Improving Massive MIMO Message Passing Detectors with Deep Neural Network

Xiaosi Tan, Weihong Xu, Kai Sun, Yunhao Xu, Yair Be'ery, *Senior Member, IEEE*, Xiaohu You, *Fellow, IEEE*, and Chuan Zhang, *Member, IEEE*

***Abstract*—In this paper, deep neural network (DNN) is utilized to improve message passing detectors (MPDs) for massive multiple-input multiple-output (MIMO) systems. A general framework to construct DNN architecture for MIMO detection is first introduced by unfolding iterative MPDs. DNN MIMO detectors are then proposed based on modified MPDs including damped BP, max-sum (MS) BP, and simplified channel hardening-exploiting message passing (CHEMP). The correction factors are optimized via deep learning for better performance. Numerical results demonstrate that, compared with the state-of-the-art (SOA) detectors including MMSE, BP, and CHEMP, the proposed DNN detectors can achieve better bit-error-rate (BER) and improve robustness against various antenna and channel conditions with similar complexity. The DNN is required to be trained only once and can be reused for multiple detections, which assures its high efficiency. The corresponding hardware architecture is also proposed. Implementation results with 65 nm CMOS technology approve the efficiency and flexibility of the proposed DNN framework, and show advantages over the SOA in terms of xxx.**

***Index Terms*—Massive MIMO detection, message passing detector (MPD), deep neural network (DNN), low-complexity training, VLSI implementation**

I. Introduction

WITH the rapid traffic growth in wireless communications, systems using multiple-input multiple-output (MIMO) configurations with a large number of antennas have attracted broad attentions in both academia and industry [1]. Due to its increased data rate, higher spectral efficiency, enhanced link reliability and coverage over conventional MIMO [2], [3], massive MIMO becomes one key technology for 5G wireless systems and beyond. However, its large scale brings high pressure to signal detection in terms of computational complexity. In recent years, deep learning (DL) has been applied to various communication problems, which achieves superior results compared to conventional methods [4]–[6]. The goal of this paper is to utilize DL in MIMO detections to propose a deep neural network (DNN)-aided massive MIMO detector.

A. Message Passing MIMO Detectors

Many massive MIMO detection methods have been presented, e.g., [7]–[11], among which the message passing detectors (MPDs)[1], including belief propagation (BP) [12], approximate message passing (AMP) [13], and channel hardening-exploiting message passing (CHEMP) detectors [14], have been paid intensive attentions and broadly researched in recent years. MPDs provide superior performance over the aforementioned detection algorithms due to its lower complexity, stronger robustness, and better detection performance as the MIMO dimension increases [12], [14].

However, the following drawbacks still hinder the practical applications of MPDs:

1) Loopy factor graph (FG): The FG defined by typical MIMO channels are fully-connected, hence heavily loopy. The bit-error-rate (BER) performance of methods like BP suffers severe degradation due to the loopiness.
2) High complexity: MPDs are still of high complexity which implies large delay and implementation difficulties. This issue will become critical for some delay sensitive applications.

To this end, some modifications of the MPDs are proposed to address the aforementioned two issues. The key methods include:

1) Message Damping: Damping is an efficient way to compensate the poor performance caused by the loopy FGs. It is carried out by simply averaging the two successive messages with damping factors. It is observed in many works, e.g., [12], [15]–[17], that damping can accelerate the convergence rate of the iterative algorithms and enhance the BER performance.

- Challenges: The optimal damping factors are difficult to find. The available method relies on Monte Carlo simulations, which brings overwhelming computational burden. In [18], a heuristic automatic damping (HAD) method is proposed to adaptively calculate the damping factor, which improves the efficiency but still requires extra calculations.

2) Reduced-Complexity Approximations: To make the MPDs more computationally affordable and implementation-friendly, various reduced complexity approximations are considered ([19]–[22]). In [19], a max-sum (MS) algorithm is proposed to reduce the complexity of BP by approximating the messages from symbol nodes. The normalized MS (NMS) and offset MS (OMS) are presented as an extension of MS in order to compensate the performance loss resulting from MS approximation. Also, an approximate scheme of CHEMP is

X. Tan, W. Xu, K. Sun, Y. Xu, X. You and C. Zhang are with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. Email: chzhang@seu.edu.cn. *(Corresponding author: Chuan Zhang.)*

Y. Be'ery is with the School of Electrical Engineering, Tel-Aviv University, Tel-Aviv, Israel.

[1] In many works, the abbreviation MPD specifies the detector proposed in [14]. However, in this paper, MPD is generalized to denote all kinds of message passing detectors.

proposed in [21], which estimates the Gaussian variance with a constant coefficient to avoid the expensive updates.

- Challenges: The correction factors and estimation coefficients greatly influence the BER results, however, are difficult to decide. [19] provides a method to update the factors based on approximate prior probabilities and precomputed errors, which requires extra computations per iteration. [21] heuristically decides the optimal coefficients by trials, and therefore lacks robustness against varying application scenarios.

Overall, the enhancements achieved by the modified MPDs ([19]–[22]) rely on the selection of correction factors. Further improvements of these MPDs are demanded for:

- A unified framework to optimize the correction factors efficiently with acceptable computational complexity;
- Improved robustness against different antenna configurations and varying channel conditions;
- Outperming or leveling linear detectors under various antenna configurations and modulations.

B. Deep Neural Network

With the advances in big data, optimization algorithms , and increased computing resources, DNN is currently the state-of-the-art (SOA) in various areas including speech processing [23], game playing [24], and computer vision [25]. In recent years, DNN has also been purposed for communication problems. For instance, various channel decoders using DNN are proposed [26]–[28]. There are also many works on learning to invert linear channels and reconstruct signals [29]–[31]. DNN has also been proposed as a blackbox to construct an end-to-end communication system in [32], [33]. In the context of massive MIMO detection, relevant researches have also been done [34]–[40]. In [34], a DNN MIMO detector named DetNet is derived by unfolding a projected gradient descent method. The work in [35] is based on virtual MIMO blind detection clustered WSN system and applies improved hopfield neural network (HNN) blind algorithm to this system. Also, DNN has been applied for symbol detection in MIMO-OFDM systems as introduced in [36], [37].

In particular, one promising approach to design deep architectures is by unfolding an existing iterative algorithm [29]. Each iteration is considered a layer and the algorithm is called a network. The learning begins with the existing algorithm as an initial starting point and uses optimization methods to find optimal parameters and improve the algorithm. From this point of view, the DL techniques provide a powerful tool to decide the optimal correction factors for the modified MPDs to achieve improved performance. Existing trials in this field, e.g. [38]–[40], have shown the ability of this method to achieve efficient DNN MIMO detection.

C. Contributions

In this paper, to the best knowledge of the authors, an efficient DNN MIMO detector based on MPDs including BP and CHEMP is proposed. Main contributions of this work are:

- A unified framework to design a DNN MIMO detector is proposed by unfolding iterations in MPDs. The mapping schemes between DNN and iterative algorithms are provided with details.
- A reduced complexity modification of CHEMP called simplified MPD (sMPD) is proposed. Three DNN MIMO detectors are then introduced based on the damped BP, MS BP, and sMPD, respectively. The optimal correction factors are decided via training.
- Training method is discussed with details. We show the ability of the proposed DNN detectors to handle multiple antenna/channel conditions with one single training.
- Numerical results are presented to show the advanced performance and robustness of the DNN detectors compared with SOA MPDs and linear detectors as minimum mean-squared error (MMSE) in various antenna/channel configurations.
- The computational complexity of the DNN detectors is discussed. For online detections, the DNN detectors achieve improved performance at similar complexity as the SOA methods.
- The ASIC implementation of the DNN-sMPD detector is designed. The implementation results are also detailed.

D. Paper Outline

The remainder of this paper is organized as below. Backgrounds of MPDs are introduced in Section II. In Section III, modified MPDs including damped BP, MS BP and sMPD are introduced. In Section IV, we present the corresponding DNN MIMO detectors based on the modified MPDs. Section V shows details of the proposed DNN detector, including the training procedure, numerical simulation results, and computational complexity. Section VI gives the hardware architecture of the proposed DNN detector while Section VII concludes the paper.

