol{n}(t) \in \mathbb{C}^{N_n \times 1}$ is an optional noise regularization term; $\boldsymbol{W}_{fb} \in \mathbb{C}^{N_r \times N_n}$ represents the weights on the feedback path which can be removed when teacher forcing is not required [27]. Correspondingly, the output signal of the time domain RC can be written as

$$\boldsymbol{y}_q^T(t) = \boldsymbol{s}_q^T(t)\boldsymbol{W}_{tout} \tag{5}$$

where $\boldsymbol{W}_{tout} \in \mathbb{C}^{N_n \times N_r}$ represents the readout layer. Note that in the training stage, if teacher forcing is enabled,

Algorithm 1 Time Domain RC-Based MIMO-OFDM Symbol Detection

```
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\boldsymbol{W}_{tout}$, $p^\star$
  for $p \in [0, N_{cp}]$ with step size $P$ do
    Generate $p$-delayed training set $d^{(p)}$
    Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the dynamics equation (4)
    Calculate the output weights $\boldsymbol{W}_{tout}^{(p)}$ using (7) and the objective value $Obj_p$ using (6)
  end for
  Find the optimal $p^\star = \arg\min_p Obj_p$ and $\boldsymbol{W}_{tout} = \boldsymbol{W}_{tout}^{p^\star}$
```

the feedback signal associated with $\boldsymbol{W}_{fb}$ is $\tilde{\boldsymbol{x}}_q$. On the other hand, the feedback signal $\tilde{\boldsymbol{x}}_q$ is replaced by $\boldsymbol{y}_q$ after training.

To drive $\boldsymbol{y}_q(t)$ to the desired time domain MIMO-OFDM symbol, we can minimize the l_2-norm distance between $\{\boldsymbol{y}_q(t)\}_{q=0}^{Q-1}$ and $\{\tilde{\boldsymbol{x}}_q(t)\}_{q=0}^{Q-1}$ through

$$\min_{\boldsymbol{W}_{tout}} \sum_{q=0}^{Q-1} \sum_{t=0}^{N_{cp}+N_{sc}-1} \left\| \boldsymbol{y}_q(t) - \tilde{\boldsymbol{x}}_q(t) \right\|_2^2 . \tag{6}$$

Therefore, the readout weights are updated by the following closed-form expression,

$$\boldsymbol{W}_{tout} = \left(\left[\boldsymbol{S}_0^T, \cdots, \boldsymbol{S}_{Q-1}^T \right]^T \right)^+ \left[\tilde{\boldsymbol{X}}_0^T, \cdots, \tilde{\boldsymbol{X}}_{Q-1}^T \right]^T \tag{7}$$

where $(\boldsymbol{S})^+$ is the pseudo-inverse of $\boldsymbol{S}$, and $\boldsymbol{S}_q$ is stacked by the trajectory of the states as

$$\boldsymbol{S}_q \triangleq [\boldsymbol{s}_q(0), \boldsymbol{s}_q(1), \cdots, \boldsymbol{s}_q(N_{sc} + N_{cp} - 1)]^T. \tag{8}$$

In addition, due to the feedback nature of RC, there exists a lag-effect on the generated state response [12]. A delay parameter can be introduced in the learning process such that the following slightly revised training set is utilized [29],

$$d^{(p)} \triangleq \Big([\boldsymbol{X}_0, \cdots, \boldsymbol{X}_{Q-1}, \boldsymbol{O}_{N_r \times p}]^T, \\ [\boldsymbol{O}_{N_r \times p}, \tilde{\boldsymbol{X}}_0, \cdots, \tilde{\boldsymbol{X}}_{Q-1},]^T \Big)$$

where p is the aforementioned delay parameter, and $\boldsymbol{O}$ represents the zero matrix. The time domain RC is trained with different values of p with $p^\star$ being the value of p generating the minimal objective value defined in (6). This input-output delay offset $p^\star$ will be used to configure the RC for testing. Overall, the training procedure of the time domain RC-based MIMO-OFDM symbol detection is summarized in Algorithm 1, where p is uniformly sampled from 0 to N_{cp} in steps of P. Symbol detection is conducted by feeding the received signal to the learned RC.

B. Time-Frequency RC–RC With Structural Information

Since the time domain RC focuses only on the received time domain signal without processing in the frequency domain, it is clear that it does not take advantage of the structural information of the underlying OFDM signalization. In OFDM systems, the training signal is sent over the frequency domain and the frequency domain provides a much cleaner view of the transmitted signal. Therefore, it is desirable for the NN-based symbol detector to conduct detection tasks in the frequency domain. Meanwhile, the underlying NN-based symbol detector should also take advantage of the time domain correlation.

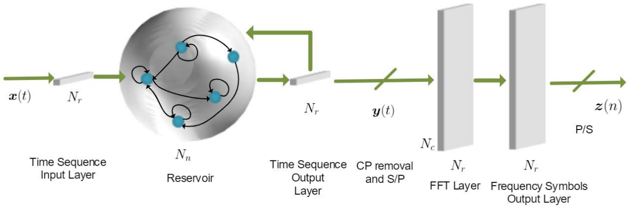


Fig. 3. RCNet: Time-Frequency RC.

To achieve these goals, we introduce the time-frequency RC whose structure is shown in Fig. 3. Following the same line of reasoning as the time domain RC, the reservoir first generates high dimensional response signals according to the input. The response signals are mapped to the desired signals through a time domain layer, a fixed FFT layer, and a frequency domain layer. Accordingly, the time domain layer output is the same as that in (5). After removing the CP and conducting an FFT on $\{\boldsymbol{y}_q(t)\}_{t=N_{cp}}^{N_{sc}+N_{cp}-1}$, the obtained frequency domain symbols are denoted as $\{\tilde{\boldsymbol{y}}_q(n)\}_{n=0}^{N_{sc}-1}$. Subsequently, the frequency domain layer output is defined as

$$\tilde{\boldsymbol{z}}_q^T(n) = \tilde{\boldsymbol{y}}_q^T(n) \text{diag}\left(\boldsymbol{w}_{fout}(n)\right) \tag{9}$$

where $\boldsymbol{w}_{fout}(n) \in \mathbb{C}^{N_r \times 1}$ is the weight specified at the nth sub-carrier; diag$(\boldsymbol{w})$ represents a diagonal matrix which has $\boldsymbol{w}$ as the main diagonal elements. Furthermore, we set the magnitude of each entry of $\boldsymbol{w}_{fout}(n)$ as 1. This allows the introduced frequency domain layer to compensate for the residual phase error after the time domain processing. It is important to note that the frequency domain processing can effectively tune the delay parameter discussed in Section III-A. This is because the FFT layer converts a shift in the time domain to a phase variation in the frequency domain. Accordingly, the added output layers in the time-frequency RC essentially leverage the structural information of the MIMO-OFDM signal.

The learning objective of the time-frequency output layer is

$$\min_{\substack{\boldsymbol{W}_{tout} \\ \{\boldsymbol{w}_{fout}(n)\}_{n=0}^{N_{sc}-1}}} \sum_{q=0}^{Q-1} \sum_{n=0}^{N_{sc}-1} \left\| \boldsymbol{z}_q(n) - \tilde{\boldsymbol{z}}_q(n) \right\|_2^2 \\ s.t. \ \text{diag}\left(|\boldsymbol{w}_{fout}(n)|\right) = \boldsymbol{I}, \quad \forall n = 0, \cdots, N_{sc} - 1. \tag{10}$$

which is equivalent to the time domain objective function defined in (6). In order to seek a proper solution to the above problem, we resort to alternative least squares (ALS) as the solver to obtain closed-form update rules for $\boldsymbol{W}_{tout}$ and $\boldsymbol{w}_{fout}(n)$. The detailed derivation can be found in the Appendix with their closed-form update rules outlined in (17) and (14) respectively. The learning algorithm of the time-frequency RC is summarized in Algorithm 2.

