This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

Harnessing Tensor Structures – Multi-Mode Reservoir Computing and Its Application in Massive MIMO

Zhou Zhou, Lingjia Liu, and Jiarui Xu

***Abstract*—In this paper, we introduce a new neural network (NN) structure, multi-mode reservoir computing (Multi-Mode RC). It inherits the dynamic mechanism of RC and processes the forward path and loss optimization of the NN using tensor as the underlying data format. Multi-Mode RC exhibits less complexity compared to conventional RC structures (e.g. single-mode RC), and offers comparable generalization performance to its single-mode counterpart. Furthermore, we introduce an alternating least square-based learning algorithm as well as the associated theoretical analysis for Multi-Mode RC. The result can be utilized to guide the configuration of NN parameters to sufficiently circumvent over-fitting issues. As a key application, we consider the symbol detection task in multiple-input-multiple-output (MIMO) orthogonal-frequency-division-multiplexing (OFDM) systems with massive MIMO employed at the base stations (BSs). Thanks to the tensor structure of massive MIMO-OFDM signals, our online learning-based symbol detection method generalizes well in terms of bit error rate even using a limited online training set. Evaluation results suggest that the Multi-Mode RC-based learning framework can efficiently and effectively combat practical constraints of wireless systems (i.e. channel state information (CSI) errors and hardware non-linearity) to enable robust and adaptive communications over the air.**

***Index Terms*—Reservoir computing, neural networks, online training, massive MIMO, 5G, imperfect CSI, and non-linearity**

I. Introduction

MASSIVE multiple-input multiple-output (MIMO) is an essential physical layer technique for the 5th generation cellular networks (5G) [1]. By employing a large number of antennas at base stations (BSs), a "favorable propagation" channel condition can be achieved. It allows inter-user interference being effectively eliminated via fairly simple linear precoding or receiving methods, e.g., conjugate beamforming for downlink, or matched filtering for uplink [2].

However, the deployment of massive MIMO in practical systems encounters several implementation constraints. Primarily, for the sake of achieving the promised benefits by massive MIMO, highly accurate channel state information (CSI) is needed [3]. On the other hand, CSI with high precision is challenging to be obtained due to the low received signal-to-noise (SNR) before beamforming/precoding, as well as the limited pilot symbols defined in modern cellular networks due to control overhead [4]. Furthermore, theoretical analysis of massive MIMO systems usually assume ideal linearity and pleasing noise figures requiring exceptionally high costs on radio frequency (RF) and mixed analog-digital components [5]. This ends up with a compromise on the hardware selection which introduces imperfectness (e.g. dynamic non-linearity) to the transmission link. The resulting non-linearity, in turn, leads to waveform distortion and thereby degrades the transmission reliability, which is challenging to be analytically tackled using model-based approaches.

Symbol detection is a critical stage in wireless communications. It is a process that accomplishes miscellaneous interference cancellation at receivers, such as inter-symbol, inter-stream, and inter-user inference, etc. Due to the potential model mismatch from the non-linearity caused by low-cost hardware devices, standard model-based signal processing approaches are no longer effective. With the advent of deep neural networks, there are growing interests in using neural networks (NNs) to handle the model mismatch [4]. In general, NN-based framework aims to compensate for the model mismatch through the non-linearity of NNs. This recent awareness of bridging learning-based approaches to the symbol detection task in massive MIMO systems has posed the following conceptual discussions in the NN design.

- Curse of Antenna Dimensionality: Since the input, hidden, and output layers of NNs are often configured as the same scale as the antennas to jointly extract and process spatial and time-frequency features, the growth of antenna numbers inevitably lead to the increase of the volume of underlying NN coefficients. As NNs essentially learn the underlying statistics of data, the corresponding increase in the parameter dimensionality often imposes an exponential need on the training data set to offer a reasonable generalization result. However, the availability of the training data for cellular networks (e.g.,4G or 5G) especially the online ones is extremely limited due to the associated control overhead [6]. Furthermore, the computational complexity is also evinced with an exponential relation to the NN scale. Learning neural weights through generic back-propagation can result in large computational complexities leading to severe processing delays which are not desirable especially for delay-sensitive applications.
- Blessings of Antenna Dimensionality: On the other hand, a large number of antennas is able to offer favorable propagation conditions [1]. Properly leveraging the asymptotic orthogonality of the wireless channels can often re-

Z. Zhou, L. Liu, and J. Xu are with Wireless@Virginia Tech, ECE Department at Virginia Tech. The work is supported in part by US National Science Foundation (NSF) under grants CCF-1937487 and CNS-2003059. The corresponding author is L. Liu (ljliu@ieee.org).

sult in surprising outcomes, which conversely transforms the curse of dimensionality into "the blessings of dimensionality" [7]. These findings align with the measurement concentration phenomena which are widely applied to simplifying machine learning frameworks [8]. Therefore, to explore learning-based strategies for massive MIMO, we can incorporate inherent structures from the spatial channel as well as the time-frequency features from the modulation waveform to the design of NNs, which is "blessed" to offer good generalization performance yet under a very limited online training.

A. Related Work

A commonly utilized approach for building learning-based symbol detectors is through unfolding existing optimization-based symbol detection methods to deep NNs, such as DetNet [9] and MMNet [10]. Since this framework is based on using explicit CSI, it usually suffers from performance drop or requires extensive hyper-parameters tuning when CSI is not perfect. Meanwhile, the resulting "very deep neural networks" are extremely demanding in computational resources which hinders their applications in practical scenarios. Alternatively, implicit CSI can be utilized to circumvent the above mentioned training issues. For example, [11] introduced a deep feedforward NN for symbol detection in single-input-single-output (SISO) OFDM systems. Due to its independence from channel models, this approach can equalize the channel under nonlinear distortion (power amplifier (PA)). However, this method uses a less-structured deep NN which is yet too complicated to train in practice, since the guaranteed generalization performance is based on extensive training over large datasets that are extremely challenging to be obtained in over-the-air scenarios. Furthermore, "uncertainty in generalization" [4] will arise if the dataset used for training the underlying NN is not general enough to capture the distribution of data encountered in testing. This is especially true for 5G and Beyond 5G networks that needs to offer reliable service under vastly different scenarios and environments.

In 4G/5G MIMO-OFDM systems, there exists different operation modes with link adaptation, rank adaptation, and scheduling on a subframe basis [12]. Therefore, it is challenging, to adopt a complete offline training-based approach. Rather, it is critical to design an online NN-based approach to conduct symbol detection in each subframe only using the limited training symbols that are present 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. On the other hand, conducting effective and efficient learning only through the limited training symbols within a subframe is extremely challenging. To achieve this goal, more structural knowledge of the wireless channel and modulation waveform need to be incorporated as inductive priors to the NNs to significantly relieve the training overhead in each subframe basis [6], [13]. Reservoir computing (RC) and its deep version have been introduced to the MIMO-OFDM symbol detection task in [14]–[17] to achieve learning on a subframe basis. To be specific, [14] is the first work using a vanilla RC structure to conduct MIMO-OFDM symbol detection, where the input and output are defined in the time domain. It shows this simple approach can achieve good symbol detection performance in short memory channels with limited training. [15]–[17] extended the RC-based symbol detection framework by adding units in width and depth to handle channel with long taps as well as more severe non-linearity. Experiments show that the extended RC framework – RCNet can effectively compensate for the distortion caused by non-linear components in wireless systems as well as mitigate miscellaneous interference completely from receiver side only using the training dataset within each subframe.

By referring to the multi-dimensional feature of massive MIMO signals (e.g. elevation and azimuth directions in the spatial domain, the time and frequency domain), we are motivated to incorporate this tensor structure into our symbol detection NN. Although the concept of tensor-driven NNs has been studied before, such as Tensorized NNs in [18], where NN weights are formulated as a tensor-train decomposition [19], and CANDECOMP/PARAFAC (CP) decomposition characterized convolutional layers for learning-acceleration from [20], our strategy is completely different from these techniques as tensor is utilized as the forward path data structure rather than NN coefficients. In its application to massive MIMO systems, instead of treating the input signal as a vector sequence, we define the received signal as a tensor sequence that is consistent with the intrinsic multiple mode property of the underlying massive MIMO signals. Such a signal processing perspective has been studied in our previous work [21]–[27] to solve conventional massive MIMO channel estimation problems. However, a more accurate explicit CSI does not sufficiently lead to an improvement of the transmission reliability, since symbol detection is conducted on a separate stage without knowledge of the channel estimation errors. Our introduced method is to directly demodulate symbols through neural networks avoiding the intermediate channel estimation stage.

B. Contributions

Uplink transmission is a typical low SNR scenario since mobile terminals often use relatively low transmission powers, and the RC framework has not yet been investigated under the scope of the massive MIMO systems. In addition, being able to conduct receive processing – symbol detection on a subframe basis is extremely important for robust and adaptive communications in the 5G and beyond 5G massive MIMO networks. Therefore, we consider the uplink symbol detection in a massive MIMO system with OFDM waveform using a "subframe by subframe" learning framework, i.e., our method accomplishes the symbol detection by using training dataset only from each subframe. The resulting multi-mode processing framework can in general be extended to process any other tasks with tensor structured sequence, such as video, social networks, and recommendation systems, etc.

We name our introduced RC-based NN structure as "multi-mode reservoir computing" (Multi-mode RC), as it inherits

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

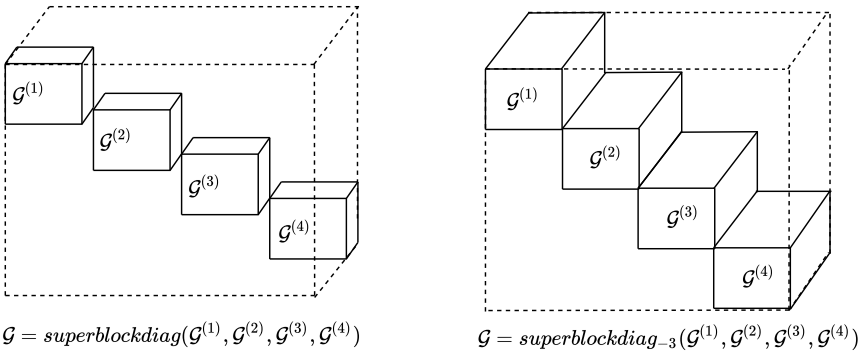


Fig. 1. Illustration of the tensor diagonalization of 3-mode tensors: The left hand-side represents a super diagonal tensor superblockdiag$(\boldsymbol{\mathcal{G}}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \boldsymbol{\mathcal{G}}^{(3)}, \boldsymbol{\mathcal{G}}^{(4)})$, the right hand side represents a super diagonal tensor only on the first two modes, i.e. superblockdiag$_{-3}(\boldsymbol{\mathcal{G}}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \boldsymbol{\mathcal{G}}^{(3)}, \boldsymbol{\mathcal{G}}^{(4)})$, where the diagonal elements are $\boldsymbol{\mathcal{G}}^{(1)}$, $\boldsymbol{\mathcal{G}}^{(2)}$, $\boldsymbol{\mathcal{G}}^{(3)}$ and $\boldsymbol{\mathcal{G}}^{(4)}$.

the dynamic mechanism of RC and processes input-output relation using a tensor format (multi-dimensional array). In our framework, a core-tensor is built as hidden features of the input tensor sequence. Desired output is then obtained through a multi-mode mapping. In terms of tensor algebra, the RC readout is learned through a Tucker decomposition with a deterministic core-tensor thanks to the aforementioned feature extraction. A theoretical analysis is then provided to show the uniqueness of the learned NN coefficients. Our experiments reveal that this uniqueness condition is related to the avoidance of over-fitting issues since it prevents a zero loss value which often results in poor generalization performance on the testing dataset. Compared to single-mode RC, Multi-Mode RC can achieve better symbol detection performance in terms of uncoded bit error rate in the low SNR regime with reduced computational complexity. In addition, the introduced method is shown to be effective to combat extreme waveform distortion, e.g. applying one-bit analog to digital converter (ADC) as the receiving quantization. The remainder of this paper is organized as follows: In Sec. II, we briefly introduce math foundations of tensor and the background of reservoir computing which are utilized to build the concept of Multi-Mode RC in Sec. III. In Sec. IV, the application of Multi-Mode RC to massive MIMO-OFDM symbol detection is discussed. Sec. V evaluates the performance of Multi-Mode RC as opposed to existing symbol detection strategies in massive MIMO-OFDM systems. The conclusion and future research directions are outlined in Sec. VI.