E. Notations

Throughout the paper, we use the following notations. Lowercase letters (e.g., x) denote scalars, bold lowercase letters (e.g., $\mathbf{x}$) denote column vectors, and bold uppercase letters (e.g., $\mathbf{X}$) denote matrices. Also, the symbol $\mathbf{I}$ denotes the identity matrix; $\log(\cdot)$ denotes the natural logarithm; and $\mathcal{CN}(\mathbf{x}, \sigma^2)$ denotes the complex Gaussian function.

II. Preliminary

A. MIMO System Model

In this paper, we consider a MIMO system with M transmitting and N receiving antennas. Each user sends an independent data stream and the base station detects the spatially multiplexed data through MIMO detection. The received signal vector, $\mathbf{y} \in \mathbb{C}^{N\times 1}$, reads

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x} \in \Theta^M$ is the transmitted symbol vector, with the constellation $\Theta = \{s_1, s_2, \ldots, s_K\}$, K is determined by modulation mode; $\mathbf{n}$ is the additive white Gaussian noise (AWGN) following $\mathcal{CN}(0, \sigma^2 \mathbf{I}_N)$; $\mathbf{H}$ denotes the channel matrix which can be described by the Kronecker model

$$\mathbf{H} = \mathbf{R}_r^{\frac{1}{2}} \mathbf{H}_w \mathbf{R}_t^{\frac{1}{2}} \tag{2}$$

according to [41], where $\mathbf{R}_r$ and $\mathbf{R}_t$ are the antenna correlation matrices at the receiver and transmitter side respectively, and $\mathbf{H}_w$ is i.i.d Rayleigh-fading channel matrix following independent Gaussian distribution.

B. Message Passing MIMO Detectors

1) BP Detector: MIMO systems can be modeled by a FG as in Fig. 1 according to [42]. BP forms a message-passing scheme by allowing observation nodes to transfer belief information with symbol nodes back and forth to iteratively improve the reliability for decision. The message updating at observation and symbol nodes at the l-th iteration is summarized in the following:

- Symbol nodes:

$$\alpha_{ij}^{(l)}(s_k) = \sum_{t=1, t\neq j}^{N} \beta_{ti}^{(l-1)}(s_k), \tag{3}$$

$$p_{ij}^{(l)}(x_i = s_k) = \frac{\exp(\alpha_{ij}^{(l)}(s_k))}{\sum_{m=1}^{K} \exp(\alpha_{ij}^{(l)}(s_m))}, \tag{4}$$

- Observation nodes:

$$\beta_{ji}^{(l)}(s_k) = \log \frac{p^{(l)}(x_i = s_k | y_j, \mathbf{H})}{p^{(l)}(x_i = s_1 | y_j, \mathbf{H})}, \tag{5}$$

where α_{ij} denotes the prior log-likelihood ratio (LLR), β_{ji} denotes the posterior LLR and p_{ij} is the prior probability of each symbol. The soft output after L iteration is given by

$$\gamma_i(s_k) = \sum_{t=1}^{N} \beta_{ti}^{(L)}(s_k), \tag{6}$$

and the s_k maximizes $\gamma_i(s_k)$ is chosen as the final decision of the transmitted signal. Details of BP are demonstrated in [12].

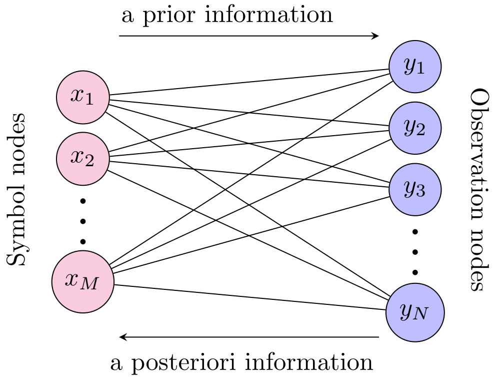


Fig. 1. Factor Graph of a massive MIMO system.

Since the FG defined by the dense MIMO channel matrix $\mathbf{H}$ is loopy as shown in Fig. 1, BP is not guaranteed to converge to the MAP. Antenna correlation can even aggravate the loopy effect due to the less randomness in channel matrix, which brings severe degradation in BER results [43]. Also, for each iteration, one division operation is needed to calculate the prior messages in Eq. (4), which brings difficulty to hardware implementation.

2) CHEMP Detector: The CHEMP detector proposed in [14] is another low-complexity MPD which exploits the channel hardening phenomenon. Specifically, multiplying Eq. (1) with $\mathbf{H}^T$ while dividing N leads to

$$\mathbf{z} = \mathbf{J}\mathbf{x} + \mathbf{v}, \tag{7}$$

where $\mathbf{z} = \mathbf{H}^T \mathbf{y}/N$, $\mathbf{J} = \mathbf{H}^T \mathbf{H}/N$ and $\mathbf{v} = \mathbf{H}^T \mathbf{n}/N$. The i-th element of $\mathbf{z}$ is $z_i = J_{ii} x_i + \sum_{j=1, j\neq i}^{2M} J_{ij} x_j + v_i = J_{ii} x_i + g_i$, in which $g_i = \sum_{j=1, j\neq i}^{2M} J_{ij} x_j + v_i$ and $v_i = \sum_{j=1}^{2M} H_{ji} n_j / N$. Let $p_j(s_k)$ denote the probability of x_j being $s_k \in \Theta$. We assume g_i to be Gaussian distributed with mean μ_i and variance σ_i^2 such that:

$$\mu_i = \mathbb{E}(g_i) = \sum_{j=1, j\neq i}^{2M} J_{ij} \mathbb{E}(x_j) = \sum_{j=1, j\neq i}^{2M} J_{ij} \sum_{s_k \in \Theta} s_k p_j(s_k), \tag{8}$$

$$\begin{aligned} \sigma_i^2 &= \mathrm{Var}(g_i) = \sum_{j=1, j\neq i}^{2M} J_{ij}^2 \mathrm{Var}(x_j) + \sigma_v^2 \\ &= \sum_{j=1, j\neq i}^{2M} J_{ij}^2 (\sum_{s_k \in \Theta} s_k^2 p_j(s_k) - (\mathbb{E}(x_j))^2) + \sigma_v^2, \end{aligned} \tag{9}$$

where $\sigma_v^2 = \frac{\sigma_n^2}{2N}$. The LLR vector of symbol x_j defined by $L_j = \{L_j(s_1), L_j(s_2), \ldots, L_j(s_K)\}$ can then be calculated by

$$L_j(s_k) = \frac{(2(z_j - \mu_j) - J_{jj}(s_k + s_1))(J_{jj}(s_k - s_1))}{2\sigma_j^2} \tag{10}$$

for $k = 1, 2, \ldots, K$. Besides,

$$p_j(s_k) = \frac{\exp(L_j(s_k))}{\sum_{k=1}^{\sqrt{K}} \exp(L_j(s_k))}. \tag{11}$$

Decision of symbols x_j's are set to be s_k with the maximum probabilities. More details of CHEMP can be found in [14].

CHEMP detector is based on channel-hardening phenomenon, which assumes that the diagonal terms of matrix $\mathbf{J}$ in Eq. (7) is much larger than the off-diagonal elements. Hence, in the above derivation, g_i's that are related to the off-diagonal terms in $\mathbf{J}$ are simplified by a Gaussian approximation. This approximation attains accuracy in systems with large number of antennas [14]. However, when the system size is not large enough, the MPD detector suffers less accurate performance and slow convergence rate [39]. Meanwhile, with higher-order modulation, Eq.s (9) and (11) will also bring difficulties to implementation.

III. Modified Message Passing Detectors

A. Message Damping

Message damping is a judicious option to mitigate the problem of loopy BP without additional complexity [12]-[15]. It can also enhance the convergence rate of CHEMP [14].

Specifically, by applying message damping, the probability $p^{(l)}$ at the l-th iteration can be smoothed as

$$p^{(l)} \leftarrow (1-\delta)p^{(l)} + \delta p^{(l-1)}, \tag{12}$$

where "←" denotes the assignment, $\delta \in [0,1]$ is the damping factor to make a weighted average of the current calculated messages and the previous ones.

Eq. (12) can be applied to both Eq. (4) in BP and Eq. (11) in CHEMP. However, the optimal damping factor is difficult to find. The available method relies on the bulky Monte Carlo simulations. In [18], the HAD method is proposed to automatically calculate the damping factor for each BP iteration based on the Kullback-Leibler (KL) divergence between messages in successive iterations. This method shows improved convergence performance compared with BP, but requires online updates of the damping factor at each iteration, which leads to extra computational cost. More details can be found in [18].