Note that there are certain implementation-related issues that need to be clarified when fully connected layers

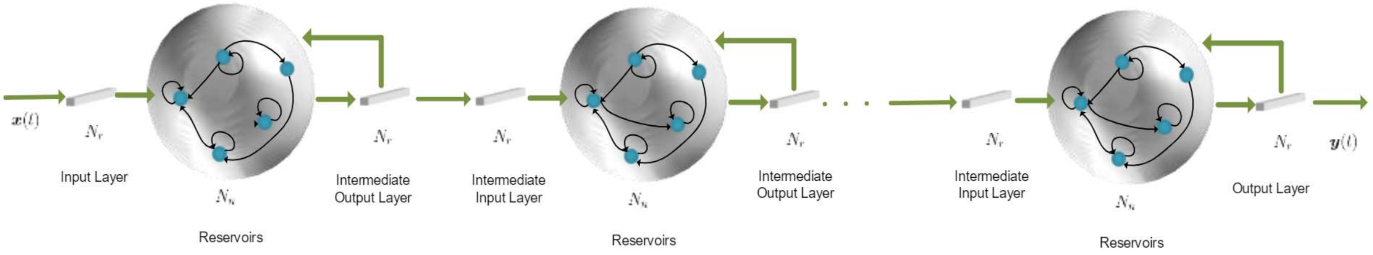


Fig. 4. RCNet: Deep Time RC.

```
Algorithm 2 Time-Frequency RC Based MIMO-OFDM Symbol Detection
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\boldsymbol{W}_{tout}$, $\{\boldsymbol{w}_{fout}(n)\}_{n=0}^{N_{sc}-1}$
  Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the dynamics equation (4)
  Initialize $\boldsymbol{w}_{fout}(n) = \mathbf{1}$, $\forall n = 1, \cdots, N_{sc}$
  while (10) does not converge do
    Update $\boldsymbol{W}_{tout}$ using (17)
    Update $\boldsymbol{w}_{fout}(n)$ using (14)
  end while
```

are employed in the frequency domain, i.e. replacing diag $(\boldsymbol{w}_{fout}(n))$ with $\boldsymbol{W}_{fout}(n)$ in (9). In this case, the learning rule of the output layers becomes

$$\min_{\substack{\boldsymbol{W}_{tout}\\ \{\boldsymbol{W}_{fout}(n)\}_{n=0}^{N_{sc}-1}}} \sum_{n=0}^{N_{sc}-1} \left\| \tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout}\boldsymbol{W}_{fout}(n) - \boldsymbol{Z}(n) \right\|_F^2$$

where $\tilde{\boldsymbol{F}}(n) \triangleq \boldsymbol{I} \otimes \boldsymbol{f}(n)$, $\boldsymbol{f}(n) \in \mathbb{C}^{N_{sc}\times 1}$ is the nth row of a Fourier matrix, $\otimes$ denotes the Kronecker product operation and

$$\boldsymbol{S} \triangleq \big[\boldsymbol{S}_0(N_{cp} : N_{sc} + N_{cp} - 1, :)^T, \\ \boldsymbol{S}_{Q-1}(N_{cp} : N_{sc} + N_{cp} - 1, :)^T\big]^T$$

Based on ALS, the update rule of $\boldsymbol{W}_{fout}(n)$ is given by

$$\boldsymbol{W}_{fout}(n) = (\tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout})^{+}\boldsymbol{Z}(n). \tag{11}$$

However, given $\boldsymbol{W}_{fout}(n)$, $\boldsymbol{W}_{tout}$ is updated by solving a Sylvester's equation which can introduce a heavy computational load. Meanwhile, this brings more parameters to learn which can lead to overfitting since the training set size is usually limited in practical systems.

IV. RCNet–Stacking RCs for a Deep Network

We now introduce RCNet by stacking multiple RCs into a "deep" RC network. Intuitively, this deep structure can be interpreted as decomposing different levels of interference cancellation for the received MIMO-OFDM signal. Based on the discussion in Section III, the basic building block of RCNet can be either the time domain RC or the time-frequency RC.

When the building block is the time domain RC, RCNet can be constructed by stacking them into a deep structure (Deep Time RC) as shown in Fig. 4. Let L denote the total number of building blocks in RCNet. Given the training set $d^{(p)}$, for the l^{th} RC, the state equations are

$$\boldsymbol{s}_q^{(l)T}(t+1) = f\Big(\boldsymbol{s}_q^{(l)T}(t)\boldsymbol{W}_s^{(l)} + \boldsymbol{y}_q^{(l-1)T}(t)\boldsymbol{W}_{in}^{(l)} \\ + \tilde{\boldsymbol{x}}_q^{T}(t)\boldsymbol{W}_{fb}^{(l)} + \boldsymbol{n}^T(t)\Big)$$

where the superscript (l) represents the l^{th} RC; $\boldsymbol{y}^{(l)}$ follows the output equation from the previous layer which is defined in (5). The output layer and intermediate output layers of this RCNet are learned sequentially, i.e. the intermediate output layer closest to the input is learned first; the next RC is learned based on the results generated by the previous one. The input of the l^{th} RC is the output of the $(l-1)^{\text{th}}$ RC after training. The teacher forcing for different RCs is the same. The final output of this RCNet generates an estimate of the desired MIMO-OFDM symbol, i.e. $\boldsymbol{y}_q^{(L)}(t)$. In the state equation of the l^{th} layer, the feedback signal associated with $\boldsymbol{W}_{fb}^{(l)}$ is replaced by $\boldsymbol{y}_q$ after training. The learning algorithm of the deep time RC is summarized in Algorithm 3. This learning method can be interpreted via the boosting framework [30]: by sharing learned features among a set of weak learners, their ensemble can result in a stronger learning ability.

```
Algorithm 3 RCNet: Deep Time RC Based MIMO-OFDM Symbol Detection
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\{\boldsymbol{W}_{tout}^{(l)}\}_{l=0}^{L-1}$, $\{p^{(l)}\}_{l=0}^{L-1}$
  $\{\boldsymbol{x}_q^{(o)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1} = \{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
  for $l$ from 0 to $L-1$ do
    $d_q(l) \triangleq \left(\{\boldsymbol{x}_q^{(l)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\tilde{\boldsymbol{x}}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}\right)$
    for $p \in [0, N_{cp}]$ with step size $P$ do
      Generate $p$-delayed training set $d^{(p)}(l)$
      Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the state equation (4)
      Calculate the output weights $\boldsymbol{W}_{tout}^{(p)}$ using (7) and the objective value $Obj_p$ using (6)
    end for
    Find the optimal $p^{(l)} = \arg\min_p Obj_p$ and $\boldsymbol{W}_{tout}^{(l)} = \boldsymbol{W}_{tout}^{p^{(l)}}$
    Use the learned $\boldsymbol{W}_{tout}^{(l)}$ and $p^{(l)}$ to generate $\{\boldsymbol{x}_q^{(l+1)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
  end for
```

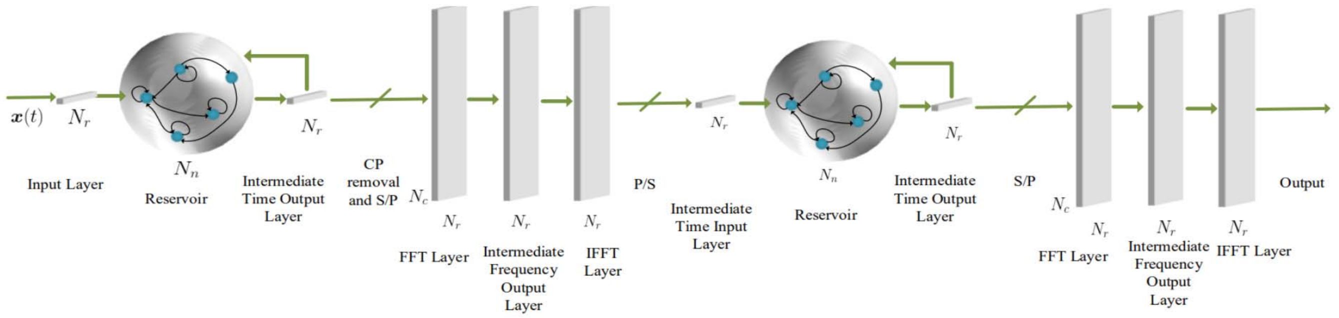


Fig. 5. RCNet: Deep Time-Frequency RC structure.