II. Preliminary

This section provides an overview of basic tensor algebra and reservoir computing which will be used in the rest of this paper as the methodology development. In our math notations: scalars, vector, matrix, and tensor are denoted by lowercase letters, boldface lowercase letters, boldface uppercase letters, and boldface Euler script letters respectively, e.g., x, $\boldsymbol{x}$, $\boldsymbol{X}$ and $\boldsymbol{\mathcal{X}}$.

A. Tensor Algebra

Tensor is an algebraic generalization to matrix. A tensor represents a multidimensional array, where the mode of a tensor is the number of dimensions, also known as ways and orders [28]. The $(i_1, i_2, \cdots, i_N)$th element of a N-mode tensor, or namely a Nth order tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$, is denoted as $x_{i_1, i_2, \cdots, i_N}$, where indices range from 1 to their capital versions.

By following matrix conventions, $rank(\boldsymbol{X})$ represents the rank of the matrix $\boldsymbol{X}$. $\boldsymbol{X}^T$, $\boldsymbol{X}^H$ and $\boldsymbol{X}^+$ respectively stands for the transpose, hermitian transpose, and Moore–Penrose pseudoinverse of the matrix $\boldsymbol{X}$. Analogously, the tensor transpose of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\boldsymbol{\mathcal{X}}^{T_\Pi}$ which means the ith mode of $\boldsymbol{\mathcal{X}}^{T_\Pi}$ correspond to the mode numbered as $\Pi(i)$ of $\boldsymbol{\mathcal{X}}$, where Π is a permutation on set $\{1, 2, .., N\}$. Moreover, blockdiag$(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_N)$ represents stacking $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_N$ as a block-diagonal matrix. We denote superblockdiag() as a super-diagonal tensor by stacking its tensor arguments as illustrated in Fig. 1. superblockdiag$_{-n}(\cdot)$ forms a super diagonal tensor except on mode n.

The definition of the mode-n matricization of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\mathbf{X}_{(n)}$, where $(i_1, i_2, \ldots, i_N)$ of $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{I_1 \times I_2 \times \cdots \times I_N}$ maps to the (i_n, j) entry of matrix $\mathbf{X}_{(n)} \in \mathbb{C}^{I_n \times I_{-n}}$, where $I_{-n} := \prod_{k \neq n} I_k$. According to [29],

$$j := 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1)\, J_k \quad \text{with} \quad J_k = \prod_{\substack{m=k+1 \\ m \neq n}}^{N} I_m. \tag{1}$$

The n-mode product of a tensor $\boldsymbol{\mathcal{X}}$ with a matrix $\boldsymbol{U} \in \mathbb{C}^{J \times I_n}$ is defined as,

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{U})_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}.$$

Tucker decomposition is often considered as a higher-order generalization of the matrix singular value decomposition. The Tucker decomposition of a tensor $\boldsymbol{\mathcal{X}}$ is defined as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \tag{2}$$

where $\boldsymbol{A}^{(n)}$ represents the nth factor matrix and $\boldsymbol{\mathcal{G}}$ is named as the core tensor. Accordingly, the mode-n unfolding of the tensor $\boldsymbol{\mathcal{X}}$ is given by

$$\boldsymbol{X}_{(n)} = \boldsymbol{A}^{(n)} \boldsymbol{G}_{(n)} (\boldsymbol{A}^{(1)} \otimes \cdots \boldsymbol{A}^{(n-1)} \otimes \boldsymbol{A}^{(n+1)} \cdots \otimes \boldsymbol{A}^{(N)})^T, \tag{3}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

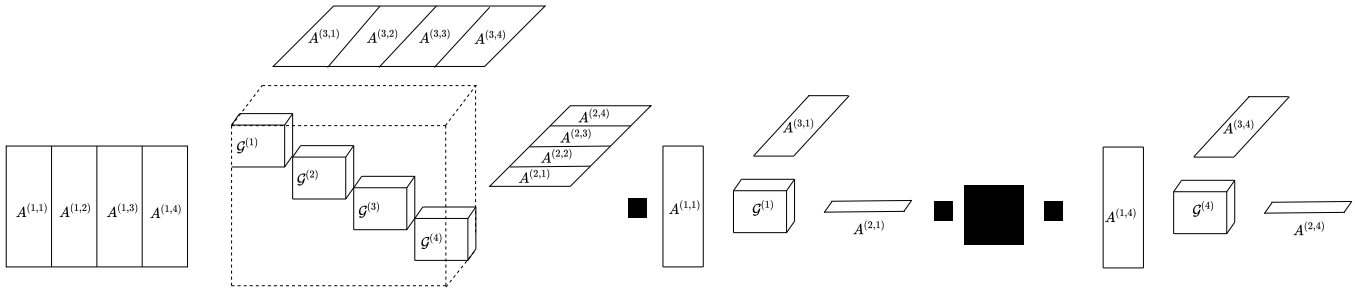


Fig. 2. Illustration for a Tucker decomposition of a three mode tensor: Core tensor and factor matrices are with four partitions.

where $\boldsymbol{G}_{(n)}$ is the mode-n unfolding of $\boldsymbol{\mathcal{G}}$. Note the above unfolding tensor has a reverse order in the Kronecker products of factor matrices which differs to [28]. This is because we alter the way to pile up the indices of unfolding tensors according to (1).

We now consider a super diagonal core tensor $\boldsymbol{\mathcal{G}}$ with K blocks, i.e.,

$$\boldsymbol{\mathcal{G}} = \text{superblockdiag}(\boldsymbol{\mathcal{G}}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \cdots, \boldsymbol{\mathcal{G}}^{(K)})$$

where $\boldsymbol{\mathcal{G}}^{(k)} \in \mathbb{C}^{I_1^{(k)} \times I_2^{(k)} \times \cdots I_N^{(k)}}$ and the matrix $\boldsymbol{A}^{(n)}$ being partitioned as

$$[\boldsymbol{A}^{(n,1)}, \boldsymbol{A}^{(n,2)}, \cdots, \boldsymbol{A}^{(n,K)}].$$

Accordingly, the resulting n-mode product between $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{A}^{(n)}$ can be written in terms of a super diagonal tensor except on the nth mode:

$$\begin{aligned}\boldsymbol{\mathcal{G}} \times_n \boldsymbol{A}^{(n)} = \text{superblockdiag}_{-n}(&\boldsymbol{\mathcal{G}}^{(1)} \times_n \boldsymbol{A}^{(n,1)}, \\ &\boldsymbol{\mathcal{G}}^{(2)} \times_n \boldsymbol{A}^{(n,2)}, \cdots, \boldsymbol{\mathcal{G}}^{(K)} \times_n \boldsymbol{A}^{(n,K)}).\end{aligned}$$

When we assume the core tensor of $\boldsymbol{\mathcal{X}}$ is super-diagonal, the Tucker decomposition defined in (2) can be alternatively expressed in terms of a summation of sub-Tucker decompositions:

$$\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} \boldsymbol{\mathcal{G}}^{(k)} \times_1 \mathbf{A}^{(1,k)} \times_2 \mathbf{A}^{(2,k)} \cdots \times_N \mathbf{A}^{(N,k)}. \quad (4)$$

An illustration for Tucker decomposition of a three mode tensor is depicted in Fig. 2. Particularly, when each diagonal component in $\boldsymbol{\mathcal{G}}$ is a scalar, (4) becomes CANDECOMP/PARAFAC (CP) [30] decomposition simply written as

$$\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} g^{(k)} \mathbf{a}^{(1,k)} \circ \mathbf{a}^{(2,k)} \circ \cdots \circ \mathbf{a}^{(N,k)}, \quad (5)$$

where $\circ$ represents the outer product between two vectors. These tensor decompositions have already been applied to many disciplines, such as neural networks, wireless communications, audio signal processing, and others. More applications of tensor decomposition can be found at [31].

B. Reservoir Computing

Recurrent neural networks (RNNs) are powerful in processing sequential data by its hidden dynamical units. However, its inherent training difficulty limited its application to a wider research area. Reservoir computing (RC) represents a simple RNN framework: The 'reservoir' is composed of nonlinear components and recurrent loops, where the non-linearity allows RC to process complex problems and the recurrent loops enable RC with memory; the 'computing' is achieved by reading out the states in the reservoir through learned NN layers. The training of this framework is conducted only on the readout layers which fundamentally circumvents gradient vanishing/explosion issues in back-propagation through time thanks to the fixed reservoir dynamics. RC today is a prolific research area, yielding compelling performance in many tasks, such as image recognition [32], [33], speech recognition [34], [35], wireless communication [6], [14]–[17], [36], etc.

A simple realization of RC noted as "vanilla RC", is characterized by a state equation and an output equation as shown in Fig. 3. The state equation is formulated with time index t by,

$$\boldsymbol{s}(t+1) = \sigma\left(\boldsymbol{W}_{tran}\begin{bmatrix}\boldsymbol{s}(t)\\ \boldsymbol{y}(t)\end{bmatrix}\right) \quad (6)$$

where σ is a nonlinear function, $\boldsymbol{s}(t)$ is a vector representing the internal reservoir state, $\boldsymbol{y}(t)$ is the input vector, and $\boldsymbol{W}_{tran}$ stands for the reservoir weight matrix which is often chosen with a spectral radius smaller than 1 in order to asymptotically achieve a "similarity" to desired dynamic model [37]. The output equation is simply treated as

$$\boldsymbol{z}(t) = \boldsymbol{W}_{out}\begin{bmatrix}\boldsymbol{s}(t)\\ \boldsymbol{y}(t)\end{bmatrix}, \quad (7)$$

where $\boldsymbol{W}_{out}$ is the output weight matrix and $\boldsymbol{z}(t)$ stands for the output. As we can see, the output is with a skip-connection to the input which assimilates to a residual arrangement [38].