B. Max-Sum Algorithm

The max-sum (MS) algorithm is an approximation strategy to eliminate the division operation in Eq. (4) and Eq. (11), which relieves the great difficulty of hardware implementation with some performance loss. Take Eq. (4) as an example, by taking logarithm for both sides of the formula and substitute the resulted summation $\sum_{m=1}^{K} \exp(\alpha_{ij}^{(l)}(s_m))$ with the dominant term $\exp(\max\limits_{s_m \in \Omega}\{\alpha_{ij}^{(l)}(s_m)\})$, we get

$$p_{ij}^{(l)}(x_i = s_k) = \exp(\alpha_{ij}^{(l)}(s_k) - \max_{s_m \in \Omega}\{\alpha_{ij}^{(l)}(s_m)\}). \tag{13}$$

It is clearly seen that the elimination of the division in Eq. (13) reduces the hardware complexity greatly. However, the prior probabilities are overestimated owing to the approximation, which results in performance degradation. To compensate the loss while keeping similar computational complexity, we can apply two modified approaches, the normalized MS (NMS) and the offset MS (OMS) algorithm [19]. Let P_1 and P_2 denote the prior probability values calculated by Eq. (4) and Eq. (13). As discussed above, P_2 will be slightly larger than P_1. NMS aims at multiplying P_2 with a positive scale factor $\lambda < 1$ to get a better approximation, while OMS is dedicated to subtracting an offset factor $\omega < 1$ from P_2. Combining both modifications, the prior probability is computed as follows:

$$\widetilde{P_1} = \lambda \cdot P_2 - \omega, \qquad \lambda < 1, \quad \omega < 1. \tag{14}$$

Combining both message damping and MS with BP, a modified BP algorithm called damped MS BP is summarized in Alg. 1. To accomplish performance enhancement, the values of λ and ω should be carefully selected. An interpolation method to choose the optimal factors is proposed in [19]. Basically, P_1 and P_2 are pre-computed at sampled values of α's, then the corresponding correction factors can be computed to minimize the error of $\widetilde{P_1}$ at each value of α. During the detection iterations, the correction factors are picked from the pre-computed list by nearest-neighbor interpolation of α. In [19], this method shows promising performance with QPSK modulation.

Algorithm 1: Damped MS BP Algorithm

```
Require: $\delta^{(l)}, \lambda^{(l)}, \omega^{(l)}$
Initialize: $p_{ij}^{(0)} \leftarrow 0.5$
Iteration:
  for $l = 1 : number\_of\_iterations$
    for $j = 1 : 2M$
      for $i = 1 : 2N$
        $\beta_{ji}^{(l)}(s_k) = \log \frac{p^{(l-1)}(x_i = s_k | y_j, \mathbf{H})}{p^{(l-1)}(x_i = s_1 | y_j, \mathbf{H})}$
        $\alpha_{ij}^{(l)}(s_k) = \sum_{t=1, t \neq j}^{N} \beta_{ti}^{(l)}(s_k)$
        $p_{ij}^{(l)}(x_i = s_k) = \exp(\alpha_{ij}^{(l)}(s_k) - \max_{s_m \in \Omega}\{\alpha_{ij}^{(l)}(s_m)\})$
        $p_{ij}^{(l)} \leftarrow (1 - \delta^{(l)})(\lambda^{(l)} p_{ij}^{(l)} - \omega^{(l)}) + \delta^{(l)} p_{ij}^{(l-1)}$
    end for
  end for
```

C. Modified CHEMP with Reduced Complexity

Various simplified versions of CHEMP have been proposed to enhance the complexity and performance trade-off, e.g. [21], [39], [44]. In this paper, we propose a novel reduced complexity variant of CHEMP, named simplified MPD (sMPD), by considering the following modifications.

1) Simplified Variance: We first consider an approximation for the variance computation in Eq. (9). Indeed, as shown by Eq. (9), the evaluation of $\mathrm{Var}(x_j)$ involves all $s_k \in \Theta$, which brings a lot of multiplication and addition operations especially when M and K are large. To relieve the implementation challenge, we consider to simplify this formula. Let $J_{\max} = \max_{i,j} J_{ij}$ and $V_{\max} = \max_j \mathrm{Var}(x_j)$, then

$$\begin{aligned}\sigma_i^2 &= \sum_{j=1, j\neq i}^{2M} J_{ij}^2 \mathrm{Var}(x_j) + \sigma_v^2 \leq (2M-1) J_{\max}^2 V_{\max} + \sigma_v^2 \\ &\approx J_{\max}^2 V_i + \sigma_v^2, \end{aligned} \tag{15}$$

where V_i is a scaling factor to estimate the above bound. With the approximation in Eq. (15) along with carefully selected V_i's, σ_i^2's can be pre-computed outside the iterations, which brings computational savings.

2) Enhancements: To enhance the performance of CHEMP in scenarios when channel-hardening is relatively weak, we modify the LLR in Eq. (10) by multiplying each $L_j(s_k)$ with a scaling factor w and adding an offset factor b [39], which leads to a modified LLR computed as

$$\widetilde{L_j}(s_k) \leftarrow \tilde{w} L_j(s_k) + \tilde{b}, \tag{16}$$

in which $\tilde{w}$ and $\tilde{b}$ are scaling and offset factors, respectively. The above re-scaling and shifting aim at compensating for the performance loss due to the Gaussian approximation. Notice that when $\tilde{w} = 1$ and $\tilde{b} = 0$, $\widetilde{L_j}(s_k)$ stays identical with $L_j(s_k)$. Therefore, with optimized $\tilde{w}$ and $\tilde{b}$, the performance is guaranteed to be no worse than the original algorithm.

3) MS estimates: Notice that Eq. (11) in CHEMP shares similar formation as Eq. (4) in BP. Hence, the MS approximation in Eq.s (13) and (14) can also be applied in CHEMP algorithm, which leads to

$$p_j(s_k) \approx \lambda \exp(L_j(s_k) - \max_k\{L_j(s_k)\}) - \omega. \tag{17}$$

To further simplify the iterations and reduce the number of unknown correction factors, we consider a joint expression of Eq.s (16) and (17) as

$$p_j(s_k) = w\exp(L_j(s_k) - \max_k\{L_j(s_k)\}) + b. \tag{18}$$

Combining message damping with the above modifications in Eq.s (15) and (18), the sMPD algorithm is proposed as summarized in Algorithm 2.

Algorithm 2: Damped sMPD Algorithm

```
**Require:** $\mathbf{z}$, $\mathbf{J}$, $\sigma_v^2$, $\delta^{(l)}$, $w^{(l)}$, $b^{(l)}$
1:  **Initialize:** $p_j^{(0)} \leftarrow 0.5, \sigma_j^2 \leftarrow J_{\max}^2 V_j, j = 1, \ldots, 2M$
2:  **Iteration:**
3:    **for** $l = 1 : number\_of\_iterations$
4:      **for** $j = 1 : 2M$
5:        $\mu_j^{(l)} = \sum_{i=1, i\neq j}^{2N} J_{ji}\mathbb{E}(x_i)$
6:        $\widetilde{L}_j^{(l)}(s_k) = (L_j^{(l)}(s_k) - \max_k\{L_j^{(l)}(s_k)\})$
7:        $p_j^{(l)}(s_k) = w^{(l)}\exp(\widetilde{L}_j^{(l)}(s_k)) + b^{(l)}$
8:        $p_j{}^{(l)}(s_k) \leftarrow (1-\delta^{(l)})p_j^{(l)}(s_k) + \delta^{(l)}p_j^{(l-1)}(s_k)$
9:      **end for**
10:   **end for**
```

IV. Proposed DNN MIMO Detector

In this section, we propose a deep neural network (DNN) MIMO detector based on the modified MPDs introduced in Section III. The DNN is constructed by unfolding the iterative algorithms, mapping each iteration as a layer in the network. The correction factors are the parameters to be optimized, and will be "learned" by the DL techniques.