Similarly, we can change the building block of RCNet from the time domain RC to the newly introduced time-frequency RC. The structure of the corresponding RCNet (Deep Time-Frequency RC) is shown in Fig. 5. The Deep Time-Frequency RC is expected to provide better performance than the Deep Time RC since it takes advantage of the structural information of OFDM signalization. Furthermore, it is important to note that compared to the Deep Time RC version of RCNet, additional IFFT layers need to be added between two adjacent time-frequency RCs to transform the former frequency domain output to a time domain signal for subsequent processing. A similar learning algorithm for this RCNet (Deep Time-Frequency RC) is summarized in Algorithm 4. Note that this learning methodology of RCNet is fundamentally different from conventional gradient-based learning in that the RC structures do not require learning of the internal state transition matrix and the input layer. Using the l_2-norm as the loss function allows us to leverage the least squares framework to arrive at a closed-form solver, which is more efficient than conventional gradient-based objective descent. Training any of the deep RC structures via backpropagation through time (BPTT) is very inefficient since BPTT essentially unfolds the RC structure into a "very deep neural network" in both the time and space dimensions, which can cause severe gradient vanishing issues during training. On the other hand, the sequential learning methodology builds on the boosting mechanism, where results learned from previous layers offer a good starting point for the learning of the deeper layers.

The number of RC components in RCNet is another tunable parameter. A validation set can be utilized for determining the proper value of L. We can simultaneously test the validation error while increasing L: Once the validation error stops decreasing, we can stop increasing L. Furthermore, the number of neurons in each RC can also be configured differently. How to optimize the number of neurons in each RC to achieve the best generalization performance can be treated as future work. The comparison of the complexity of time RC with conventional approaches can be found in Table I in our previous work [24]. Following a similar derivation, the additional number of FLOPS required for the time-frequency RC is on the order of $O(N_r N_{sc}(Q + 1 + Qlog(N_{sc})))$, with training the corresponding deep structures being L times more computationally expensive.

Algorithm 4 RCNet: Deep Time-Frequency RC Based MIMO-OFDM Symbol Detection

```
Input: $\{d_q\}_{q=0}^{Q-1}$
Output: $\{\boldsymbol{W}_{tout}^{(l)}\}_{l=0}^{L-1}$, $\{p^{(l)}\}_{l=0}^{L-1}$
  $\{\boldsymbol{x}_q^{(o)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1} = \{\boldsymbol{x}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
  for $l$ from 0 to $L-1$ do
    $d_q(l) \triangleq \left(\{\boldsymbol{x}_q^{(l)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\tilde{\boldsymbol{x}}_q(t)\}_{t=0}^{N_{cp}+N_{sc}-1}\right) \cong \left(\{\boldsymbol{x}_q^{(l)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}, \{\boldsymbol{z}_q(n)\}_{n=0}^{N_{sc}-1}\right)$
    Generate the state matrix $\{\boldsymbol{S}_q^{(p)}\}_{q=0}^{Q-1}$ according to the state equation (4)
    Initialize $\boldsymbol{w}_{fout}(n) = \mathbf{1}, \forall n = 1, \cdots, N_{sc}$
    while (10) does not converge do
      Update $\boldsymbol{W}_{tout}$ using (17)
      Update $\boldsymbol{w}_{fout}(n)$ using (14)
    end while
    Use the learned $\boldsymbol{W}_{tout}$ and $\boldsymbol{w}_{fout}(n)$ to generate $\{\boldsymbol{x}_q^{(l+1)}(t)\}_{t=0}^{N_{cp}+N_{sc}-1}$
  end for
```

V. Performance Evaluation

In this section, we provide performance evaluations for the introduced RCNet framework under relevant scenarios. The modulation scheme used to generate $z(n)$ is set to be 16-QAM. The simulation parameters in the performance evaluation are configured as the follows: $N_r = 4$, $N_t = 4$, $N_{sc} = 1024$, $N_{cp} = 160$, $Q = 4$, and $N_d = 13$. Note that in this case the training overhead is only 23.5%, which is significantly lower from most other NN-based detection methods that use a prohibitively high training overhead. The channel model adopted in the evaluation is the WINNER II channel model defined in [31], where the transmitter and receiver are configured with uniform linear arrays having half-wavelength antenna spacing. Furthermore, the communication scenario is chosen to be the urban macrocell NLOS outdoor-to-indoor case.

Recall from Fig. 1 that the first Q OFDM symbols constitute the training set and next N_d OFDM symbols are the data symbols that make up the testing set. Therefore, the first Q OFDM symbols will be used to train the RCNet while the bit error rate (BER) will be evaluated using the testing set (data symbols). This training and testing procedure is conducted for 100 consecutive sub-frames. The number of neurons for each layer of the RCNet (the component of the shallow RC) is set as 128. The number of layers L is set as 3 for the deep structures. Furthermore, a time window is added to the input layer of each RC unit as suggested in [24], where the length is set as 128. The state transition matrix $\boldsymbol{W}_s$ is generated randomly to satisfy the echo state property [27], where the spectral radius is chosen to be smaller than 1. The input weights matrix $\boldsymbol{W}_{in}$ is generated randomly from a uniform distribution. In our evaluations, adding teacher forcing does not show an improvement in performance. Therefore, $\boldsymbol{W}_{fb}$ is set to zero.

A. BER Performance Under Tx Non-Linearity

To evaluate the symbol detection performance, we compare RCNet to shallow RC-based strategies as well as conventional methods. To incorporate the Tx non-linearity in our evaluation, the following RAPP model was adopted for the power amplifier (PA),

$$f(x) = \frac{x}{\left[1 + \left(\frac{|x|}{x_{sat}}\right)^{2\rho}\right]^{1/2\rho}} \tag{12}$$

where x represents the PA input signal, ρ is the smoothing parameter, and x_{sat} is the saturation level. When $x \ll x_{sat}$, we have $f(x) \approx x$, implying that the PA output signal has no distortion compared with the input. In our evaluation, we set $\rho = 3$ and $x_{sat} = 1$. As a benchmark, two conventional symbol detection methods, namely linear minimum mean squared error (LMMSE) and sphere decoding (SD) [32], are selected. Since these two methods rely on the knowledge of channel state information (CSI), we utilize LMMSE as the channel estimation strategy based on the Q OFDM symbols of reference signals (training set) assuming perfect knowledge of the noise variance and the linearity of the underlying wireless link.