III. Multi-Mode Reservoir Computing

In this section, we introduce the framework of Multi-Mode RC. It processes a sequence-in and sequence-out task, where the time sequences are configured with more than one explicit modes, i.e., the input sequence is formulated as $\boldsymbol{Y}(t)$ or $\boldsymbol{\mathcal{Y}}(t)$ rather than a scalar-wise sequence $y(t)$ or a vector-wise $\boldsymbol{y}(t)$.

A. Two-Mode Reservoir Computing

For ease of discussion, we begin by considering a two-mode RC. The architecture is comprised of a recurrent module, a feature queue, and an output mapping.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

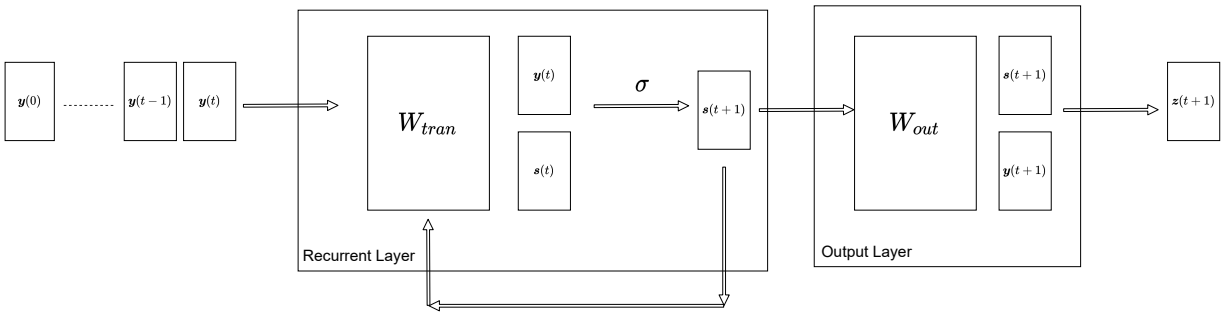


Fig. 3. Illustration for a vanilla RC.

Recurrent Module: A recurrent module maps input sequence $\boldsymbol{Y}(t) \in \mathbb{C}^{N_{in}^{(1)} \times N_{in}^{(2)}}$ to a state sequence $\boldsymbol{S}(t) \in \mathbb{C}^{N_s \times N_s}$, where N_s represents the number of neurons defined on each mode of $\boldsymbol{S}(t)$[1]. Therefore, the total number of neurons is N_s^2. To equally obtain observations from the row-space and column-space of $\boldsymbol{Y}(t)$, we define the recurrent equation as,

$$\boldsymbol{S}(t+1) = \sigma\left(\boldsymbol{W}_{tran}^{(1)} \begin{bmatrix} \boldsymbol{S}(t), & \boldsymbol{O} \\ \boldsymbol{O}, & \tilde{\boldsymbol{Y}}(t) \end{bmatrix} \boldsymbol{W}_{tran}^{(2)\ T}\right) \tag{8}$$

where

$$\tilde{\boldsymbol{Y}}(t) = \text{blockdiag}(\boldsymbol{Y}(t), \boldsymbol{Y}(t-1), \cdots, \boldsymbol{Y}(t-T')),$$

T' is a hyper-parameter representing the length of input window, σ is a non-linear function, $\boldsymbol{W}_{tran}^{(1)} \in \mathbb{C}^{N_s \times (N_s + T' N_{in}^{(1)})}$, $\boldsymbol{W}_{tran}^{(2)} \in \mathbb{C}^{N_s \times (N_s + T' N_{in}^{(2)})}$ are reservoir weight matrices applied on the row and column spaces respectively. Note that (8) also can be written as a sum in the form of (4). Accordingly, the state equation (8) can be regarded as an extension of the standard state equation (6) by incorporating independent mappings into the row and column spaces of state $\boldsymbol{S}(t)$ and input $\boldsymbol{Y}(t)$. It also can be equivalently written via the form of (6) through vectorizing the matrix-wise state and input. Rather than directly applying vectorized state and input to the standard RC, our introduced approach preserves the multi-mode feature of the input signal, where the advantages of using this strategy will be discussed in the analysis and evaluation sections of this paper.

Feature Queue: Our definition of the feature queue $\boldsymbol{G}(t)$ is a queue of sequence, i.e., at a given time t, the sample $\boldsymbol{G}(t)$ is a queue which is stacked up by current state sample $\boldsymbol{S}(t)$ and input sample $\tilde{\boldsymbol{Y}}(t)$. We opt for a simple formulation and use diagonalizing operation to write the queue as follows,

$$\boldsymbol{G}(t) = \text{blockdiag}(\boldsymbol{S}(t), \boldsymbol{S}^T(t), \tilde{\boldsymbol{Y}}(t), \tilde{\boldsymbol{Y}}^T(t)). \tag{9}$$

$\boldsymbol{G}(t)$ is also called extended state sequence as it is obtained with a skip connection to the input. The presence of $\tilde{\boldsymbol{Y}}^T(t)$ and $\boldsymbol{S}^T(t)$ is to create a fair treatment on the row and column space of $\tilde{\boldsymbol{Y}}(t)$ and $\boldsymbol{S}(t)$.

[1]For simplicity, we assume $\boldsymbol{S}(t)$ as a square matrix. In general, it also can be designed as a non-square matrix. Meanwhile, the size of N_s is configured through experiments in order to maintain a balance between overfitting and underfitting according to the datasets. The RC structure based on non-square $\boldsymbol{S}(t)$ is left as our future work.

Output Mapping: An output layer ensures that the feature queue can be identically mapped back to our desired output size. It is defined as:

$$\begin{aligned} \boldsymbol{Z}(t) &= \boldsymbol{W}_{out}^{(1)} \boldsymbol{G}(t) \boldsymbol{W}_{out}^{(2)\ T} \\ &= \boldsymbol{G}(t) \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)} \end{aligned} \tag{10}$$

where $\boldsymbol{W}_{out}^{(1)} \in \mathbb{C}^{N_{out}^{(1)} \times N_f^{(1)}}$ and $\boldsymbol{W}_{out}^{(2)} \in \mathbb{C}^{N_{out}^{(2)} \times N_f^{(2)}}$; $N_f^{(1)} := 2N_s + T'(N_{in}^{(1)} + N_{in}^{(2)})$ and $N_f^{(2)} := 2N_s + T'(N_{in}^{(1)} + N_{in}^{(2)})$ respectively represent the size of row and column of $\boldsymbol{G}(t)$; Meanwhile, $N_{out}^{(1)}$ and $N_{out}^{(2)}$ respectively stand for the size of row and column of $\boldsymbol{Z}(t)$.

Loss Function: In this paper, the loss function is defined to handle sequence-to-sequence tasks. Given a set of $\{\boldsymbol{Y}_q(t), \boldsymbol{Z}_q(t)\}$ as the input-output pairs for training, where q stands for the batch index, our objective aims to generate $\boldsymbol{Z}_q(t)$ by using $\boldsymbol{Y}_q(t)$. Therefore, we use

$$\min_{\boldsymbol{W}_{out}^{(1)}, \boldsymbol{W}_{out}^{(2)}} \sum_{q=1}^{N_K} \sum_{t=1}^{N_T} \| \boldsymbol{Z}_q(t) - \boldsymbol{G}_q(t) \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)} \|_F^2, \tag{11}$$

where $\| \cdot \|_F$ is the Frobenius norm of a matrix. Although the loss function is simply formulated via a least square framework, it offers a connection between the RC readout learning and alternating least squares (ALS) algorithm which has been widely used and can be easily analyzed in the context of tensor decomposition [39]. We can further stack $\boldsymbol{Z}_q(t)$ and $\boldsymbol{G}_q(t)$ to 4-mode tensors along the time axis and the batches to have $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out}^{(1)} \times N_{out}^{(2)} \times N_T \times N_K}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_f^{(1)} \times N_f^{(2)} \times N_T \times N_K}$ respectively. Accordingly, the loss objective (11) can be rewritten in a concise way,

$$\min_{\boldsymbol{W}_{out}^{(1)}, \boldsymbol{W}_{out}^{(2)}} \| \boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)} \|_F^2. \tag{12}$$

In the training stage, we feed a batch of sequences to RC and solve the problem (12) using alternating least squares, where $\boldsymbol{W}_{out}^{(1)}$ and $\boldsymbol{W}_{out}^{(2)}$ are iteratively updated by solving the following matrix-wise least square problems,

$$\boldsymbol{W}_{out}^{(1)} = \arg\min_{\boldsymbol{W}_{out}^{(1)}} \| \boldsymbol{Z}_{(1)} - \boldsymbol{W}_{out}^{(1)} \boldsymbol{G}_{(1)} (\boldsymbol{W}_{out}^{(2)} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K})^T \|_F$$

$$\boldsymbol{W}_{out}^{(2)} = \arg\min_{\boldsymbol{W}_{out2}} \| \boldsymbol{Z}_{(2)} - \boldsymbol{W}_{out}^{(2)} \boldsymbol{G}_{(2)} (\boldsymbol{W}_{out}^{(1)} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K})^T \|_F.$$

The iterative process continues until a certain stopping criterion is reached. In this ALS formulation, $\boldsymbol{G}_{(1)}$ and $\boldsymbol{G}_{(2)}$

TABLE I
NOTATIONS OF MULTI-MODE RC

Notations	Definitions
T'	Input window length
N	Number of signal mode in Multi-Mode RC
$N_{in}^{(n)}$	Number of RC input at mode n
$N_{out}^{(n)}$	Number of RC output at mode n
$N_f^{(n)}$	Feature queue size at mode n
N_T	Input and output sequence length
N_K	Training batch size
N_s	Number of neurons on each mode of a state tensor
τ	Delay configuration in RC state response
$\boldsymbol{\mathcal{Y}}(t) \in \mathbb{C}^{N_{in}^{(1)} \times N_{in}^{(2)} \times \cdots \times N_{in}^{(n)}}$	A tensor sequence as the input of Multi-Mode RC
$\boldsymbol{\mathcal{G}}(t) \in \mathbb{C}^{N_f^{(1)} \times N_f^{(2)} \times \cdots \times N_f^{(N)}}$	A tensor sequence as the internal feature of Multi-Mode RC
$\boldsymbol{\mathcal{Z}}(t) \in \mathbb{C}^{N_{out}^{(1)} \times N_{out}^{(2)} \cdots \times N_{out}^{(N)}}$	A tensor sequence as the output of RC
$\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{N_{in}^{(1)} \times N_{in}^{(2)} \times \cdots \times N_{in}^{(N)} \times N_T \times N_K}$	A higher order tensor by stacking $\boldsymbol{\mathcal{Y}}(t)$ through time samples and batches
$\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_f^{(1)} \times N_f^{(2)} \times \cdots \times N_f^{(N)} \times N_T \times N_K}$	A higher order tensor by stacking $\boldsymbol{\mathcal{G}}(t)$ through time samples and batches
$\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out}^{(1)} \times N_{out}^{(2)} \cdots N_{out}^{(N)} \times N_T \times N_K}$	A higher order tensor by stacking $\boldsymbol{\mathcal{Z}}(t)$ through time samples and batches

represent the mode-1 and mode-2 unfoldings of tensor $\boldsymbol{\mathcal{G}}$. However, directly using this ALS calculation often requires large memory resources due to the Kronecker products. Therefore, we introduce an alternative approach to calculate the ALS which is discussed in Appendix.