A. Deep Neural Network

DNN, also often called deep feedforward neural network, is one of the quintessential deep learning models. A deep neural network model can be abstracted into a function f that maps the input $\mathbf{x}_0 \in \mathbb{R}^{N_0}$ to the output $\mathbf{y} \in \mathbb{R}^{N_L}$,

$$\mathbf{y} = f(\mathbf{x}_0; \boldsymbol{\theta}), \tag{19}$$

where $\boldsymbol{\theta}$ denotes the parameters that result in the best function approximation of mapping the input data to desirable outputs.

In general, a DNN has a multi-layer structure, composing together many layers of function units (see Fig. 2). Between the input and output layers, there are multiple hidden layers. For an L-layer feed-forward neural network, the mapping function in the l-th layer with input $\mathbf{x}_{l-1}$ from $(l-1)$-th layer and output $\mathbf{x}_l$ propagated to the next layer can be defined as

$$\mathbf{x}_l = f^{(l)}(\mathbf{x}_{l-1}; \boldsymbol{\theta}_l), \tag{20}$$

where $\boldsymbol{\theta}_l$ denotes the parameters of l-th layer, and $f^{(l)}(\mathbf{x}_{l-1}; \boldsymbol{\theta}_l)$ is the mapping function in l-th layer.

According to [29], a DNN can be designed by unfolding the iterative algorithm, mapping each iteration to a layer in the network. This is resulted from the similarities between the message passing FG and DNN. For example, a comparison of BF FG with DNN is summarized in Table I, which confirms that they share similar structures. The MPD is then improved by the DL optimization methods. Hence, a DNN-aided MIMO detector can be developed by unfolding the MPD algorithms including BP and CHEMP, which is introduced in the following section.

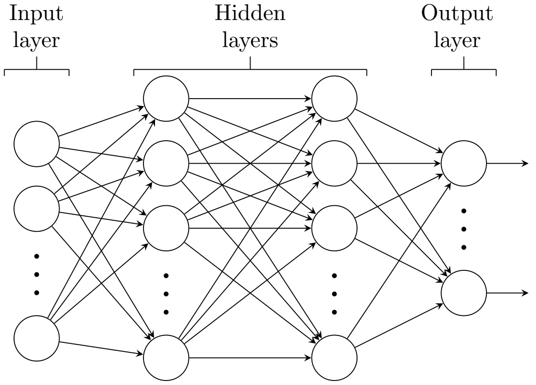


Fig. 2. Architecture of a deep neural network.

TABLE I
BP FG vs. DNN: THE SIMILARITIES

BP FG	**DNN**
Nodes	Neurons
Transmitted signals $\mathbf{x}$	Input data $\mathbf{x}$
Received signals $\mathbf{y}$	Output data $\mathbf{y}$
l-th iteration	l-th hidden layer
Belief messages $\boldsymbol{\alpha}^{(l)}$, $\boldsymbol{\beta}^{(l)}$, $\boldsymbol{p}^{(l)}$	Hidden signals $\mathbf{x}_l$
Message updating rules Eq. (3)-(5)	Mapping function between layers Eq. (24)
Correction factors $\boldsymbol{\delta}$, $\boldsymbol{\lambda}$, $\boldsymbol{\omega}$	Parameters $\boldsymbol{\theta}$

B. Multiscale Correction Factors

The purpose of damping, re-scaling, and offset factors are to correct the iterated messages and compensate the performance loss caused by modifications, hence we call all of them the correction factors. In damped BP, the damping factors can be varied at each iteration. In the selection of normalized/offset factors for MS BP and sMPD, we can further extend those factors to be different for each message $p_{ij}^{(l)}$. Actually, all the correction factors can be set distinct for each message at each iteration, and the calculation of the prior probability can be expressed in a more generalized way.

Specifically, by extending the damping factors, Eq. (12) can be re-written as

$$p_{ij}^{(l)} \Leftarrow (1-\delta_{ij}^{(l)})p_{ij}^{(l)} + \delta_{ij}^{(l)}p_{ij}^{(l-1)}, \tag{21}$$

which forms a multi-scale damped BP. Meanwhile, Line 9 in Alg. 1 can similarly be revised to

$$p_{ij}^{(l)} \Leftarrow (1-\delta_{ij}^{(l)})(\lambda_{ij}^{(l)}p_{ij}^{(l)} - \omega_{ij}^{(l)}) + \delta_{ij}^{(l)}p_{ij}^{(l-1)}, \tag{22}$$

while Line 8 and 9 in Alg. 2 can be reformulated as

$$p_j^{(l)} \Leftarrow (1-\delta_j^{(l)})(w_j^{(l)}\exp(\tilde{L}_j^{(l)}(s_k)) + b_j^{(l)}) + \delta_j^{(l)}p_j^{(l-1)}(s_k), \tag{23}$$

which provide multiple scaled damped MS and damped sMPD approximations.

These multiscale extensions aim at further improvement of the performance. However, they also result in a greater number of parameters to be optimized, especially when the number of antennas are large. This is a complex optimization problem for traditional approaches, but can be handled by the powerful tools in DL.

C. The DNN Detector

As described in Section III, the l-th iteration in the modified MPDs can be summarized by two kinds of messages, including the LLR $\boldsymbol{L}^{(l)}$ and the probability $\boldsymbol{p}^{(l)}$, and the set of correction factors $\boldsymbol{\Delta}^{(l)}$. With $\boldsymbol{L}^{(l-1)}$ and $\boldsymbol{p}^{(l-1)}$ computed from the previous layer $l-1$, we update $\boldsymbol{L}^{(l)}$ and $\boldsymbol{p}^{(l)}$ in order with the help of $\boldsymbol{\Delta}^{(l)}$. This process counts as a full iteration step in MPDs, which can be mapped to a hidden layer in a DNN. In this way, each MPD can be unfolded to construct a DNN MIMO detector, with $\boldsymbol{\Delta}^{(l)}$ as the set of training parameters to be learned.

Specifically, the proposed MPD-based DNN detectors can be generalized by the following formulas:

$$\begin{cases}\{\boldsymbol{L}^{(l)},\boldsymbol{p}^{(l)}\} = f^{(l)}(\boldsymbol{L}^{(l-1)},\boldsymbol{p}^{(l-1)};\boldsymbol{\Delta}^{(l)}),\\ \qquad\qquad\qquad\quad \boldsymbol{O} = \sigma(\boldsymbol{L}^{(L)}),\end{cases} \tag{24}$$

where $f^{(l)}(\boldsymbol{L}^{(l-1)},\boldsymbol{p}^{(l-1)};\boldsymbol{\Delta}^{(l)})$ summarizes the l-th iteration in modified MPDs. O is the output of the DNN while σ denotes a sigmoid or a softmax function which rescale the output $\boldsymbol{L}^{(L)}$ at the final iteration L into range $[0,1]$.

Therefore, with the three modified MPDs discussed in Section III, damped BP, MS BP, and sMPD, three different DNN detectors are proposed as summarized in Table II:

- **DNN-dBP:** For BP, $\boldsymbol{L}^{(l)} = \{\alpha_{ij}^{(l)},\beta_{ij}^{(l)}\}$ while $\boldsymbol{p}^{(l)} = \{p_{ij}^{(l)}\}$. When we derive the DNN based on damped BP, Eq. (21) is used and $\boldsymbol{\Delta} = \{\boldsymbol{\delta}^{(1)},\dots,\boldsymbol{\delta}^{(L)}\}$, where $\boldsymbol{\delta}^{(l)} = \{\delta_{ij}^{(l)}\}$ are the damping factors at each layer. For simplicity, we denote this method as DNN-dBP.
- **DNN-MS:** When the damped MS is applied, $\boldsymbol{p}^{(l)}$'s are computed by Eq. (22). In this case, $\boldsymbol{\Delta} = \{\boldsymbol{\delta}^{(1)},\dots,\boldsymbol{\delta}^{(L)},\boldsymbol{\lambda}^{(1)},\dots,\boldsymbol{\lambda}^{(L)},\boldsymbol{\omega}^{(1)},\dots,\boldsymbol{\omega}^{(L)}\}$, where $\boldsymbol{\delta}^{(l)} = \{\delta_{ij}^{(l)}\}$ are the damping factors, $\boldsymbol{\lambda}^{(l)} = \{\lambda_{ij}^{(l)}\}$ are the normalized factors and $\boldsymbol{\omega}^{(l)} = \{\omega_{ij}^{(l)}\}$ are the offset factors at each iteration. This algorithm is called DNN-MS in the following context.
- **DNN-sMPD:** In sMPD, $\boldsymbol{L}^{(l)} = \{L_j^{(l)}\}$ and $\boldsymbol{p}^{(l)} = \{p_j^{(l)}\}$, while $\boldsymbol{p}^{(l)}$ is calculated with Eq. (23). The training factors $\boldsymbol{\Delta} = \{\boldsymbol{\delta}^{(1)},\dots,\boldsymbol{\delta}^{(L)},\boldsymbol{w}^{(1)},\dots,\boldsymbol{w}^{(L)},\boldsymbol{b}^{(1)},\dots,\boldsymbol{b}^{(L)}\}$, which are the damping, re-scaling, and offset factors accordingly. The proposed DNN detector is name DNN-sMPD.