We first consider the BER performance when the PA input power is in the linear region. In this case, the PA input power is backed off to be far from the PA's saturation region. Accordingly, the input back-off (IBO), which is defined as the ratio between the PA's saturation power to the input power, is chosen to be greater than 8 dB. In Fig. 6 we show the bit error rate (BER) plotted as a function of the received signal to noise ratio (SNR) in dB for various symbol detection schemes, where the SNR is calculated as the average signal to noise ratio across all subcarriers at the receiver side. From the results we can observe that all RC-based methods including the shallow structures and RCNet can perform better than conventional methods when the transmission power is low. This is because the estimated CSI is inaccurate in the low-SNR regime resulting in poor performance of the conventional model-based methods. For example, the performance of LMMSE symbol detection is greatly dependent on the accuracy of the channel state information (CSI), which is estimated using over-the-air pilots in practical systems. Therefore, inherent error in the CSI estimate can deteriorate the symbol detection performance, depending further on the modulation scheme used. On the other hand, RC-based methods are able to learn the underlying features of the channel without explicitly relying on the underlying CSI. Furthermore, it can be seen from the results that the two versions of RCNet provide slight performance improvement over their shallow counterparts demonstrating the benefits of the deep network structure.

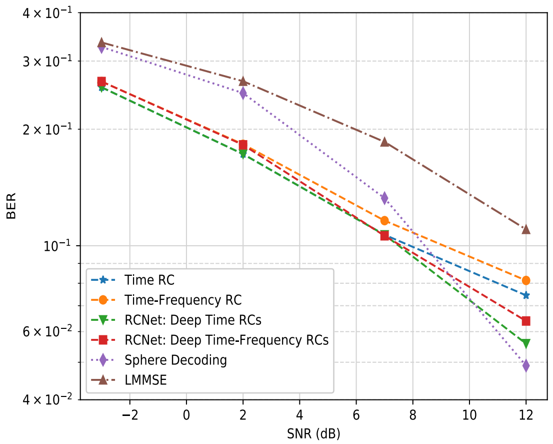


Fig. 6. Average BER for RCNet-based methods and conventional methods in PA linear region.

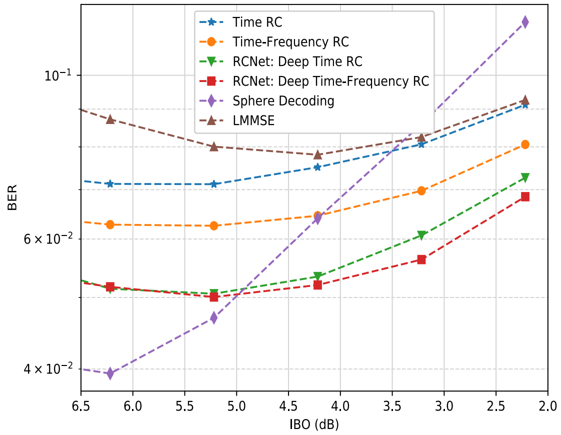


Fig. 7. Average BER for RCNet-based methods and conventional methods by varying IBO to operate in PA's non-linear region, where the corresponding SNR ranges from 13 dB to 17dB.

In Fig. 7, we show the BER performance when the PA input power is close to the saturation region. In this case, the PA's output is distorted due to the compression effect. The distortion occurs when the peak-to-average-power-ratio (PAPR) of the PA input signal is higher than the IBO, where the PAPR of an OFDM signal $x(t)$ is defined as $\|x(t)\|_\infty^2/\|x(t)\|_2^2$. In our evaluation, the signal's PAPR is controlled in the range from 6 dB to 9 dB. Therefore, in order to investigate the BER performance under the PA's compression effect, we choose the IBO to be below 6.5 dB as shown in Fig. 7. The results clearly show that all RC-based methods perform relatively well when the IBO is low, especially when it is lower than 5 dB. Note that the PA efficiency is substantially higher when it is operating at a low IBO. This clearly suggests that RC-based methods can provide an improvement in PA efficiency by compensating for the transmitted waveform distortion at the receiver. Furthermore, the results in Fig. 7 demonstrate the benefits of utilizing the signal's structural

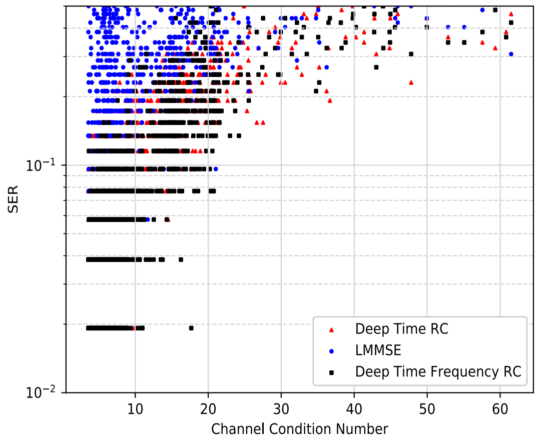


Fig. 8. SER distribution over channel condition numbers.

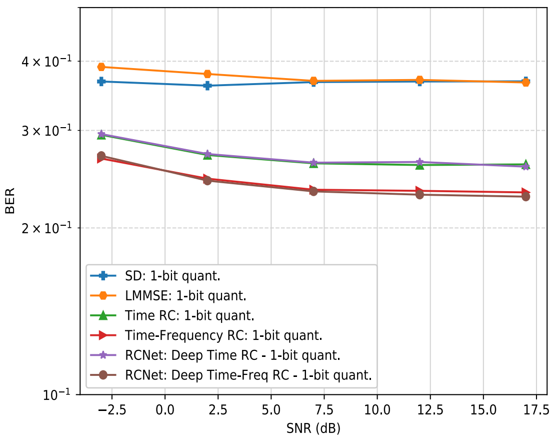


Fig. 9. Average BER curves for RCNet-based methods and conventional methods using 1-bit resolution ADCs.

information in the underlying NN design: the Time-Frequency RC performs significantly better than the Time RC and the Deep Time-Frequency RC performs better than the Deep Time RC. This is because the newly introduced Time-Frequency RC does take advantage of the OFDM signalization in the design of the underlying network structure of RC to address *Challenge 2*. On the other hand, the evaluation results also clearly show the power of the deep nature of the introduced RCNet to address *Challenge 3* without additional training overhead: RCNet performs substantially better than its shallow counterparts with the same training overhead. Note that in Fig. 7, as the IBO reduces, the received SNR will also increase leading to operation in the high SNR region at the receiver. Note that there are existing methods such as Digital Pre-Distortion (DPD) to mitigate the PA's non-linearity by pre-preprocessing the transmit data in the digital domain. However, such methods rely on accurate PA models and parameters, and cannot compensate for the loss when operating in the PA's compression region, but only when operating in its distortion region. Additionally, such mitigation methods rely on corrective processing at the transmitter, whereas RCNet can compensate for this distortion completely at the receiver, therefore, allowing PA operation at the transmitter in the non-linear regime for enhanced efficiency.

Furthermore, in Fig. 8, we investigate the distribution of symbol error rate (SER) over all subcarriers as a function of the channel correlation: The x-axis is the condition number of the spatial channel from all subcarriers. The SER is evaluated in the PA's nonlinear region with an IBO of 6 dB. Fig. 8 shows that the performance of all the methods is affected by the channel condition number. However, compared with the LMMSE approach, both the Time RC and the Time-Frequency RC can achieve significantly better SER performance when the condition number is not too large. Furthermore, Fig. 8 clearly suggests that both the Time RC and the Time Frequency RC achieve a lower average SER than the LMMSE approach under any given channel condition number.