Since reaction delays exist in RC systems [16], we often need to add another parameter τ, namely "Delay of RC states" in the loss objective to optimize. Therefore, we have the following augmented loss objective,

$$\min_{\tau<\tau_{max}} \min_{\boldsymbol{W}_{out}^{(1)}, \boldsymbol{W}_{out}^{(2)}} \sum_{q=1}^{N_K} \sum_{t=1}^{N_T} \|\boldsymbol{Z}_q(t) - \boldsymbol{G}_q(t+\tau) \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)}\|_F^2, \tag{13}$$

where τ_{max} represents the upper bound of τ to search. Accordingly, the samples of $\boldsymbol{G}_q(t)$ ranging from time index $t = 1$ to $t = N_T + \tau_{max}$ are obtained by concatenating $\{\boldsymbol{Y}(t)\}_{t=1}^{N_T}$ with a τ_{max}-length zero matrix at the end as the RC input. At testing stage, the learned τ are applied to truncate the RC state sequence such that the output sequence becomes $\{\boldsymbol{G}(t+\tau) \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)}\}_{t=1}^{N_T}$, since the output is anticipated as a N_T-length sequence. In general, the output sequence can be truncated off more or less samples in order to match the desired sequence of the tasks.

B. Multi-Mode Reservoir Computing

The structure of multi-mode (beyond 2-mode) reservoir computing is illustrated in Fig. 4. As a general framework of 2-mode RC, each component is respectively extended as

- **Recurrent Module:**

$$\boldsymbol{\mathcal{S}}(t+1) = \sigma(\text{superblockdiag}(\boldsymbol{\mathcal{S}}(t), \tilde{\boldsymbol{\mathcal{Y}}}(t)) \times_1 \boldsymbol{W}_{tran}^{(n)} \times_2 \cdots \times_N \boldsymbol{W}_{tran}^{(N)}) \tag{14}$$

$$\tilde{\boldsymbol{\mathcal{Y}}}(t) = \text{superblockdiag}(\boldsymbol{\mathcal{Y}}(t), \cdots, \boldsymbol{\mathcal{Y}}(t-T')) \tag{15}$$

- **Feature Queue**:

$$\boldsymbol{\mathcal{G}}(t) = \text{superblockdiag}(\text{comb}(\boldsymbol{\mathcal{S}}(t)), \text{comb}(\tilde{\boldsymbol{\mathcal{Y}}}(t))) \tag{16}$$

where

$$\text{comb}(\boldsymbol{\mathcal{S}}(t)) := \text{superblockdiag}(\boldsymbol{\mathcal{S}}(t), \boldsymbol{\mathcal{S}}^{T_{\Pi_1}}(t), \boldsymbol{\mathcal{S}}^{T_{\Pi_2}}(t) \cdots),$$

and $\Pi_1, \Pi_2 \cdots$ stand for permutation patterns which are up to $N!$ cases.

- **Output Layer**:

$$\boldsymbol{\mathcal{Z}}(t) = \boldsymbol{\mathcal{G}}(t) \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)} \cdots \times_N \boldsymbol{W}_{out}^{(N)} \tag{17}$$

- **Loss Function**:

$$\min_{\tau} \min_{\boldsymbol{W}_{out}^{(1)}, \boldsymbol{W}_{out}^{(2)}, \cdots, \boldsymbol{W}_{out}^{(N)}} \sum_{q=1}^{N_K} \sum_{t=1}^{N_T} \|\boldsymbol{\mathcal{Z}}_q(t) - \boldsymbol{\mathcal{G}}_q(t+\tau) \times_1 \boldsymbol{W}_{out}^{(1)} \cdots \times_N \boldsymbol{W}_{out}^{(N)}\|_F^2 \tag{18}$$

Similarly to the 2-mode case, the output weights are learned through alternating least squares. The optimization (18) can also be formulated as a high order tensor decomposition as defined in (12). Moreover, to further avoid model overfitting, regularization terms can be added in the loss function, such as ridge regression, i.e., $\|\boldsymbol{W}_{out}^{(1)}\|_F^2 + \|\boldsymbol{W}_{out}^{(2)}\|_F^2 + \cdots + \|\boldsymbol{W}_{out}^{(N)}\|_F^2$. In addition, we can observe that the optimization problem (18) is not the canonical Tucker decomposition defined in [40]. This is because the factor matrices are not designed as full column-rank in our framework. On the contrary, we choose the factor matrices with full row-rank to fulfill the mechanism of RC that is "yielding desired output through dimension reduction from internal memory states."

C. Theoretic Analysis

We now study the condition on the uniqueness of solving (18) via alternating least squares. Through our derivation as presented in Appendix, we can arrive at the following theorem.

Theorem 1: Given $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out}^{(1)} \times N_{out}^{(2)} \times \cdots \times N_{out}^{(N)} \times N_T \times N_K}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_f^{(1)} \times N_f^{(2)} \times \cdots \times N_f^{(N)} \times N_T \times N_K}$ with rank-$(N_{out}^{(1)}, N_{out}^{(2)}, \cdots, N_{out}^{(N)}, N_T, N_Q)$ [2] and rank-$(N_f^{(1)}, N_f^{(2)}, \cdots, N_f^{(N)}, N_T, N_K)$ respectively, and $\forall n$,

[2]This stands for the multi-mode rank of a tensor. The definition can be found in Appendix.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

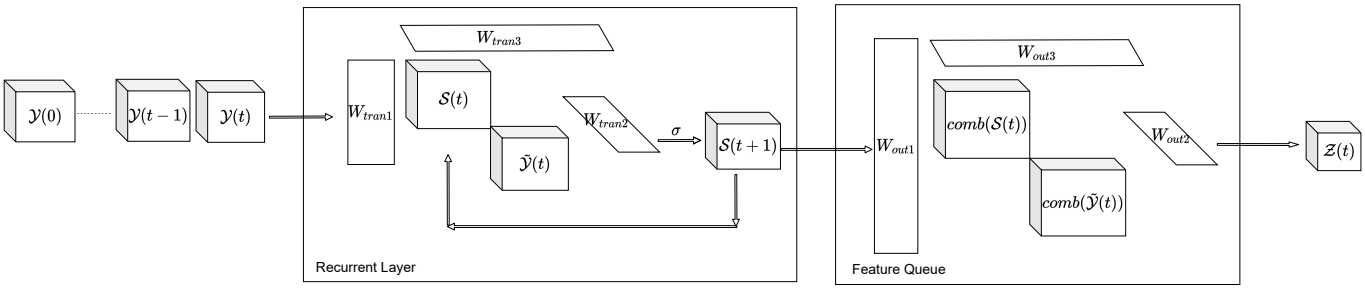


Fig. 4. Illustration of Three-Mode Reservoir Computing Architecture

$N_f^{(n)} \geq N_{out}^{(n)}$, $N \geq 2$, the achieved minimization of (18) is unique by using ALS when the initialization factor matrices are chosen as full rank and

$$\sum_{i \neq n} N_{out}^{(i)} + N_T + N_K \geq N_f^{(n)}, \forall n. \tag{19}$$

The above theorem reveals that the uniqueness condition of the Multi-Mode RC learning is characterized by the shape of the output tensor and feature core-tensor. Alternatively, if we use a single batch for training, i.e., merge the last two modes of the tensors into one, the shape of the output tensor and core-tensor respectively become $\boldsymbol{\mathcal{Z}} \in \mathbb{C}^{N_{out}^{(1)} \times N_{out}^{(2)} \times \cdots \times N_{out}^{(N)} \times (N_T N_K)}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{N_f^{(1)} \times N_f^{(2)} \times \cdots \times N_f^{(N)} \times (N_T N_K)}$. Therefore, the uniqueness condition can be rewritten as

$$\sum_{i \neq n} N_{out}^{(i)} + N_T N_K \geq N_f^{(n)}, \forall n. \tag{20}$$

In our experiments, we observe that when the uniqueness condition holds, the loss-value is often greatly larger than zero. The model thereby does not over-fit to the training dataset which can offer generalization on the unseen testing dataset. More related discussions on this observation are in the evaluation section.

On the other hand, Multi-Mode RC can be analyzed as an advance of conventional RC by imposing particular structures on the output layer, where the conventional RC refers to RC operating on single-mode data structures, i.e., scalars and vectors. To gain this insight, we consider vectorizing the output layer of a 2-mode RC. According to (10), the resulting output equation of the 2-mode RC via a single-mode RC based formulation is given by,

$$Vec(\boldsymbol{Z}(t)) = (\boldsymbol{W}_{out}^{(2)} \otimes \boldsymbol{W}_{out}^{(1)}) Vec(\boldsymbol{G}(t)).$$

The above equation reveals that the output weight of multi-mode RC is forged as a Kronecker product of two sub-matrices to process a "vectorized" $\boldsymbol{G}(t)$. As opposed to single-mode RC using a fully connected layer, the resulting Kronecker output layer is with less freedom on parameters which requires a less amount of data to fit. Meanwhile, the Kronecker structure can further reduce time and space complexity.

We present the complexity analysis results in Table II, where we assume that the conventional RC and Multi-Mode RC are with the same input-output size. In this table, the time complexity of the forward path is calculated by the matrix product between the output layer and RC state at each sample, while the complexity on output learning is from the matrix inverse operations involved in solving the loss objectives of the entire training data set. The memory costs in the forward path are calculated based on the size of internal states and output weights. Meanwhile, the memory spent on learning is on the same scale as the size of the internal state. Moreover, the input buffer length is ignored in this table for simplicity. However, it can be easily calculated by substituting $N_{in}^{(n)}$ as $N_{in_n} T'$ in this table. Note that energy-efficiency is an important topic for the adoption of neural network-based processing. Our previous work [41] shows that RC-based symbol detector is a "green" solution compared to the traditional signal processing-based method with lower energy consumption per information bit. Due to recent active research in the field of neuromorphic computing chips [42]–[44], it is expected that the energy-efficiency of the introduced multi-mode RC can be high. A comprehensive study on this aspect of the multi-mode RC will be treated as a future topic.

IV. Application: Online Symbol Detection for Multi-user Massive MIMO

In this section, we will briefly review the transceiver architecture of multi-user massive MIMO-OFDM systems and elaborate on how to apply Multi-Mode RC to symbol detection of an **uplink** massive MIMO network.