An example of the structure of the proposed DNN detectors is shown in Fig. 3 with three BP iterations presented. Suppose the MIMO system considered includes M transmitting and N receiving antennas. In general, the input layer has M elements which are initialized with the prior information. For a detector with L BP iterations, the DNN will contain L hidden layers, each layer contains M blue neurons that corresponds to $\boldsymbol{f}$ in Eq. (24), which represents a full iteration in BP of updating the posterior then the prior messages. The choice of $\boldsymbol{f}$ depends on the different modified BP algorithms. Finally, the output layer contains the sigmoid/softmax neurons. To increase the number of iterations in the DNN detector, we only need to concatenate a certain amount of identical hidden layers with blue neurons in Fig. 3 between the input layer and the output layer.

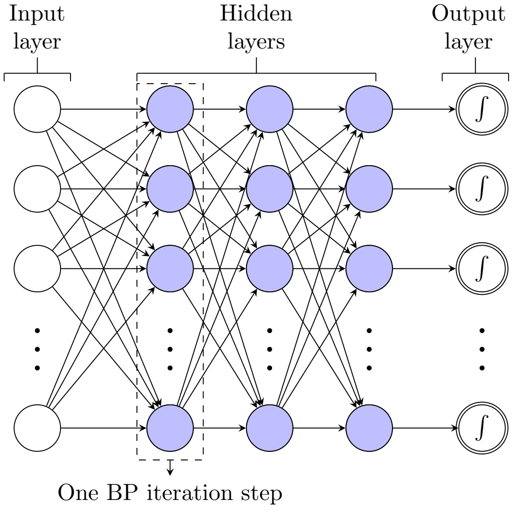


Fig. 3. The structure of the DNN detector with 2 BP iterations.

The cross entropy is adopted to express the expected loss of the neural network output $\boldsymbol{O}$ and the transmitted symbol $\boldsymbol{x}$, which evaluates the performance of the detector as following:

$$L(\boldsymbol{x},\boldsymbol{O}) = -\frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{K} x_i(s_k)\log(O_i(s_k)). \tag{25}$$

The mini-batch stochastic gradient descent (SGD) method is used to minimize the loss function L and decide the optimal correction factors (Δ). With the aid of the advanced DL libraries like Tensorflow [45], these optimizations can be done very efficiently.

V. Numerical Results

For i.i.d. Rayleigh and correlated fading MIMO channels with different antenna configurations, numerical results of the proposed DNN detectors are given. DNN detector based on modified BP, DNN-dBP and DNN-MS, as well as DNN detector based on modified CHEMP, DNN-sMPD, are all considered. The performance of DNN is compared with the original BP algorithm [12], MS BP algorithm [19], and CHEMP [14], with the results of MMSE and the ML detection using sphere decoding (SD) as benchmarks. In the simulations, the MPDs are all mapped to the real domain, and modulation scheme is set as 16-QAM. No channel coding is considered.

A. DNN Architecture and Training Details

To numerically demonstrate the performance of the proposed DNN MIMO detectors, the architecture of the neural

TABLE II
SUMMARY OF THE PROPOSED DNN MIMO DETECTORS: DNN-DBP AND DNN-MS

Method	**DNN-dBP**	**DNN-MS**	**DNN-MPD**
The iterative algorithm	Damped BP [12]	MS BP [19]	Damped sMPD (Alg. 2)
Training parameters $\boldsymbol{\Delta}$	$\boldsymbol{\delta}$	$\boldsymbol{\delta}, \boldsymbol{\lambda}, \boldsymbol{\omega}$	$\boldsymbol{\delta}, \boldsymbol{w}, \boldsymbol{b}$
Inputs	$\boldsymbol{y}, \boldsymbol{\delta}^{(0)}, p_{ij}^{(0)}$	$\boldsymbol{y}, \boldsymbol{\delta}^{(0)}, \boldsymbol{\lambda}^{(0)}, \boldsymbol{\omega}^{(0)}, p_{ij}^{(0)}$	$\boldsymbol{z}, \boldsymbol{J}, V_j, \boldsymbol{\delta}^{(0)}, \boldsymbol{w}^{(0)}, \boldsymbol{b}^{(0)}, p_j^{(0)}$
Mapping functions $\boldsymbol{f}^{(l)}$ at the l-th iteration	$\beta_{ji}^{(l)}(s_k) = \log \frac{p^{(l-1)}(x_i=s_k \mid y_j, \mathbf{H})}{p^{(l-1)}(x_i=s_1 \mid y_j, \mathbf{H})}$ $\alpha_{ij}^{(l)}(s_k) = \sum_{t=1, t\neq j}^{N} \beta_{ti}^{(l)}(s_k)$ $p_{ij}^{(l)}(s_k) = \frac{\exp(\alpha_{ij}^{(l)}(s_k))}{\sum_{m=1}^{K} \exp(\alpha_{ij}^{(l)}(s_m))}$ $p_{ij}^{(l)} \leftarrow (1-\delta_{ij}^{(l)}) p_{ij}^{(l)} + \delta_{ij}^{(l)} p_{ij}^{(l-1)}$	$\beta_{ji}^{(l)}(s_k) = \log \frac{p^{(l-1)}(x_i=s_k \mid y_j, \mathbf{H})}{p^{(l-1)}(x_i=s_1 \mid y_j, \mathbf{H})}$ $\alpha_{ij}^{(l)}(s_k) = \sum_{t=1, t\neq j}^{N} \beta_{ti}^{(l)}(s_k)$ $p_{ij}^{(l)}(s_k) = \exp(\alpha_{ij}^{(l)}(s_k) - \max_{s_m \in \Omega} \{\alpha_{ij}^{(l)}(s_m)\})$ $p_{ij}^{(l)} \leftarrow (1-\delta_{ij}^{(l)}) \lambda_{ij}^{(l)} p_{ij}^{(l)} - \omega_{ij}^{(l)} + \delta_{ij}^{(l)} p_{ij}^{(l-1)}$	$\mu_j^{(l)} = \sum_{i=1, i\neq j}^{2M} J_{ji} \mathbb{E}(x_i)$ $L_j^{(l)}(s_k) = \frac{J_{jj}(s_k - s_1)(2(z_j - \mu_j^{(l)}) - J_{jj}(s_k + s_1))}{2\sigma_j^{2(l)}}$ $\widetilde{L_j}^{(l)}(s_k) = (L_j^{(l)}(s_k) - \max_k \{L_j^{(l)}(s_k)\})$ $p_j^{(l)}(s_k) = \exp(w_{jk}^{(l)} \widetilde{L_j}^{(l)}(s_k) + b_{jk}^{(l)})$ $p_j^{(l)} \leftarrow (1-\delta_j^{(l)}) p_j^{(l)} + \delta_j^{(l)} p_j^{(l-1)}$
Loss function	$L(\boldsymbol{x}, \boldsymbol{O}) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} x_i(s_k) \log(O_i(s_k))$		

network should be carefully selected. The settings of DNN-dBP, DNN-MS, and DNN-sMPD in our simulations are summarized in Table III, and some details of these settings are discussed in this section.

1) Configurations and Neurons: As described in Section IV-C, the number of neurons are decided simply according to the number of transmitting antennas M. Define $\rho = M/N$ as the system loading factor. For antenna configurations, we fix $M = 8$ and consider systems with $N = 32$, 64 and 128 (or $\rho = 0.25$, 0.125, and 0.0625) in the numerical tests.