B. BER Performance Under Rx Non-Linearity

The non-linearity in the receiver stems primarily from the quantization of the received signal due to the finite resolution analog-to-digital conversion. For a MIMO-OFDM signal, the in-phase and quadrature components are quantized by a pair of analog-to-digital converters (ADCs), where the input-output relation of the ADCs can be defined as

$$q(x) = \begin{cases} \Delta\lceil x/\Delta\rceil - \Delta/2, & \text{if } |x| < A_{max} \\ A_{max}\cdot \mathrm{sign}(x), & \text{otherwise} \end{cases} \tag{13}$$

in which x is the ADC's input, $\lceil\cdot\rceil$ is the ceiling function, $\Delta > 0$ represents the quantization interval, and A_{max} is the maximal input amplitude of ADC such that $A_{max} = (2^n - 1)\Delta/2$, where n is the number of quantization bits. To be specific, when $\Delta = 2\ A_{max}$, $q(x)$ represents a one-bit ADC. In the state-of-the-art, low-resolution ADCs are often utilized to digitize a signal with a large bandwidth at high frequencies (e.g., mmWave bands and Terahertz bands) or to reduce the power consumption of the underlying ADCs. Note that almost all existing work in the symbol detection domain with limited resolution ADCs makes system assumptions such as large number of antennas and ideal CSI, which are either not compatible with the setting in RCNet or not feasible in practice. Also, to the best of our knowledge, we did not find existing work that can generate stable channel estimation results as well as conduct symbol detection in a MIMO-OFDM system with quantized CSI.

In Fig. 9 and Fig. 10, we investigate the BER performance of uncoded MIMO-OFDM signals under 1-bit and 2-bit quantization respectively ($n = 1$ and $n = 2$) using QPSK modulation as a function of the received SNR. Since only 1-bit and 2-bit ADCs are used, the resulting quantization errors are usually large. From the figures, we can see that conventional methods are very sensitive to quantization errors, for e.g., the SD strategy completely collapses in both cases. On the other hand, all RC-based methods can handle the large quantization errors very well showing the benefits of adopting this particular learning-based approach. It is important to note that the RC-based strategies outperform the conventional LMMSE and SD in all SNRs of interest. Furthermore, a considerable gain can be achieved from Time-Frequency RC to Time RC showing the importance of incorporating structural information in the underlying NN design for MIMO-OFDM symbol detection. As for the comparison

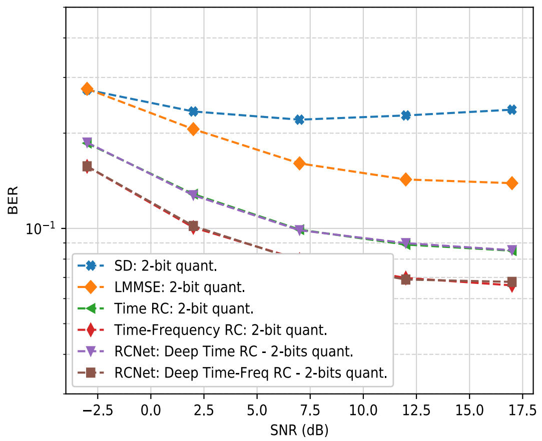


Fig. 10. Average BER curves for RCNet-based methods and conventional methods using 2-bit resolution ADCs.

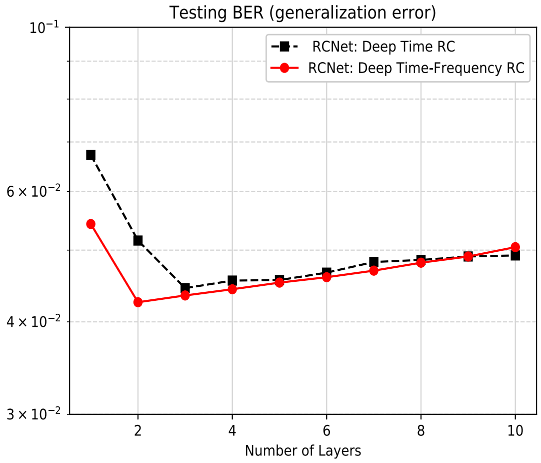


Fig. 11. Testing BER (Generalization Error) for RCNet as a function of L.

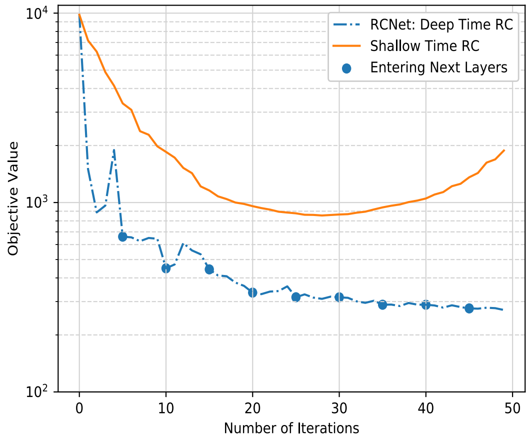


Fig. 12. Learning curves of the Shallow Time RC and RCNet (Deep Time RC).

between the shallow RCs and the RCNet structures, we see very marginal BER performance improvement. This might be due to the fact that the shallow RC already takes the best advantage of the available limited information for symbol detection while additional iteration does not provide performance improvement. Identifying the detailed reasons behind this phenomenon can be treated as a future extension of this work. Note that even though Fig. 6 shows that deep time-frequency RC performs slightly worse than the time RC in the PA's linear region, i.e., in the absence of non-linearities, their performance is typically close to each other in the absence of system impairments. On the other hand, Fig. 7 and Figs. 9, 10 demonstrate that in the PA's non-linear operation region and with extremely low-resolution ADCs in the receiver respectively, both the time-frequency RC and the deep time-frequency RC provide better performance than their time RC counterparts. Finally, even though the uncoded BER range of 0.25 to 0.4 presented in Fig. 9 and Fig. 10 represents an infeasible operational range for a radio receiver, the performance gain of RCNet over model-based methods in this extremely challenging situation of 1-bit or 2-bit quantization and highly distorted CSI demonstrates its applicability and the advantage it offers over conventional methods in such challenging scenarios. By incorporating channel coding, it is possible to improve the RCNet performance further to a feasible operating point for LTE/LTE-Advanced systems.

C. Learning Convergence of RCNet

The evaluation results presented in previous sections show that RC-based methods are effective in the low SNR regime and under the effects of transmitter and receiver non-linearities. These evaluations use $L = 3$ for the RCNet structures. Intuitively, we can increase L to yield better training performance, however, a higher L may cause over-fitting. Fig. 11 shows the testing BER of RCNet as a function of L considering only the transmitter non-linearity: increasing L does ensure a decrease in the generalization error up to $L = 3$ in general. For implementation, a suitable value for L that minimizes the testing BER may be determined via fine-tuning.

We now evaluate the convergence behavior of RCNet under transmit non-linearity. For an SNR of 12 dB, we train different RCs under the same channel realization. The objective function used for tracing the number of iterations is defined in (6) for the Time RC and in (10) for the Time-Frequency RC. Fig. 12 shows the learning curve of the time domain RCs where each iteration corresponds to a fixed delay parameter. For the delayed training of RCNet, we set $P = 5$ and choose the delay parameters uniformly from 0 to N_{cp} in Algorithm 3. Therefore, the training of each RC layer of RCNet requires 5 iterations and a total of $L = 10$ layers are considered in Fig. 12. The legend "entering next layers" represents a single iteration in the for-loop on L in Algorithm 3 and Algorithm 4. For instance, when the iterator l changes from $l = L_1$ to $l = L_1 + 1$, the output signal of layer L_1 "enters" layer $L_1 + 1$. For the shallow time RC, we set $P = 50$ in Algorithm 1, meaning that the resolution used in the search for the optimum delay is finer for the case of shallow RC than that for the case of RCNet. Therefore, there are a total of 50 iterations for the shallow Time RC. From Fig. 12, we clearly observe that a finer delay parameter cannot result in a lower objective value, whereas adding extra layers of RCs on top of the shallow RC can decrease the objective value.