A. Multi-user Massive MIMO-OFDM System

We assume N_u scheduled users are distributed in a cell communicating to a base station (BS) equipped with a massive rectangular array as shown in Fig. 5, where each user is mounted with N_q antennas. Note that N_u is the number of scheduled users in each subframe (not the number of users in the massive MIMO network) and can change from subfame to subframe depending on BS's scheduling strategies. The transmitted signals from all users to BS can be written as $\boldsymbol{X}(t) \in \mathbb{C}^{N_u \times N_q}$. Let $\boldsymbol{x}(t) = \text{vec}(\boldsymbol{X}(t)) \in \mathbb{C}^{N_t \times 1}$, where $N_t = N_u N_q$. Each entry of $\boldsymbol{x}(t)$ is a time sequence which stands for a stream of OFDM signals. For convenience, the OFDM signal $\boldsymbol{x}(t)$ is organized as OFDM resource grids as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

TABLE II
TIME COMPLEXITY AND MEMORY USAGE COMPARISON

NN Operations	Time	Memory
Standard RC forward	$\mathcal{O}(\prod_n N_{out}^{(n)}(\prod_n N_{in}^{(n)} + N_s^N))$	$\mathcal{O}(\prod_n N_{out}^{(n)}(\prod_n N_{in}^{(n)} + N_s^N))$
Multi-Mode RC forward	$\mathcal{O}(\sum_n N_{out}^{(n)}(\prod_n N_{in}^{(n)} N! + N_s^N))$	$\mathcal{O}(\sum_n N_f^{(n)} N_{out}^{(n)} + \prod_n N_{in}^{(n)} N! + N_s^N)$
Standard RC learning	$\mathcal{O}((\prod_n N_{in}^{(n)} + N_s^N)^3 N_K)$	$\mathcal{O}((\prod_n N_{in}^{(n)} + N_s^N) N_K)$
Multi-Mode RC learning	$\mathcal{O}(\sum_n {N_f^{(n)}}^3 \prod_n N_{out}^{(n)} N_K)$	$\mathcal{O}(\prod_n N_{in}^{(n)} N! + N_s^N)$

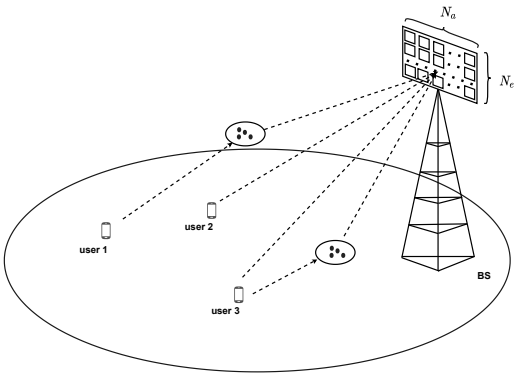


Fig. 5. Uplink transmission in the multi-user massive MIMO system.

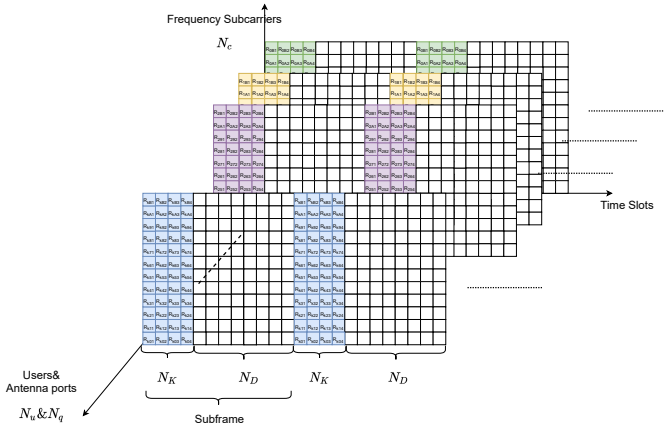


Fig. 6. Massive MIMO-OFDM resource grids (subframe-subcarrier) structure for RC training and symbol detection

illustrated in Fig. 6. In the OFDM resource grids, the horizontal direction represents OFDM symbols, while the vertical direction stands for sub-carriers indices. OFDM symbols are constructed into subframes where the time domain signal $\boldsymbol{x}(t)$ is obtained by applying an inverse Fourier transform (IFFT) on symbols across the subcarriers. Cyclic-prefix (CP) is added for each OFDM symbol. Let N_c denote the number of subcarriers and $N_D + N_K$ be the number of OFDM symbols within a subframe. Here, N_K is the number of OFDM symbols that are used as pilots/reference signals whereas N_D OFDM symbols within the same subframe are used for data transmission. Each element on the resource grids is modulated by quadrature amplitude modulation (QAM). Note that pilots/reference signals are used to conduct CSI estimation at the receiver in modern 4G and 5G networks. Furthermore, N_K **is configured to be smaller than** N_D to reduce over-the-air signaling overhead. Meanwhile, N_K is often designed to be proportional to the number of streams to offer a reliable CSI estimation.

The received signal at the BS can be expressed as

$$\boldsymbol{Y}(t) = r\left(\sum_{\ell=0}^{L} \boldsymbol{\mathcal{H}}(\ell) \times_3 \boldsymbol{x}(t-\ell) + \boldsymbol{\mathcal{N}}(t)\right), \quad (21)$$

where $r(\cdot)$ is a function which characterizes the non-linearity at the receiver, such as ADCs, as well as model mismatch; $\boldsymbol{\mathcal{H}}(\ell) \in \mathbb{C}^{N_a \times N_e \times N_t}$ is a tensor which defines a spatial channel response at the ℓth-delay, where the total number of delays is denoted as L; N_a is the number of azimuth antennas, N_e is the number of elevation antennas of the massive MIMO BS antenna array. Our objective is to train a NN $\mathcal{D}$ which can recover $\boldsymbol{X}(t)$ by using $\boldsymbol{Y}(t)$, i.e.,

$$\mathcal{D}(\boldsymbol{Y}(t)) = \boldsymbol{X}(t), \quad (22)$$

such that the NN is learned by

$$\min_{\mathcal{D}} f\left(\{\mathcal{D}(\boldsymbol{Y}(t))\}, \{\boldsymbol{X}(t)\}\right). \quad (23)$$

B. Online Symbol Detection by Multi-Mode RC

Multi-Mode RC serves as the detection NN, $\mathcal{D}$, through the over-the-air pilots/reference signals on a subframe basis. The pilots defined in existing massive MIMO-OFDM systems of each subframe is directly utilized as the training dataset. The following data symbols in the same subframe is the testing dataset, i.e., we train the NN using N_K pilots to detect N_D data symbols in each subframe. This constraint makes the learning framework different from conventional NNs to enable online learning for robust and adaptive communications. The azimuth direction is set as the first mode of Multi-Mode RC and the elevation direction is set as the second mode. Each OFDM pilot symbol is considered as one training batch. Accordingly, the input sequence length equals the number of subcarriers, N_c, plus CP. The output is then truncated to be a N_c-length sequence following the process as described under equation (13). The symbols of each stream on each subcarrier are obtained through quantization and demodulation.

In massive MIMO systems, symbol detection can be conducted through either a joint or a decomposed approach:

1) Joint Processing: In the joint processing, data symbols are obtained through a single Multi-Mode RC with a multi-head output, where the size of the first and second mode of the RC output sequence are respectively N_q and N_u. The mode order of the output node also can be reversely configured. This is because Multi-Mode RC treats equally on each output mode

according to the generation rule of the internal feature queue as shown in (16). A well trained joint model is anticipated to yield a good symbol detection performance since all interference and imperfect factors are handled jointly.

2) Decomposed Processing: The decomposed approach refers to learning the output weight through a decomposed way. For instance, in the case of 2-mode RC, it has $N_u \times N_q$ pairs of $(\boldsymbol{w}_{out}^{(1)}, \boldsymbol{w}_{out}^{(2)})$ to learn, where the vector weight $\boldsymbol{w}$ maps the internal states to a scalar entry of the output tensor sequence. In this framework, the training on each decomposed output weight is based on their individual loss allowing the training through a parallel manner which can significantly reduce the computation latency. On the other hand, the decomposed method takes extra resources on storage and computation compared to the joint approach. For instance, in 2-mode RC based joint approach, the size of output weight matrices $\boldsymbol{W}_{out}^{(1)}$ and $\boldsymbol{W}_{out}^{(2)}$ are respectively $N_{out}^{(1)} \times N_f^{(1)}$ and $N_{out}^{(2)} \times N_f^{(2)}$. In the decomposed way, the shapes of output mapping on each mode are respectively $1 \times N_f^{(1)}$ and $1 \times N_f^{(2)}$. Thus, there are $N_{out1} N_{out2} \times (N_f^{(1)} + N_f^{(2)})$ output weights in total for the decomposed approach which is higher than $N_{out}^{(1)} \times N_f^{(1)} + N_{out}^{(2)} \times N_f^{(2)}$ from the joint way. Overall, the map from MIMO-OFDM parameters to the Multi-Mode RC parameters is summarized in Table III.

V. Performance Evaluations

This section provides performance evaluations of the introduced Multi-Mode RC for uplink symbol detection in a multi-user massive MIMO-OFDM scenario. We choose uncoded bit error rate (BER) as the quantitative metric to evaluate the reliability of the underlying link. Table III contains simulation parameters of the massive MIMO-OFDM system. The default system configuration is: $N_a = 8$, $N_e = 8$, $N_c = 512$, $N_{cp} = 32$, $N_q = 2$, $N_u = 2$, $N_K = 4$ and $N_D = 12$. Note that in this setting, the pilots/reference signals overhead is 25% which is inline with the 3GPP 5G NR standard [45]. Furthermore, it is important to note that the underlying massive MIMO channels are generated strictly according to the clustered delay line (CDL) model specified in 3GPP Technical Report (TR) 38.901 to ensure the corresponding performance evaluation is relevant and practical. To be specific, the transmitter and receiver are configured with uniform linear arrays having half-wavelength antenna spacing, and the power delay profile is configured with a cluster delay rate of 3. The maximum delay spread of the channel is set as the length of the CP in OFDM. Each obtained BER point is collected over 100 consecutive subframes. SNR is defined as the average power ratio between noise-free received signal and the additive noise. The configuration of the Multi-Mode RC is set as follows: $T' = N_{CP}$, $N_s = 8$, the number of ALS iterations is set as 6, the state transition matrix $\boldsymbol{W}_{tran}$ on each mode is independently generated with its spectral radius less than 1. Meanwhile, the input weights matrix $\boldsymbol{W}_{in}$ is generated independently for each mode from a uniform distribution on $[-1, 1]$.

A. Parsing Multi-Mode RC

We first parse various components of a Multi-Mode RC to offer more insights on the underlying NN structure. We consider three types of Multi-Mode RC in the evaluation: 1) Our introduced one; 2) Our introduced one without using the tensor permutation to construct the feature queue in (16); 3) Our introduced one using a large number of N_s, where $N_s = 128$. Fig. 7 and Fig. 8 respectively show the training and testing BERs under different numbers of iteration when $SNR = 15$dB. As we can observe that without tensor permutation in the feature queue, the RC performs underfitting to the task. This is because the feature from different modes of the input tensor sequence has not been equally extracted. Meanwhile, if we increase the number of neurons, the NN model complexity increases. Accordingly, it "overfits" the training data as the training BER dramatically decreases, whereas the testing BER increases.

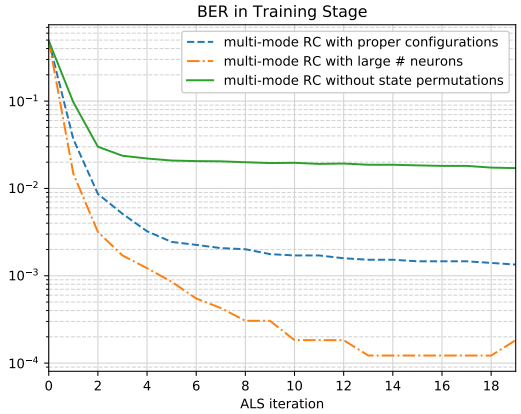


Fig. 7. Training BER of multi-mode RC with respect to iterations in ALS.