2) The depth of DNN: The depth of the DNN relates to the number of MPD iterations, which is another vital factor for implementation. As mentioned in Section IV-C, if the number of iterations is L, the depth of the network will also be L. To properly select L, it's important to keep a good balance between the BER performance and the complexity. In our case, L is decided with a greedy search method as follows: (i) A searching range of possible values of L, $[l_{min}, l_{max}]$, is decided by the BER performance of the original BP. (ii) Starting with the smallest value $L = l_{min}$, we train the DNN detectors and test the trained network to obtain the BER performance, till it plateaus. (iii) For simplicity, this process is done once for $M = 8, N = 32$ for each DNN detectors in i.i.d channels. The obtained L for each detector is summarized in Table III.

3) Training Parameters: The correction factors in the MPDs presented in Table II are all extended to be different for each message. However, for the relative DNN detectors, the number of training parameters will be large in massive MIMO systems, which brings overwhelming pressure for the training process. Hence, simulations are carried out to select the most effective parameters which are sufficient to compensate for the performance. We first consider the damping factors $\boldsymbol{\delta}$ in both DNN-dBP and DNN-MS, for which trainings are done to compare the performance with optimized $\{\delta^{(l)}\}$ (different factors for each iteration) and $\{\delta_{ij}^{(l)}\}$ (different factors for each message). It can be inferred from Fig. 4 that $\{\delta^{(l)}\}$ is sufficient to enhance the performance of both algorithms and achieves superior BER results. Therefore, the damping factors for all the DNN detectors are reduced to $\{\delta^{(l)}\}$. Similarly, from Fig. 5 we can observe that for DNN-MS, multi-scaled scaling factors are sufficient to optimize the BER performance and the offset factors can be skipped. However, for DNN-sMPD, both factors are required to attain the advanced performance. Overall, the selected formation of training parameters for each DNN detector is summarized in Table III.

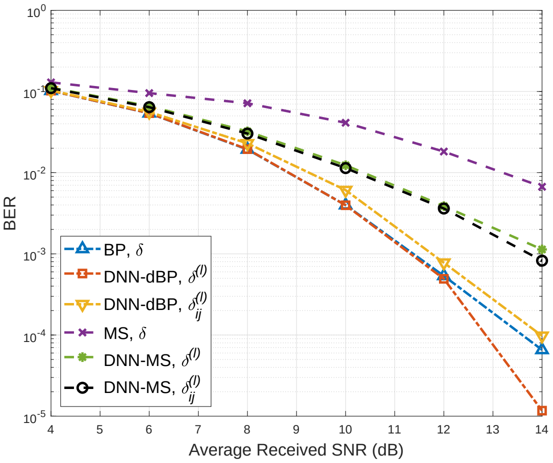


Fig. 4. BER performance of DNN-dBP and DNN-MS with different settings of δ when $M = 8$, $N = 32$. Here, δ denotes fixed damping factor $\delta = 0.5$, $\delta^{(l)}$ means that damping factor varies for each layer, while $\delta_{ij}^{(l)}$ means different factors for each message. $\delta^{(l)}$'s and $\delta_{ij}^{(l)}$'s are all optimized via training. Notice that these results are only used to decide the damping factor, hence other factors are not considered here for DNN-MS.

4) Training details: The DNN is implemented on the advanced DL framework Tensorflow [45]. We train the network using a variant of the SGD method for optimizing deep networks, named Adam Optimizer [46]. The signal-to-noise ratios (SNRs) are ranging from 4dB to 16dB (every 2dB). We use batch training with 100 random data samples (20 for each SNR step) at each iteration. For DNN-dBP and DNN-sMPD, the network is trained for 5000 iterations, the DNN-MS case is trained for 10000 iterations. Notice that only one offline training is performed for each simulated antenna configuration for DNN-dBP and DNN-MS, while only one training is done for DNN-sMPD since all simulated cases share the same M. All the simulation results in different channel conditions are calculated with the same trained network. The training

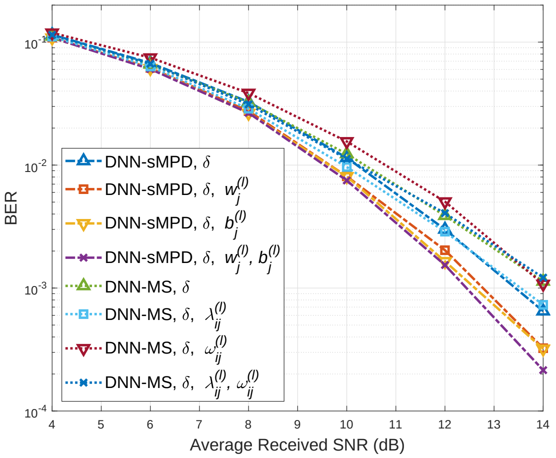


Fig. 5. BER performance of DNN-sMPD and DNN-MS with different settings of scaling and offset factors when $M = 8, N = 32$. Here, a fixed damping factor $\delta = 0.5$ is applied for all cases. DNN-sMPD and DNN-MS are trained with scaling factors $(w_j^{(l)}, \lambda_{ij}^{(l)})$ only, offset factors $(b_j^{(l)}, \omega_{ij}^{(l)})$ only, and both, to decide the training parameters.

TABLE III
THE SETTINGS OF DNN DETECTORS IN THE NUMERICAL TESTS.

Method	DNN-dBP	DNN-MS	DNN-sMPD
Training Parameters	$\delta^{(l)}$	$\delta^{(l)}, \lambda_{ij}^{(l)}$	$\delta^{(l)}, w_j^{(l)}, b_j^{(l)}$
SNRs for training	{4, 6, 8, 10, 12, 14, 16} dB		
Mini-batch size	100		
Size of training data	100000	1000000	500000
Optimization method	Adam optimizer		

parameters are all initialized as 0.5.

B. Numerical Results

1) Antenna Configurations: We first present the performance of the proposed DNN MIMO detectors under different antenna configurations in i.i.d. Rayleigh fading channels. In Fig. 6, we fix $M = 8$ and consider $N = 32$, 64, and 128 ($\rho = 0.25$, 0.125, and 0.0625), respectively. The BER performance of the proposed DNN-dBP, DNN-MS, DNN-sMPD is compared with various MPDs including CHEMP [14], sMPD (Alg. 2), DNN-MPD [39], and MMSE. Here, the damping factor of CHEMP and sMPD are both optimized through simulation trials, and the correction factors in sMPD are set as $w = 1$ and $b = 0$, which means no compensation is applied. The optimal SD [47] result is shown as the benchmark. L denotes the number of iterations, or equivalently the number of hidden layers in the DNN detectors.

In Fig. 6a, we observe that DNN-dBP achieves the best performance among the presented MPDs, attaining a leading gap of 2 dB at the BER 10^{-5} compared to MMSE. DNN-sMPD shows similar performance as MMSE, which is 1 dB behind DNN-MPD at BER 10^{-5} but with reduced complexity. However, DNN-sMPD outperforms both sMPD and CHEMP, which confirms the performance enhancement achieved via training. When N is raised to 64, as demonstrated in Fig. 6b, the performance gap among the MPDs are much smaller. DNN-dBP still attains the best results, outperforming CHEMP by only 0.5 dB at the BER of 10^{-5}. DNN-sMPD and DNN-MPD both show similar performance as CHEMP, which are slightly ahead of MMSE. In a larger-scaled system when $N = 128$, it can be seen from Fig. 6c that the performances of the detectors are similar with the $N = 64$ case. DNN-dBP still dominates the results, while DNN-sMPD and DNN-MS show similar performance as MMSE. However, these performances are achieved with a much smaller number of iterations $L = 3$.

2) Channel Correlation: In Fig. 7, the performance of DNN-dBP and DNN-MS are compared in correlated channels. The number of antennas is set as $M = 8$ and $N = 32$ while different settings of correlations are considered, including correlations only at the transmitting side (Tx-Cor.), the receiving side (Rx-Cor.), and both sides (Rx-Tx-Cor.). The correlation factor [41] is set as 0.3 for all the scenarios. The performance of DNN detectors are compared with the original BP [15], MS BP [19], and MMSE. Here, the damping factors of BP and MS BP are selected by simulations, compensation schemes like OMS and NMS are not considered for MS BP. Also, CHEMP-based algorithms are not presented here due to their degraded performance in correlated channels.