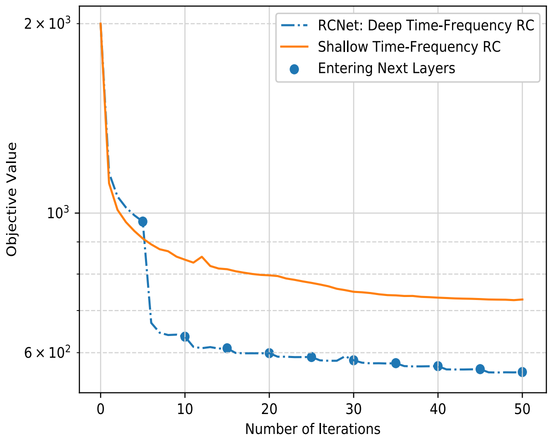


Fig. 13. Learning curves of the Shallow Time-Frequency RC and RCNet (Deep Time-Frequency RC).

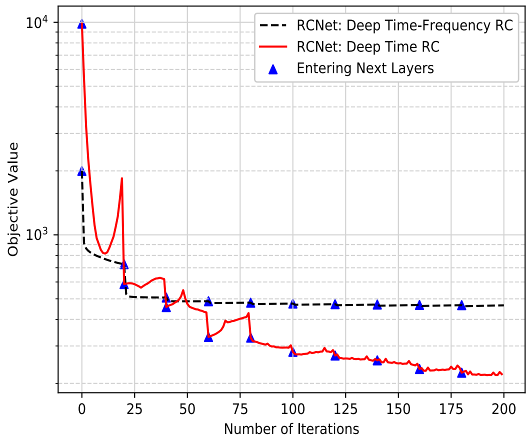


Fig. 14. Learning curve of the Deep Time RC and Deep Time-Frequency RC structures in RCNet.

A similar conclusion can be drawn by observing the learning curve of the Time-Frequency RC and RCNet as depicted in Fig. 13. For RCNet (Deep Time-Frequency RC), we fix the number of iterations of the ALS for solving each RC layer to be 5 in Algorithm 4. Therefore, as in Fig. 12, the training for each RC layer of the RCNet requires 5 iterations and a total of $L = 10$ layers are considered in Fig. 13. As shown in Fig. 13, the objective value decreases significantly by adding one extra RC layer. Compared with the shallow Time-Frequency RC, the fitting error of RCNet (Deep Time-Frequency RC) is significantly smaller. To gain an intuition of the role of the different RC layers in RCNet, the layers can be thought of as learning different levels of features, with each layer producing a different output distance compared to the training target, so that the signals generated by the deeper layers are closer to the target. Therefore, deepening the RCNet structure can lead to a decrease in the training error, as compared to the shallow structure. The comparison between the two RCNet structures (Deep Time RC and Deep Time-Frequency RC) is presented in Fig. 14. In this comparison, we set the number of iterations of each RC layer to be 20 with L being swept from 1 to 10. Fig. 14 shows that the Deep Time-Frequency RC yields a stable objective value where increasing L no longer helps in deceasing the training error. On the other hand, increasing L always helps to reduce the training error for the Deep Time RC. From the generalization error on the testing set shown in Fig. 11, the stability characteristics of the learning curve for the Deep Time-Frequency RC provide an efficient way to determine a suitable value for L that can avoid overfitting issues.

D. Comparison With Other NNs

In Table II, we present the performance comparison between the RC-based symbol detectors (shallow RCs and RCNet) and symbol detectors constructed using alternate NN architectures. Specifically, other popular RNN architectures are considered for performance comparison against RCNet. The variants include long short-term memory (LSTM), bidirectional-LSTM (Bi-LSTM) and gated recurrent unit (GRU) that are more robust against the short-term memory issue in vanilla RNNs. Bi-LSTM is an extension of traditional LSTMs, training on both the original input sequence as well as its reversed copy leading to a doubling of the training time for a given size of the reference signal set. To keep the comparison fair, the number of units in the recurrent hidden layer for the LSTM, Bi-LSTM and GRU structures is set to 128, in line with the 128 neurons used in each layer of RCNet. In this comparative analysis, only the PA non-linearities in the transmitter are exercised by choosing a low input back-off (IBO) of 2.2 dB while ignoring quantization effects of the ADCs in the receiver. It can be seen that in the case of all three RNN-variants, even a prohibitively large training set of 500 reference OFDM symbols, amounting to a reference signal (training) overhead of $\eta = 97.5\%$, is insufficient to achieve an acceptable BER for data transmission. On the other hand, with only 4 reference OFDM symbols, i.e., a reference signal overhead of $\eta = 23.5\%$, all four RCNet detection methods achieve better BER performance than conventional detection methods (SD and LMMSE) listed in Table II. Note that SD performs worse than LMMSE since we are operating in the non-linear region of the PA, as can be confirmed from the results in Fig. 7.

As a benchmark to compare RCNet against, we also evaluate another deep learning-based MIMO symbol detection scheme *DetNet*, presented in [7]. A key requirement of DetNet is the availability of perfect CSI during testing for MIMO symbol detection. However, perfect CSI is either infeasible or extremely costly to obtain in practical wireless systems. In order to provide a fair comparison of DetNet against RCNet which does not rely on CSI for detection, we evaluate DetNet's performance in the 4×4 IID Gaussian channel[2] in the following two cases: 1) Training with perfect CSI, and 2) Training with estimated CSI. In Case 1), we assume perfect CSI is available in the training phase of DetNet. Note that the perfect training CSI assumption is valid for DetNet since the amount of training data is abundant for the receiver to obtain the close-to-perfect CSI. On the other hand, during the testing phase of DetNet, like in the case of LMMSE and SD, we assume the

[2] When training DetNet with the IID Gaussian channel, the online training overhead consists only of the pilot overhead of 23.5%, which is needed for CSI estimation, assuming the network is trained offline with a 60% overhead.

TABLE II
BIT ERROR RATE (BER) COMPARISON WITH ALTERNATE NN-BASED METHODS (PA IBO = 2.2 dB)

Detection Framework	**Training Symbols**	η **(%)**	**Training Epochs**	**Training BER**	**Testing BER**
Time RC [24]	4	23.5	N/A	2×10^{-3}	9×10^{-2}
Time-Frequency RC				8×10^{-3}	8×10^{-2}
RCNet: Deep Time RC				1.4×10^{-3}	7.3×10^{-2}
RCNet: Deep Time-Frequency RC				3.5×10^{-2}	6.9×10^{-2}
LSTM	4	23.5	300	3×10^{-2}	4.7×10^{-1}
Bi-LSTM				4.6×10^{-5}	4.7×10^{-1}
GRU				6×10^{-2}	4.7×10^{-1}
LSTM	10	43.5	300	2.2×10^{-1}	4.7×10^{-1}
Bi-LSTM				5.1×10^{-2}	4.7×10^{-1}
GRU				2.7×10^{-1}	4.8×10^{-1}
LSTM	20	60.6	300	2×10^{-1}	4×10^{-1}
Bi-LSTM				9.9×10^{-2}	4×10^{-1}
GRU				2.5×10^{-1}	4.1×10^{-1}
LSTM	500	97.5	30	4.5×10^{-1}	4.4×10^{-1}
Bi-LSTM				4.4×10^{-1}	4.3×10^{-1}
GRU				4×10^{-1}	4.4×10^{-1}
LSTM	5000	99.7	30	4.4×10^{-1}	4.3×10^{-1}
LMMSE	4	23.5	N/A	N/A	9.2×10^{-2}
Sphere Decoder					1.3×10^{-1}
DetNet [7] (IID Gaussian Channel)	3000	83.5	2×10^5		
Training with Perfect CSI				7×10^{-2}	1.7×10^{-1}
Training with Estimated CSI				1.3×10^{-1}	1.5×10^{-1}
MMNet [23] (3GPP MIMO Channel)					
SNR = $13 - 18$ dB	3096	38.24	1000	2.4×10^{-1}	2.55×10^{-1}