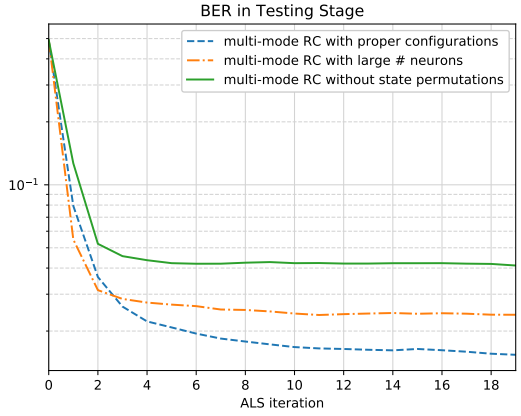


Fig. 8. Testing BER of Multi-Mode RC with respect to iterations in ALS.

B. Uniqueness Conditions

Now, we investigate how the uniqueness condition defined in Theorem 1 determines the training and testing performance. For convenience, we set $N_c = 64$ and use single-batch based

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

TABLE III
NOTATIONS OF MULTI-MODE RC BASED MIMO-OFDM SYMBOL DETECTION

Notations	Definition	Corresponding notations in Multi-Mode RC
N_t	Number of antennas stacked from all users	N/A
N_u	Number of users	$N_{out}^{(1)}$ in joint processing
N_q	Number of antennas at each user	$N_{out}^{(2)}$ in joint processing
N_a	Number of azimuth antennas	$N_{in}^{(1)}$
N_e	Number of elevation antennas	$N_{in}^{(2)}$
N_c	Number of OFDM sub-carriers	N_T
N_K	Number of pilot symbols in a frame	N_K-Training batches (if use multi-batch training)
N_D	Number of data symbols in a frame	Testing batches
$\boldsymbol{x}(t) \in \mathbb{C}^{N_t \times 1}$	Transmitted Signal	Desired output
$\boldsymbol{X}(t) \in \mathbb{C}^{N_u \times N_s}$	Transmitted Signal	Desired output
$\boldsymbol{Y}(t) \in \mathbb{C}^{N_a \times N_e}$	Received Signal	Input
$\boldsymbol{\mathcal{Y}} \in \mathbb{C}^{N_a \times N_e \times T}$	Stacked tensor of received signal	Input

training in this evaluation. According to (20), the critical condition for this task becomes,

$$N_u + N_T N_K == 2N_s + N_a T' + N_e T'$$
$$N_q + N_T N_K == 2N_s + N_a T' + N_e T'$$

When we only change parameters T' and N_s while fixing the rest based on our default setup, the critical conditions become $130 == 8T' + N_s$. This condition is plotted as the dashed red line in Fig. 9 and Fig. 10. Meanwhile, we also plot the contours of the log-loss as well as the BER in the same (N_s, T') plane. As shown in Fig. 9, the loss is guaranteed to be greater than a threshold, e.g., -4.00, when the uniqueness condition holds. On the other hand, when the condition is violated, the loss tends to be close to zero. In this case, the RC model overly fits to the training data which brings a high risk of over-fitting. This result is also consistent with the BER contour plotted in Fig. 9. Note that even though satisfying the uniqueness condition can potentially avoid overfitting, it may cause underfitting as we can observe the high BER below the condition line in Fig. 10.

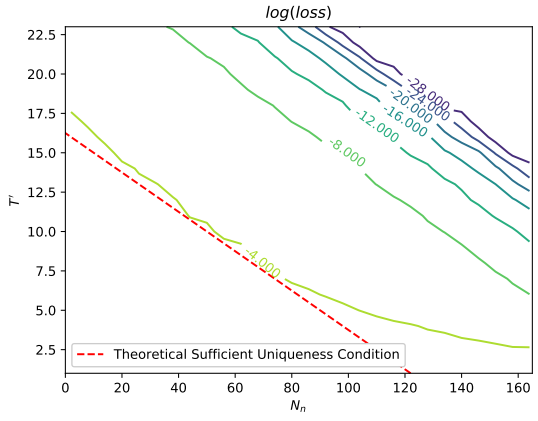


Fig. 9. Log loss contour in (N_s, T') plane in training stage

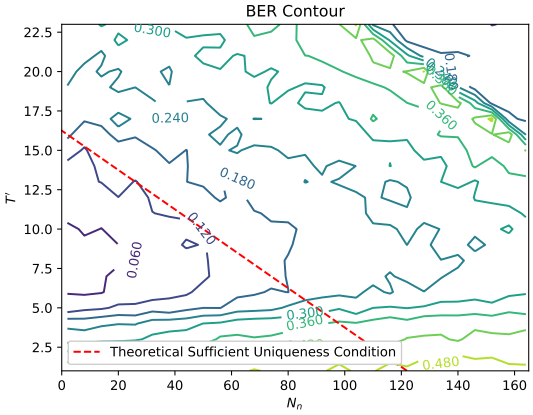


Fig. 10. BER contour in (N_s, T') plane in testing stage

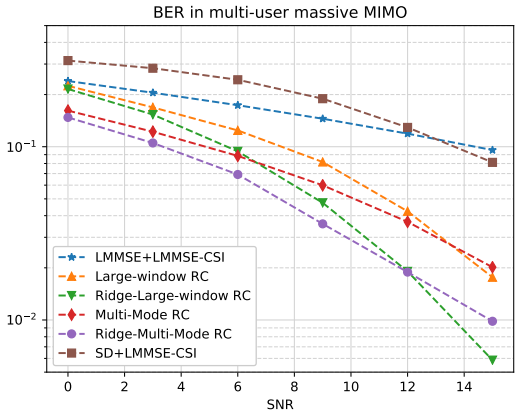


Fig. 11. BER in a multi-user massive MIMO system with 64 antennas at the BS (8×8 antenna array).

C. Comparison with State-of-Art Detection Strategies

We now investigate the BER versus SNR using different approaches. In Fig. 11, the compared methods are: 1) LMMSE+LMMSE-CSI which uses linear minimum mean square error (LMMSE) based symbol detection under the LMMSE estimated CSI. 2) SD+LMMSE-CSI which uses sphere decoding for symbol detection based on the LMMSE estimated CSI. 3) Large-window RC refers to the windowed echo state network (WESN) introduced in [15] by vectorizing the input as a vector and setting the input buffer size as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

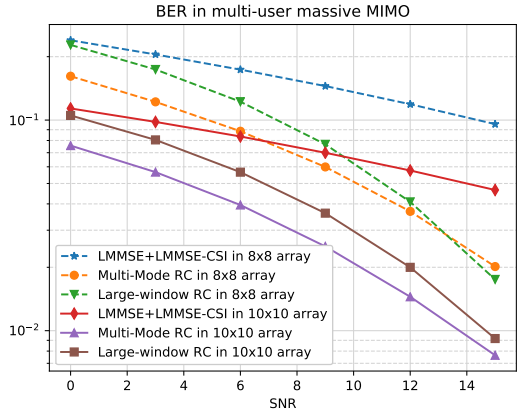


Fig. 12. BER in multi-user massive MIMO with 64 (8×8) and 100 (10×10) antennas at the BS.

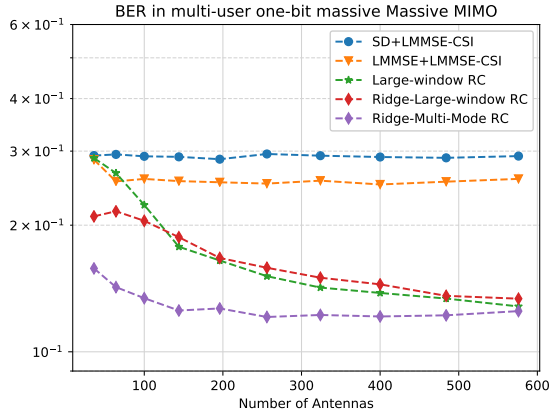


Fig. 13. BER in one-bit multi-user massive MIMO systems with different antenna numbers.

TABLE IV
CPU TIME OF DIFFERENT ALGORITHMS FOR SYMBOL DETECTION AT SNR = 15 DB

Approaches	Training\Channel Estimation	Testing\Symbol Detection
LMMSE+LMMSE-CSI	0.15s	0.10s
SD+LMMSE-CSI	0.15s	13.625s
Large-Window RC	127.87 s	16.39 s
Multi-Mode RC	71.84 s	7.812s

52. 4) Ridge-large-window RC refers to the same WESN but using l_2 norm as a penalty term to the output weights in the loss function. 5) Multi-Mode RC is the introduced method. 6) Ridge-Multi-Mode RC standards for the same Multi-Mode RC but also adding a l_2 norm as the regularization on the output weights in the loss objective. Fig. 11 clearly demonstrates the performance gain of the Multi-Mode RC over the signal processing-based methods (LMMSE+LMMSE-CSI and SD+LMMSE-CSI). Meanwhile, we can see that the Multi-Mode RC is more robust than the single-mode RC as the multi-mode feature of MIMO-OFDM signals is leveraged for the symbol detection.

In addition, we investigate the BER performance by increasing the array size. Fig. 12 shows that when the number of antennas increases (e.g improve the antenna array from 8×8 to 10×10), the BER curves of all methods are improved. On the other hand, the Multi-Mode RC continues showing its advantage over other methods. We also show the CPU time of different algorithms under the 8×8 BS antenna scenario on a desktop with Intel i5-9400 CPU @ 2.90 Hz and 16G RAM. As shown in Table IV, the LMMSE algorithm is the fattest one since it is implemented by leveraging the highly optimized broadcasting feature in the Python Numpy package. We want to highlight further that due to limitations on the usage of Python packages, this comparison cannot fully reveal the computational advantage of Multi-Mode RC over conventional methods. However, we can observe that Multi-Mode RC indeed performs faster than traditional RC in terms of the learning speed due to the reduced degree of freedom in its output layers as discussed in Sec. IV.

D. Performance Evaluation under Receiving Non-linearity —- Low Resolution ADCs

To show the advantage of RC-based approach in other model mismatch scenarios, we evaluate the BER performance when low precision quantization is added in the link which is extremely relevant to massive MIMO systems. We consider the extreme case of using one-bit ADC which quantizes the in-phase and quadrature components to 1 or -1. The definition of the quantizer for any one of the components is,

$$q(x) = A_{max} \cdot \text{sign}(x) \tag{24}$$

where A_{max} is the maximum magnitude of the quantizer where we set it as 0.6 in the evaluation. Fig. 13 clearly shows that the Multi-Mode RC is the most robust method in this scenario. Meanwhile, we can observe the saturation phenomena of the BER curve when the antenna number increases. This is because the quantization level on each antenna is set as a fixed value. Intuitively, the performance can be further improved by optimizing the quantization level on each antenna. This will be considered as our future work.