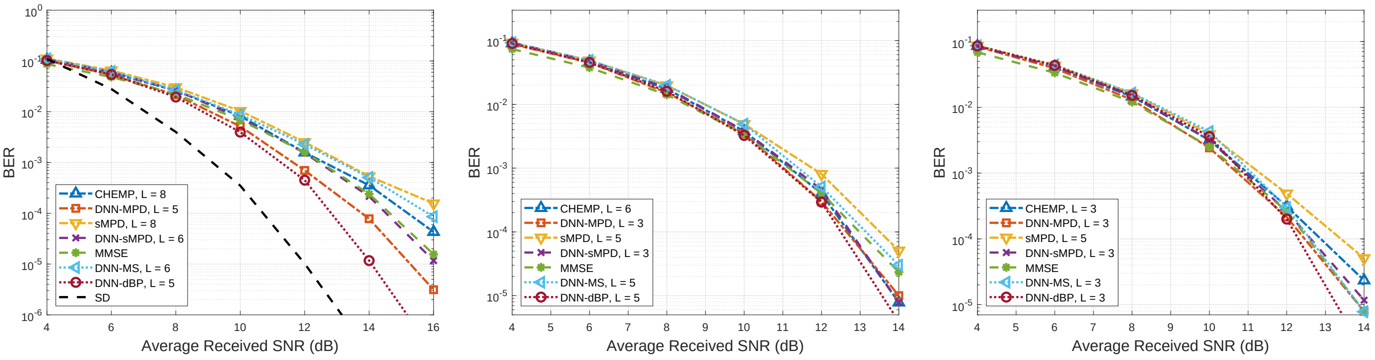
As shown in Fig. 7a, DNN-dBP shows similar performance as BP in the i.i.d. channels, with just a slight improvement. MS BP suffers large degradation from BP, while DNN-MS results achieve some improvements, however, are still far from satisfying. When channel correlations are considered, the performances of DNN detectors attain much more noticeable advantages. In Fig. 7b, when Rx-Cor. or Tx-Cor. is considered, DNN-dBP again shows the best performance. For instance, it outperforms BP and MMSE with 1 dB at BER of 10^{-4} when Tx-Cor. is considered. Similar to the i.i.d. cases, DNN-MS curves show great improvements compared with MS BP, but still have a small degradation from BP results. When the correlation is considered at both transmitting and receiving sides, we can observe from Fig. 7c that both DNN-dBP and DNN-MS can outperform BP as well as MMSE, which are about 1 dB ahead when BER= 10^{-3}.

From the above numerical results, the performance of the proposed DNN detectors can be summarized as follows:

- DNN-dBP can achieve the best performance and outperforms all the discussed MPDs in both Fig.s 6 and 7. However, its advantages are negligible in i.i.d. channels since the BER results are similar to the original BP.
- DNN-MS achieves much better performance compared to MS BP, and it attains the best results in spatially correlated channels.
- DNN-sMPD obtains similar performance as MMSE at different antenna configurations, which outperforms CHEMP with few iterations.

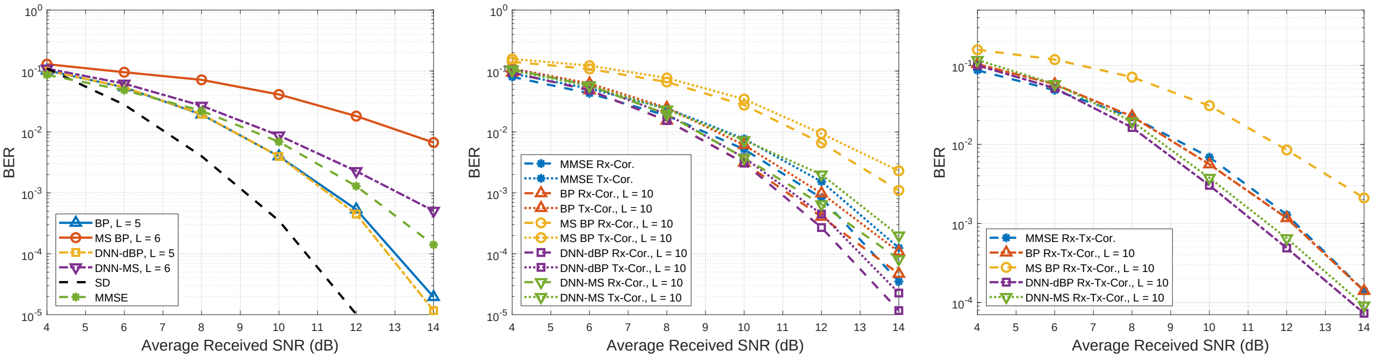
C. Complexity Analysis

1) Offline Training: In our numerical tests, we train the DNN once for each antenna configuration for DNN-dBP and DNN-MS. In the case of DNN-sMPD, only one training is carried out for all online simulations. The training requires a large amount of data according to Table III. The specific amount of these inputs depends on the number of training parameters. Generally, a DNN with a larger number of trainable



(a) $M = 8,\ N = 32,$ (b) $M = 8,\ N = 64$ (c) $M = 8,\ N = 128$

Fig. 6. Performance comparison of SD, MMSE, CHEMP [14], sMPD (Alg. 2, DNN-MPD [39], DNN-sMPD, DNN-dBP, and DNN-MS with various antenna configurations. L denotes the number of iterations in the MPDs.



(a) $M = 8,\ N = 32$, i.i.d. Rayleigh (b) $M = 8,\ N = 32$, Tx-Cor. and Rx-Cor. (c) $M = 8,\ N = 32$, Rx-Tx-Cor.

Fig. 7. Performance comparison of SD, MMSE, BP [15], MS BP [19], DNN-dBP, and DNN-MS in various correlated channels, including correlations only at the transmitting side (Tx-Cor.), the receiving side (Rx-Cor.), and both sides (Rx-Tx-Cor.). The correlation factors are all set to 0.3, and L denotes the number of iterations in the MPDs.

parameters requires higher training complexity. As shown in Table IV, DNN-dBP only has L damping factors to be trained. Meanwhile, DNN-MS requires $L+MNL$ parameters including the damping and the scaling factors, while $L+2ML$ parameters are needed for DNN-sMPD. This is why the number of training data shown in Table III for DNN-MS is much larger than the other two detectors. On the other hand, the training complexity also scales with the complexity of the original MPD algorithms since simulations are required to complete the training process. These complexities are compared in the following section. Notice that the training in this paper is all done offline, and the complexity can be handled by powerful computational and storage devices. The trained network can be stored for multiple online uses. Another inevitable issue of the DNN is that the ”optimized” network depends on the range of the training data. In practical problems, the training data should be generated with certain scenarios that we focus on to reach optimal performance.

2) Online Detection: The computational complexity of the proposed DNN detectors are compared with the other MPD algorithms in Table IV. The BP we consider are based on the real domain single-edged BP detector proposed in [15], which achieves a complexity of order $O(MNK)$ at each iteration. MS BP removes M division operations from every BP iteration. The online parts of DNN-dBP and DNN-MS share the same order of complexity as BP and MS BP, respectively, while DNN-MS requires MNL extra multiplications for the scaling factors. On the other hand, CHEMP [14] and DNN-sMPD both requires a preprocessing step before iterations to compute $\mathbf{J}$ and $\mathbf{z}$, which has a cost of $O(M^2N)$. DNN-sMPD requires an extra step to evaluate σ_j^2 according to Eq. (15) before iterations start, which requires M comparisons, M additions and M multiplications. In each iteration, the complexity of CHEMP is of order $O(M^2K)$. DNN-sMPD has the same order of complexity, but it reduces M^2K multiplications and $M(M-1)K$ additions for computing σ_j^2's. Meanwhile, one division is removed by Eq. (18), while M multiplications and M additions are required by DNN-sMPD for the correction factors. Hence, the proposed DNN-dBP achieves improved BER performance with the same computation complexity as the original BP. DNN-MS detection reduces the complexity by eliminating divisions that are difficult to implement, and it outperforms the MS algorithms significantly without extra computational cost. DNN-sMPD also outperforms CHEMP with reduced complexity and elimination of divisions. A detailed complexity comparison among the MPDs for the simulations in Fig. 6 is demonstrated in Fig. 8.