LMMSE channel estimator is adopted to obtain the underlying CSI estimate. Accordingly, the estimated CSI is utilized in the testing phase. In Case 2), we assume the LMMSE estimator is also adopted in the training phase of DetNet to make sure the training and testing environments are the same. It is important to note that in both cases the training/channel estimation overhead is the same: the training overhead in the training phase + CSI estimation overhead in the testing phase. Further realizing that a PA backoff of 2.2 dB maps to a received SNR of 17 dB for RCNet, the same SNR value is used to train and test DetNet with its default configurations of parameters such as number of training iterations, learning rate and batch sizes. Table II clearly suggests that RCNet can outperform DetNet in both cases. To be specific, when training with perfect CSI, DetNet can achieve a testing BER of 0.17 with a training overhead of 83.5% (a training overhead of 60% and a pilot overhead of 23.5% for CSI estimation) On the other hand, the testing BER for DetNet is 0.15 when trained with estimated CSI. Both DetNet results in IID channels are significantly higher than 0.069, the corresponding testing BER of RCNet (Deep Time-Frequency RC). Furthermore, we faced significant difficulty ensuring the stability and convergence of the training process of DetNet with the spatially-correlated 3GPP SCM channel. This demonstrates that DetNet cannot be extended to realistic correlated MIMO channels. Additionally, for DetNet, a fixed training overhead is needed to train the underlying neural network in addition to the pilot overhead for channel estimation. This leads to a much higher training symbol overhead compared to RCNet. Furthermore, once the channel statistics have changed, the underlying neural network needs to be trained again under the new channel statistics.

For an exhaustive comparison with other NN-based detection methods, we also consider an iterative MIMO symbol detection framework such as MMNet [23]. Our evaluation of the MMNet framework utilizes the urban macrocell NLOS outdoor-to-indoor scenario in the Quadriga channel simulator, which is the same scenario used in the WINNER II channel model for evaluating RCNet. From Table II, it can be seen that for a 4×4 MIMO system, even with a training overhead of 38.24%, the BER performance is as high as 0.255 across the SNR range used for 3GPP MIMO channels in [23]. Note that this setting for evaluating MMNet is significantly different from [23], where massive MIMO channels (16×64 or 32×64) are considered. A best-effort hyperparameter tuning (learning rate and number of layers) is also performed in order to achieve the lowest possible SER for the 4×4 system, which is very different from the configurations for which MMNet parameters are originally trained. The poor performance of MMNet in the 4×4 MIMO setting may be understood from the fact that iterative methods such as MMNet rely on the asymptotic property of the massive MIMO channel to achieve optimality [33], which does not hold for small number of transmit and receive antennas.

Overall, from the comparison with other RNN-based symbol detection strategies we can clearly see that RCNet offers advantages of requiring very limited training overhead to provide good performance of MIMO-OFDM symbol detection tasks. Compared with conventional model-based symbol detection strategies, we can see that RCNet provides advantages such as being robust to model mismatch and RF non-linearities (transmitter and/or receiver). On the other hand, the state-of-art NN-based symbol detection strategies, such as *DetNet* and *MMNet* either require extensive amount of training data, time, and computational resources compared to RCNet or are not well-suited to a MIMO configuration with limited antennas.

VI. Conclusion and Future Work

RC-based symbol detectors have been introduced for MIMO-OFDM systems to work under very limited training sets. This paper introduced a deep RNN-based network called *RCNet* to 1) incorporate structural information of the OFDM signalization, and 2) deepen the original shallow RC-based symbol detection strategies to further improve the detection performance. Incorporating structural information has been achieved through the invention of Time-Frequency RC where the learning is done both in the time and frequency domains to take advantage of the time-frequency structure of the underlying OFDM signal. Meanwhile, the deep nature of RCNet has been achieved by extending a shallow RC structure to a deep RC structure in the following two ways: cascading time domain RCs and cascading Time-Frequency RCs. The associated learning algorithms have been developed for each of these extensions and extensive evaluation has been conducted. Experimental results showed that RCNet can outperform conventional methods using a limited training set under non-linear RF effects of the wireless link demonstrating the effectiveness of incorporating structural information and deepening RCs for symbol detection.

An important area of exploration for future research is how to determine the optimal L, i.e., the number of RCs in RCNet especially under receiver non-linearity. Connections to the boosting method may provide insights on designing the number of neurons in each layer. Another interesting direction for future work is how such RC-based detection methods can be combined with transmit-side precoding to jointly optimize the link performance with limited CSI feedback at the transmitter. The full potential of the RCNet symbol detection method is yet to be explored. From a theoretical standpoint, it would also be meaningful to analyze the functionality of each layer in the interference cancellation for a multi-user MIMO network.

Appendix

We now consider using alternative least squares to solve the problem (10). When $\boldsymbol{W}_{tout}$ is given, (10) can be rewritten as

$$\begin{aligned}\min_{\boldsymbol{w}_{fout}(n)} \; & \|\tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout}\mathrm{diag}(\boldsymbol{w}_{fout}(n)) - \boldsymbol{Z}(n)\|_F^2 \\ s.t. \; & \mathrm{diag}(|\boldsymbol{w}_{fout}(n)|) = \boldsymbol{I}, \quad \forall n = 0, \cdots, (N_{sc}-1),\end{aligned}$$

where the definition of $\tilde{\boldsymbol{F}}(n)$, $\boldsymbol{Z}(n)$ and $\boldsymbol{S}$ are given in Sec. III-B. By spelling out the objective, the optimization problem is equivalent to

$$\begin{aligned}\min_{\boldsymbol{w}_{fout}(n)} \; & -\mathrm{Real}\big(\mathrm{Tr}(\bar{\boldsymbol{Z}}(n)\mathrm{diag}(\boldsymbol{w}_{fout}(n))\boldsymbol{Z}(n)^H)\big) \\ s.t. \; & \mathrm{diag}(|\boldsymbol{w}_{fout}(n)|) = \boldsymbol{I}, \quad \forall n = 0, \cdots, (N_{sc}-1),\end{aligned}$$

where $\bar{\boldsymbol{Z}}(n) \triangleq \tilde{\boldsymbol{F}}(n)\boldsymbol{S}\boldsymbol{W}_{tout}$. In addition,

$$\begin{aligned}&\mathrm{Tr}\big(\bar{\boldsymbol{Z}}(n)\mathrm{diag}(\boldsymbol{w}_{fout}(n))\boldsymbol{Z}(n)^H\big) \\ &\qquad = \sum_j w_{fout,j}(n)\mathrm{Tr}\big(\bar{\boldsymbol{z}}^j(n)\boldsymbol{z}^j(n)^H\big)\end{aligned}$$

where $\bar{\boldsymbol{z}}^j(n)$ is the jth column of $\bar{\boldsymbol{Z}}(n)$, $\boldsymbol{z}^j(n)$ is the jth column of $\boldsymbol{Z}(n)$ and $w_{fout,j}(n)$ is the j-th entry of $\boldsymbol{w}_{fout}(n)$. In order to minimize the objective function value, we can select

$$\angle w_{fout,j}(n) = -\angle(\boldsymbol{z}^j(n)^H\bar{\boldsymbol{z}}^j(n)) \tag{14}$$

where $\angle$ represents the angle of a complex number. When $\{\boldsymbol{w}_{fout}(n)\}_{n=0}^{N_{sc}-1}$ is fixed in (10), $\boldsymbol{W}_{tout}$ is learned by

$$\min_{\boldsymbol{W}_{tout}} \|\tilde{\boldsymbol{F}}\boldsymbol{S}\boldsymbol{W}_{tout} - \hat{\boldsymbol{Z}}\|_F^2 \tag{15}$$

where $\tilde{\boldsymbol{F}} \triangleq \mathrm{diag}(\boldsymbol{F}, \cdots, \boldsymbol{F}) \in \mathbb{C}^{QN_{sc}\times QN_{sc}}$ in which $\boldsymbol{F}$ is the Fourier matrix.