E. Comparison with Model-based Learning Approaches

In this section, we compare the Multi-Mode RC against a state-of-the-art model-based symbol detection NN, MMNet [10]. MMNet is a deep NN structure based on unfolding iterative soft-thresholding algorithms, which adds degrees of flexibility on certain parameters in the NN for training. In our evaluation, MMNet is configured using the aforementioned training dataset associated with the LMMSE-estimated CSI. As the legends shown in Fig. 14, we choose three MMNet operation modes as the benchmark methods: 1) MMNet-Online-I contains only scalar trainable parameters. It assumes the channel additive noise in both testing and training stages are with homogeneous distributions. The "Online" scheme refers to a training framework described in the paper, by which a NN for conducting symbol detection on the first subcarrier is trained from scratch using 1000 iterations and N_K pilot symbols, while other NNs with respect to the remaining sub-carriers are fine-tuned based on the first NN

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2022.3164203, IEEE Transactions on Wireless Communications

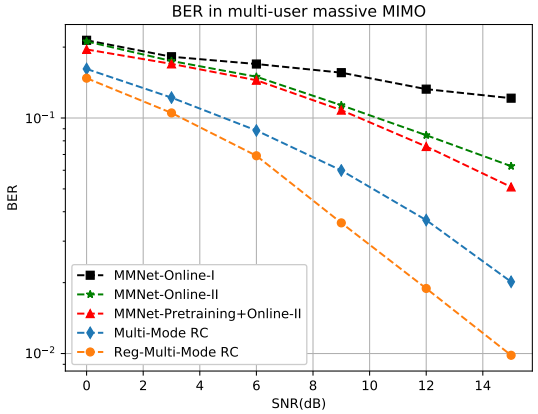


Fig. 14. BER in a multi-user massive MIMO system with 64 antennas at the BS (8×8 antenna array).

with 3 additional iterations using their individual training symbols. 2) MMNet-Online-II has matrix-form parameters as noise variance estimators per layer, which is considered as an advanced structure of MMNet-Online-I using the same training strategy. 3) MMNet-Pretraining-Online-II adds a pre-trained training stage to MMNet-Online II. The NNs are initialized with pre-trained weights using 256 pilot symbols from historical training datasets with different channel realizations.

Note that, while MMNet achieves notable performance on both i.i.d Gaussian channels and spatially-correlated channels in [10], the utilized training symbols are relatively larger than the setup in this paper (e.g. 500 pilot symbols associated with perfect CSIs which is not consistent to the online over-the-air scenario as in our paper). Furthermore, the computation and memory requirements on MMNet are significantly higher than the introduced method. To be specific, MMNet requires N_c NNs to estimate symbols on all subcarriers, each of which stacks 10 layers of neurons. Therefore, the number of training iterations significantly increases. In our method, we only use 4 pilot symbols and a single NN to jointly accomplish the symbol detection task. Our evaluation results demonstrate that the Multi-Mode RC outperforms MMNet in the multi-user massive MIMO scenario with a steep performance improvement slope. Overall, Reg-Multi-Mode RC shows $5-6$ dB gain in SNR compared with MMNet.

VI. Conclusion

In this paper, we presented a NN structure, Multi-Mode RC, for symbol detection in massive MIMO-OFDM systems. We elaborated on the NN architecture and its configuration for the symbol detection task. The introduced Multi-Mode RC framework is shown to be able to effectively cope with the model mismatch, waveform distortion as well as interference in the systems. Numerical results demonstrated the advantages of Multi-Mode RC in the following aspects: It can offer lower BER than conventional single-mode RC frameworks in low SNR regime while achieving reduced computational complexity. Compared to other model-based learning approaches, the introduced method can operate on a subframe-basis thus completely relying on the limited over-the-air pilots/reference symbols. This attractive feature enables us to train the symbol detection task using a compatible signal overhead as modern cellular networks.

In our future work, we will consider the optimization of the quantization thresholds at each antenna port. Since quantization is an irreversible process, adaptive quantization strategies are a promising approach to preserve the waveform information. Furthermore, incorporating gradient-free learning algorithms into RC framework is another interesting direction.

Appendix
Proof of Theorem 1

Lemma 1: Given two full-rank matrices $\boldsymbol{X} \in \mathbb{C}^{M_1 \times M_2}$ and $\boldsymbol{G} \in \mathbb{C}^{H \times M_2}$, where $H \geq M_1$, the following least squares problem has a unique solution if and only if $rank(\boldsymbol{X}) = rank(\boldsymbol{W}^\star)$, where $\boldsymbol{W}^\star$ is the optimum.

$$\min_{\boldsymbol{W}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{G}\|_F \tag{25}$$

Proof: Based on the assumptions, we have $rank(\boldsymbol{X}) = \min\{M_1, M_2\}$ and $rank(\boldsymbol{G}) = \min\{H, M_2\}$. In general, (25) has a unique solution if and only if $rank(\boldsymbol{G}) = H$ [46].

$rank(\boldsymbol{G}) = H$ implies $M_2 \geq H$. Accordingly, the solution is given by $\boldsymbol{X}\boldsymbol{G}^+$, where $\boldsymbol{G}^+$ represents the Moor-Penrose inverse of matrix $\boldsymbol{G}$. Since $\boldsymbol{X}$ is with full row rank, $\boldsymbol{G}^+$ is with full column rank and $H \geq M_1$, we have $rank(\boldsymbol{X}\boldsymbol{G}^+) = rank(\boldsymbol{X})$. Thus, we have $rank(\boldsymbol{W}^\star) = rank(\boldsymbol{X}\boldsymbol{G}^+) = rank(\boldsymbol{X})$.

On the contrary, suppose $rank(\boldsymbol{W}^\star) = rank(\boldsymbol{X})$ holds. We have $H \leq M_2$, otherwise there exists $\boldsymbol{W}^\star + \boldsymbol{H}$ which is also a minimum of (25), where $\boldsymbol{H}$ is a non-zero solution of $\boldsymbol{H}\boldsymbol{G} = \boldsymbol{0}$. Since $\boldsymbol{G}$ is assumed with full rank, we have $rank(\boldsymbol{G}) = H$. ■

Lemma 2: Given the same assumption as Lemma 1, the sufficient and necessary condition for the uniqueness of (25) can be expressed as

$$M_2 \geq H \tag{26}$$

Proof: Since the necessary and sufficient condition for the uniqueness is $rank(\boldsymbol{G}) = H$. Therefore, the uniqueness condition can be alternatively written as (25) is $M_2 \geq H$. ■

Before proceeding on the proof of Theorem 1, we introduce the following concepts to characterize rank properties of tensor [40].

Definition 1: The mode-n rank of a tensor $\boldsymbol{\mathcal{G}}$ is the mode-n unfolding of $\boldsymbol{G}$, i.e., $\boldsymbol{G}_{(n)}$.

Definition 2: A N-order tensor $\boldsymbol{\mathcal{G}}$ is with rank-$(M_1, M_2, \cdots, M_N)$ when its mode-1 rank, mode-2 rank to mode-N rank are equal to M_1, M_2, and M_N, respectively.

Lemma 3: Given $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_N}$ and $\boldsymbol{\mathcal{G}} \in \mathbb{C}^{H_1 \times H_2 \times \cdots \times H_N}$, which are with rank-$(M_1, M_2, \cdots, M_N)$ and rank-$(H_1, H_2, \cdots, H_N)$ respective, and $H_n \geq M_n$, $N > 2$, the minimization of

$$\min_{\boldsymbol{W}_1, \cdots, \boldsymbol{W}_N} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \cdots \times_N \mathbf{W}_N\|_F \tag{27}$$

is unique if

$$rank(\boldsymbol{X}_{(n)}) = rank(\boldsymbol{W}_n^\star) \tag{28}$$

where $\boldsymbol{W}_1^\star, \cdots, \boldsymbol{W}_N^\star$ is the optimum.

Proof: We prove this theorem by mathematical induction. According to Lemma 1, the uniqueness holds for $\min_{\boldsymbol{W}_1} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W}_1\|_F$. Then, we assume it holds for N order tensor $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{G}}$ with $N-1$ factor matrices.

Now, we consider the case with N factor matrices,

$$\begin{aligned}&\min_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_N} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \cdots \times_N \mathbf{W}_N\|_F \\ &= \min_{\boldsymbol{W}_N}\{\min_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{N-1}} \|\boldsymbol{X}_{(N)} - \mathbf{W}_N \boldsymbol{G}_{(N)}(\mathbf{W}_1 \otimes \cdots \otimes \mathbf{W}_{N-1})^T\|_F\}\end{aligned}$$

We denote $\boldsymbol{W}_N^\star$ as one of the optima of the above problem. Since $\boldsymbol{W}_N^\star$ is with full row rank based on the assumption, we have tensor $\boldsymbol{\mathcal{G}} \times_N \boldsymbol{W}_N^\star$ with rank-$(H_1, H_2, \cdots, M_N)$. This is because $\boldsymbol{G}_{(n)}$ is with full row rank. Therefore, $(\boldsymbol{W}_1^\star, \cdots, \boldsymbol{W}_{N-1}^\star)$ as an optimum of $\min_{\boldsymbol{W}_1,\cdots,\boldsymbol{W}_{N-1}} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_1 \mathbf{W}_1 \times_2 \mathbf{W}_2 \cdots \times_N \mathbf{W}_N^\star\|_F$ is unique based on the inductive assumption. We then assert $\boldsymbol{W}_N^\star$ is unique. Otherwise it contracts to Lemma 1. ∎

Remarks:

- The proof does not conduct the mathematical induction on the tensor mode. This is because the uniqueness does not hold for matrix decomposition, i.e., $\min_{\boldsymbol{W}_1,\boldsymbol{W}_2} \|\boldsymbol{X} - \boldsymbol{W}_1 \boldsymbol{G} \boldsymbol{W}_2^T\|_F$.
- This theorem is only a sufficient condition for the uniqueness of (27).

Theorem 2: Under the same condition as Lemma 3, using ALS to solve (27) by initializing the factor matrices with full rank, the achieved solution is unique when

$$\sum_{i \neq n} M_i \geq H_n.$$

Proof: If we solve the optimization problem (27) through ALS as well as initializing factor matrices as full rank, updated factor matrices are with full rank at each iteration if we assume

$$\sum_{i \neq n} M_i \geq H_n$$

due to Lemma 2. Here, the prerequisite of Lemma 2 is met because in the updating rule for $\boldsymbol{W}_n$, $rank(\boldsymbol{G}_{(n)}(\mathbf{W}_1 \otimes \cdots \mathbf{W}_{n-1} \otimes \mathbf{W}_{n+1} \otimes \cdots \otimes \mathbf{W}_N)^T) = H_n$ is guaranteed by the full rank initialization assumption on factor matrices. Therefore, when ALS terminated at an optimum of (27), we can assert this optimum is the unique by using Lemma 3. ∎ With minor revisions on the statement of the above Theorem, we can arrive at Theorem 1.