$$\hat{\boldsymbol{Z}} \triangleq [\hat{\boldsymbol{Z}}_0^T, \cdots, \hat{\boldsymbol{Z}}_{Q-1}^T]^T,$$

and

$$\hat{\boldsymbol{Z}}_q \triangleq [\hat{\boldsymbol{z}}_q(0), \hat{\boldsymbol{z}}_q(1), \cdots, \hat{\boldsymbol{z}}_q(N_{sc}-1)]^T, \tag{16}$$

in which $\hat{\boldsymbol{z}}_q(n) \triangleq \mathrm{diag}(\boldsymbol{w}^*_{fout}(n))\boldsymbol{z}_q(n)$. Therefore, the learning rule of $\boldsymbol{W}_{tout}$ is

$$\boldsymbol{W}_{tout} = \boldsymbol{S}^+\tilde{\boldsymbol{F}}^H\hat{\boldsymbol{Z}}. \tag{17}$$

References

[1] Z. Zhou, L. Liu, V. Chandrasekhar, J. Zhang, and Y. Yi, "Deep reservoir computing meets 5G MIMO-OFDM systems in symbol detection," in *Proc. 34th AAAI Conf. Artif. Intell*, 2020, pp. 1266–1273.

[2] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. C. 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. Song, J. Bai, Y. Yi, J. Wu, and L. Liu, "Artificial intelligence enabled Internet of Things: Network architecture and spectrum access," *IEEE Comput. Intell. Mag.*, vol. 15, no. 1, pp. 44–51, Feb. 2020.

[4] *Multiplexing and Channel Coding*, Standard 3GPP TS TS 36.212, Rev. 16.0.0, 2020.

[5] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.

[6] 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.

[7] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[8] K.-I. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," *Neural Netw.*, vol. 6, no. 6, pp. 801–806, Jan. 1993.

[9] *Physical Channels and Modulation*, Standard 3GPP Std. TS 36.211, Rev. 16.0.0, 2020.

[10] *Physical Channels and Modulation in NR*, Standard 3GPP TS TS 38.211, Rev.16.0.0, 2020.

[11] 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.

[12] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, Aug. 2009.

[13] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, 2013, pp. 1310–1318.

[14] R. Shafin *et al.*, "Realizing green symbol detection via reservoir computing: An energy-efficiency perspective," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[15] G. E. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Oct. 2012.

[16] S. El Hihi and Y. Bengio, "Hierarchical recurrent neural networks for long-term dependencies," in *Proc. NIPS*, 1996, pp. 493–499.

[17] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[18] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “English (US)how to construct deep recurrent neural networks,” in *Proc. ICLR*, 2014, pp. 1–13.

[19] H. Jaeger, “Discovering multiscale dynamical features with hierarchical echo state networks,” Jacobs Univ. Bremen, Bremen, Germany, Tech. Rep. 10, 2007.

[20] C. Gallicchio and A. Micheli, “Echo state property of deep reservoir computing networks,” *Cognit. Comput.*, vol. 9, no. 3, pp. 337–350, Jun. 2017.

[21] C. Gallicchio, A. Micheli, and L. Pedrelli, “Deep reservoir computing: A critical experimental analysis,” *Neurocomputing*, vol. 268, pp. 87–99, Dec. 2017.

[22] 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.

[23] 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.

[24] 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.

[25] J. Joung, C. K. Ho, K. Adachi, and S. Sun, “A survey on power-amplifier-centric techniques for Spectrum- and energy-efficient wireless communications,” *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 315–333, 1st Quart., 2015.

[26] *Study on 3D Channel Model for LTE*, Standard 3GPP TR TR 36.873, Rev. 12.7.0, 2018.

[27] H. Jaeger, “The ‘echo state’ approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn, Germany, German Nat. Res. Center Inf. Technol. GMD Tech. Rep.*, vol. 148, no. 34, p. 13, 2001.

[28] U. D. Schiller and J. J. Steil, “Analyzing the weight dynamics of recurrent learning algorithms,” *Neurocomputing*, vol. 63, pp. 5–23, Jan. 2005.

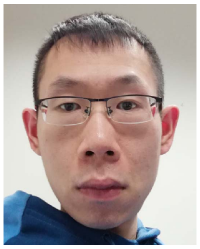
[29] G. Holzmann and H. Hauser, “Echo state networks with filter neurons and a delay&sum readout,” *Neural Netw.*, vol. 23, no. 2, pp. 244–256, Mar. 2010.

[30] Y. Freund, “Boosting a weak learning algorithm by majority,” *Inf. Comput.*, vol. 121, no. 2, pp. 256–285, Sep. 1995.

[31] J. Meinilä, P. Kyösti, T. Jämsä, and L. Hentilä, “Winner II channel models,” in *Radio Technologies and Concepts for IMT-Advanced*. Chichester, U.K.: Wiley, 2009, pp. 39–92.

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

[33] C. Jeon, R. Ghods, A. Maleki, and C. Studer, “Optimality of large MIMO detection via approximate message passing,” in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1227–1231.



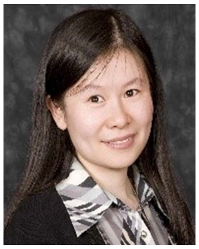
Zhou Zhou received the B.S. degree in communications engineering from the University of Electronic Science and Technology of China (UESTC) in 2011. Since 2018, he has been with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, as a Research Assistant. His current research interests include the broad area of neural networks, machine intelligence, and wireless communications.

Lingjia Liu (Senior Member, IEEE) is currently an Associate Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech. He is also the Associate Director of Wireless@VT. Prior to joining VT, he was an Associate Professor with the EECS Department, University of Kansas (KU). He spent more than four years working with 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 mainly include 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 Scholar 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, and the 2020 WOCC Charles Kao Best Paper Award.

Shashank Jere received the B.S. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2014, and the M.S. degree in electrical engineering from the University of California at Los Angeles in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Virginia Tech. From 2016 to 2019, he worked as a Platform and Product Development Engineer with Qualcomm Technologies Inc., San Diego, CA, USA. His current research interests include the broad area of wireless communications, neural networks, and machine learning.

Jianzhong (Charlie) Zhang (Fellow, IEEE) received the Ph.D. degree from the University of Wisconsin-Madison, Madison, WI, USA. He is a Senior VP and the Head of the Standards and Mobility Innovation Laboratory, Samsung Research America, where he leads research, prototyping, and standards for 5G and future multimedia networks. From 2009 to 2013, he was the Vice Chairman of the 3GPP RAN1 working group and led development of LTE and LTE-advanced technologies, such as 3-D channel modeling, UL-MIMO, CoMP, and carrier aggregation for TD-LTE.



Yang Yi (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Shanghai Jiao Tong University, and the Ph.D. degree in electrical and computer engineering from Texas A&M University. She is currently an Associate Professor with the Bradley Department of Electrical and Computer Engineering, Virginia Tech. Her research interests include very large scale integrated (VLSI) circuits and systems, neuromorphic architecture for brain-inspired computing systems, and low-power circuits design with advanced nanotechnologies for high speed wireless systems.