Appendix
Low Complexity Factor Matrix Calculation in ALS

At each step of using alternating least squares to solve the factor matrices $\boldsymbol{W}_{out}^{(1)}$ and $\boldsymbol{W}_{out}^{(1)}$ in (12), suppose we directly solve the following sub-problem to obtain $\boldsymbol{W}_{out}^{(1)}$,

$$\boldsymbol{W}_{out}^{(1)} = \arg\min_{\boldsymbol{W}_{out}^{(1)}} \|\boldsymbol{Z}_{(1)} - \boldsymbol{W}_{out}^{(1)} \boldsymbol{G}_{(1)} (\boldsymbol{W}_{out}^{(2)} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K})^T\|_F.$$

The resulting memory costs on $\boldsymbol{W}_{out}^{(2)} \otimes \boldsymbol{I}_{N_T} \otimes \boldsymbol{I}_{N_K}$ are very large. To cope with the bottleneck from memory units, we alternatively solve $\boldsymbol{W}_{out}^{(1)}$ via the following approach,

$$\begin{aligned}\boldsymbol{W}_{out}^{(1)} &= \arg\min_{\boldsymbol{W}_{out}^{(1)}} \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W}_{out}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)}\| \\ &\overset{(a)}{=} \arg\min_{\boldsymbol{W}_{out}^{(1)}} \|\boldsymbol{\mathcal{Z}} - \sum_k \boldsymbol{\mathcal{G}}^{(k)} \times_1 \boldsymbol{W}_{out}^{(1)}(k) \times_2 \boldsymbol{W}_{out}^{(2)}(k)\| \\ &= \arg\min_{\boldsymbol{W}_{out}^{(1)}} \|\boldsymbol{Z}_{(1)} - \sum_k \boldsymbol{W}_{out}^{(1)}(k) (\boldsymbol{\mathcal{G}}^{(k)} \times_2 \boldsymbol{W}_{out}^{(2)}(k))_{(1)}\| \\ &= \arg\min_{\boldsymbol{W}_{out}^{(1)}} \|\boldsymbol{Z}_{(1)} - \boldsymbol{W}_{out}^{(1)} [(\boldsymbol{\mathcal{G}}^{(1)} \times_2 \boldsymbol{W}_{out}^{(2)}(1))_{(1)}, \cdots, (\boldsymbol{\mathcal{G}}^{(K)} \times_2 \boldsymbol{W}_{out}^{(2)}(K))_{(1)}]^T\|\end{aligned}$$

where (a) comes from the partition definition of Tucker decomposition in (4), $\boldsymbol{W}^{(1)}(k)$ is the kth sub-block of matrix $\boldsymbol{W}^{(1)}$, and $K := N! + 1$. As the above calculation suggests, we can first calculate mode-2 product between each partitioned core tensor and factor matrix, then concatenate them as a big matrix to calculate a pseudoinverse to reach a least squares solution of $\boldsymbol{W}_{out}^{(1)}$. Similar tricks can be apply to calculate $\boldsymbol{W}_{out}^{(2)}, \cdots, \boldsymbol{W}_{out}^{(N)}$ in general multi-mode RC.

References

[1] T. L. Marzetta, "Massive MIMO: an introduction," *Bell Labs Tech. J.*, vol. 20, pp. 11–22, 2015.

[2] X. Gao, O. Edfors, F. Rusek, and F. Tufvesson, "Massive MIMO performance evaluation based on measured propagation data," *IEEE Trans. Wireless Commun.*, vol. 14, no. 7, pp. 3899–3911, 2015.

[3] K. T. Truong and R. W. Heath, "Effects of channel aging in massive MIMO systems," *J. of Commun. and Netw.*, vol. 15, no. 4, pp. 338–351, 2013.

[4] 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 Trans. Wireless Commun.*, vol. 27, no. 2, pp. 212–217, 2020.

[5] J. Vieira, S. Malkowsky, K. Nieman, Z. Miers, N. Kundargi, L. Liu, I. Wong, V. Öwall, O. Edfors, and F. Tufvesson, "A flexible 100-antenna testbed for massive mimo," in *2014 IEEE Globecom Wkshps*, 2014, pp. 287–293.

[6] Z. Zhou, S. Jere, L. Zheng, and L. Liu, "Learning with knowledge of structure: A neural network-based approach for MIMO-OFDM detection," in *2020 54th Asilomar Conf. on Signals, Syst., and Comput.*

[7] E. Björnson, E. G. Larsson, and T. L. Marzetta, "Massive MIMO: Ten myths and one critical question," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 114–123, 2016.

[8] A. N. Gorban and I. Y. Tyukin, "Blessing of dimensionality: mathematical foundations of the statistical physics of data," *Philosophical Trans. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, no. 2118, p. 20170237, 2018.

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

[10] 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, 2020.

[11] H. Ye, G. Y. Li, and B. 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, 2018.

[12] 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, 2012.

[13] Z. Zhou, S. Jere, L. Zheng, and L. Liu, "Learning for integer-constrained optimization through neural networks with limited training," in *NeurIPS Wkshps on Learn. Meets Combinatorial Algorithms*, Dec. 2020.

[14] 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.

[15] 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, 2020.

[16] Z. Zhou, L. Liu, V. Chandrasekhar, J. Zhang, and Y. Yi, "Deep reservoir computing meets 5G MIMO-OFDM systems in symbol detection," in *AAAI Conf. on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1266–1273.

[17] Z. Zhou, L. Liu, S. Jere, J. C. Zhang, and Y. Yi, "RCNet: incorporating structural information into deep RNN for MIMO-OFDM symbol detection with limited training," *IEEE Trans. Wireless Commun.*, Jan. 2021.

[18] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *NeurIPS*, 2015, pp. 442–450.

[19] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. on Scientific Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[20] V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky, "Speeding-up convolutional neural networks using fine-tuned cp-decomposition," in *ICLR*, 2015.

[21] L. Cheng, Y. Wu, J. Zhang, and L. Liu, "Subspace identification for DOA estimation in massive/full-dimension MIMO systems: Bad data mitigation and automatic source enumeration," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 5897–5909, 2015.

[22] R. Shafin, L. Liu, J. Zhang, and Y. Wu, "DoA estimation and capacity analysis for 3-D millimeter wave massive-MIMO/FD-MIMO OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 10, pp. 6963–6978, 2016.

[23] R. Shafin, L. Liu, Y. Li, A. Wang, and J. Zhang, "Angle and delay estimation for 3-D massive MIMO/FD-MIMO systems based on parametric channel modeling," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5370–5383, 2017.

[24] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 7, pp. 1524–1538, 2017.

[25] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and S. Li, "Channel estimation for millimeter-wave multiuser MIMO systems via parafac decomposition," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7501–7516, 2016.

[26] R. Shafin and L. Liu, "Multi-cell multi-user massive FD-MIMO: Downlink precoding and throughput analysis," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 487–502, 2019.

[27] Z. Zhou, L. Liu, and J. Zhang, "FD-MIMO via pilot-data superposition: Tensor-based DOA estimation and system performance," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 931–946, 2019.

[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[29] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "Tensorly: Tensor learning in python," *J. of Machine Learning Research*, vol. 20, no. 26, 2019.

[30] P. Comon, "Tensors : A brief introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 44–53, 2014.

[31] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[32] A. Jalalvand, G. Van Wallendael, and R. Van de Walle, "Real-time reservoir computing network-based systems for detection tasks on visual contents," in *2015 7th Intl. Conf. on Comp. Intell., Commun. Syst. and Netw.* IEEE, pp. 146–151.

[33] Z. Tong and G. Tanaka, "Reservoir computing with untrained convolutional neural networks for image recognition," in *2018 24th Intl. Conf. on Pattern Recognition (ICPR)*. IEEE, pp. 1289–1294.

[34] F. Triefenbach, A. Jalalvand, B. Schrauwen, and J.-P. Martens, "Phoneme recognition with large hierarchical reservoirs," *Advances in Neural Info. Process. Syst.*, vol. 23, pp. 2307–2315, 2010.

[35] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Reservoir-based techniques for speech recognition," in *The 2006 IEEE Intl. Joint Conf. on Neural Netw. Proceedings*. IEEE, pp. 1050–1053.

[36] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Sci.*, vol. 304, no. 5667, pp. 78–80, 2004.

[37] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Netw.: Tricks of the trade*. Springer, 2012, pp. 659–686.

[38] Z. Allen-Zhu and Y. Li, "What can ResNet learn efficiently, going beyond kernels?" in *NeurIPS*, vol. 32, 2019, pp. 9017–9028.

[39] L. De Lathauwer and D. Nion, "Decompositions of a higher-order tensor in block terms—Part III: Alternating least squares algorithms," *SIAM J. on Matrix Anal. and Appl.*, vol. 30, no. 3, pp. 1067–1083, 2008.

[40] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms—Part II: Definitions and uniqueness," *SIAM J. on Matrix Anal. and Appl.*, vol. 30, no. 3, pp. 1033–1066, 2008.

[41] R. Shafin, L. Liu, J. Ashdown, J. Matyjas, M. Medley, B. Wysocki, and Y. Yi, "Realizing green symbol detection via reservoir computing: An energy-efficiency perspective," in *2018 IEEE Intl. Conf. Commun. (ICC)*, pp. 1–6.

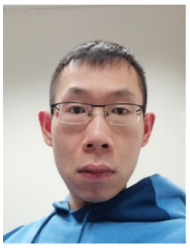
[42] C. Zhao, B. T. Wysocki, C. D. Thiem, N. R. McDonald, J. Li, L. Liu, and Y. Yi, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 4, pp. 265–276, 2016.

[43] F. Nowshin, L. Liu, and Y. Yi, "Energy efficient and adaptive analog ic design for delay-based reservoir computing," in *2020 IEEE 63rd Intl. Midwest Symp. Circuits and Syst. (MWSCAS)*, 2020, pp. 592–595.

[44] K. Hamedani, L. Liu, and Y. Yi, "Energy efficient MIMO-OFDM spectrum sensing using deep stacked spiking delayed feedback reservoir computing," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 1, pp. 484–496, 2021.

[45] *Physical channels and modulation in NR*, 3GPP TS TS 38.211, Rev. 16.0.0, 2020.

[46] C. D. Meyer, *Matrix analysis and applied linear algebra*. SIAM, 2000, vol. 71.



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 joined the Bradley Department of Electrical and Computer Engineering at 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 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. Currently, he is a Professor in the ECE Department of Virginia Tech (VT) and is also the Associate Director of Wireless@VT. He spent more than four years working in the Mitsubishi Electric Research Laboratory (MERL) and the Standards and Mobility Innovation Lab, Samsung Research America (SRA), where he received Global Samsung Best Paper Award in 2008 and 2010. He was leading Samsung's efforts on multiuser MIMO, CoMP, and HetNets in 3GPP LTE/LTE-Advanced standards. His general research interests lie in enabling technologies for Beyond 5G/6G networks including machine learning for wireless networks, massive MIMO, massive MTC communications, and mmWave communications. Dr. Liu's research received 8 Best Paper Awards. In 2021, he received VT College of Engineering Dean's Award for Excellence in Research.

Jiarui Xu received the B.S. degree in Telecommunication Engineering from 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 in the Bradley Department of Electrical and Computer Engineering (ECE) at Virginia Tech, Blacksburg, VA, USA. Her research interests include deep learning, machine learning, and their applications in wireless communications.