The recently proposed DNN based MIMO detector, DetNet[34], shows advantages in the sense that the knowledge of the channel noise variance or SNR level is not required. It is

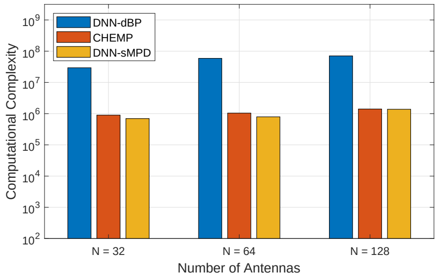


Fig. 8. Complexity comparison of the MPDs for simulations in Fig. 6. BP, MS BP, and DNN-MS share similar complexity as DNN-dBP, hence only DNN-dBP is listed in this figure. The complexity here is calculated with distinct costs for different operations according to [48].

TABLE IV
COMPLEXITY COMPARISON OF THE MPDS

Method	#TrainParam.[a]	Preproc.[b]	Iterations[c]
BP [15]	—	—	$O(MNKL)$
MS BP [19]	—	—	$O(MNKL)$
CHEMP [14]	—	$O(M^2N)$	$O(M^2KL)$
DNN-dBP	L	—	$O(MNKL)$
DNN-MS	$L+MNL$	—	$O(MNKL)$
DNN-sMPD	$L+2ML$	$O(M^2N)$	$O(M^2KL)$

[a] Number of training parameters.
[b] The complexity of computing $\mathbf{J}$ and $\mathbf{z}$.
[c] L denotes the number of iterations and K denotes the modulation constellation size.

based on unfolding a projected gradient descent like algorithm for ML optimization, which is not our focus and hence is fundamentally different from our work which requires channel noise variance knowledge. It achieves great performance at a similar level of complexity for online detection of $O(MNL)$. However, a large number of hidden layers of DNN is needed to get satisfactory results, which also adds to the burden of the offline training cost.

VI. Hardware Implementation

In this section, the hardware architectures for the proposed DNN-sMPD are presented, which are responsible for the online detection. The DNN-sMPD only needs to be trained offline once. The obtained weights can be applied to the online detection.

A. Overview of Hardware Architectures

The DNN-sMPD detector for an $M \times N$ massive MIMO uplink system is divided to two parts, namely interference cancellation processing elements (IPEs) and constellation matching processing elements (CPEs), similar to [22]. Precisely, $2M$ IPEs and $2M$ CPEs are implemented in the DNN-sMPD detector to iteratively estimate the M user symbols in real domain. Each IPE computes the Gaussian approximation of the interference from the other $M-1$ user as shown in Alg. 2 Line 5. Each CPE computes the expectation of the symbol based on the K different constellation points. As shown in Fig. 9, each IPE transmits the Gaussian approximation to the corresponding CPE while each CPE connects to all IPEs except the corresponding IPE to propagate the expectation of its symbol.

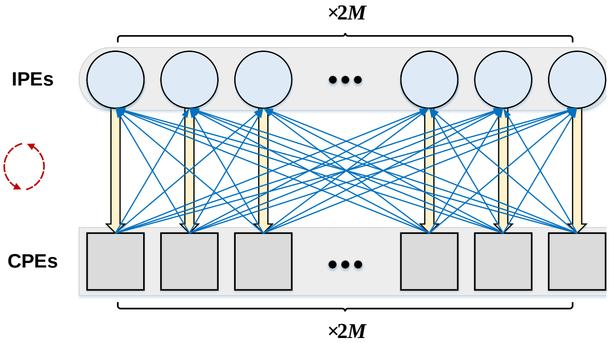


Fig. 9. Full-parallel architecture for DNN-sMPD.

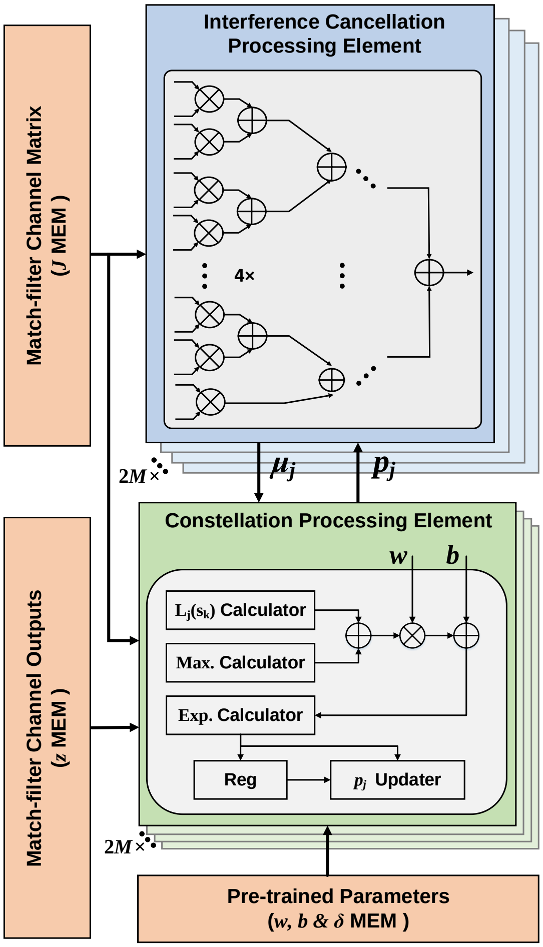


Fig. 10. Detailed block diagram for DNN-sMPD with $M \times N = 8 \times 128$.

In Fig. 10, the pre-calculated match-filter channel matrix $\mathbf{z} = \mathbf{H}^T\mathbf{y}/N$ and match-filter channel outputs $\mathbf{J} = \mathbf{H}^T\mathbf{H}/N$ are stored in memory. The pre-trained paremeters w, b, δ in Alg. 2 are stored in the parameter memory.

B. Detailed Structure of IPE

Fig. 10 illustrates the overall block diagram of propose DNN-sMPD detector. Each IPE for 8×128 MIMO system consists of 15 multipliers and 14 adders in real domain. Each multiplier computes the product of the expectation of the symbol and the corresponding element in matrix **J**. An adder tree composed of 14 adders is adopted for the addition of the results of 15 multipliers to compute the Gaussian approximation of the interference.

C. Detailed Structure of CPE

For a 8×128 MIMO system using 16-QAM modulation mode, each CPE is composed of 4 $L_j(s_k)$ calculators, a maximum calculator, 4 exponent calculators, 4 registers, and 4 p_j updaters. Specifically, 4 $L_j(s_k)$ calculators compute the LLR vector of symbol x_j as is shown in line 6 in Alg. 2. Each $L_j(s_k)$ calculator uses 2 adders, 4 multipliers and 1 shift operation since $s_k - s_1$, $s_k + s_1$ and $\frac{1}{2\sigma^2}$ can be calculated in advance. The maximum calculator of 4 numbers utilizes 3 comparators of two inputs with a layered comparative structure. The maxinum value of 4 $L_j(s_k)$ is used to compute the modified LLR vector of symbol x_j by being subtracted $L_j(s_k)$, multiplying with w and adding an offset factor b. The exponent calculator in the CPE is implemented using look-up table (LUT). The output of the exponent calculator of the previous iteration is stored in the register. The probability of s_k is calculated in the p_j updater by the weighted sum of the output of the exponent calculator in this iteration and the value of the register.

D. ASIC Results

The proposed 8×128 DNN-sMPD is implemented using 65 nm CMOS technology and synthesized by Synopsys Design Compiler. Assuming the clock frequency is f_{clk}, the average detecting cycles are T_{cycle} and the modulation constellation size is Q, the throughput of detector is defined as:

$$\text{Throughput[Gb/s]} = \frac{\log_2(Q) \times M}{T_{cycle}} \times f_{clk}. \quad (26)$$

We compare our design with five SOA MIMO detectors, including two CHEMP-based MPD [21], [22], two MMSE-based detectors [49], [50], and one expectation propagation detector [51]. To ensure a fair comparison, the clock frequency and area for different technologies are normalized to 65nm as

$$f_{clk} \propto s, \;\; A \propto \frac{1}{s^2} \quad (27)$$

where f_{clk}, s, A denote the clock frequency, the technology rate, and area, respectively[49]. The number of receiving attennas M is scaled down to 8 by assuming that the area and critical-path delay decrease by factors of $8/M$ and $\log_2 8/\log_2 M$ [21], respectively. For the detector adopting MPD or CHEMP algorithm, the area is assumed to be scaled by the computational complexity as [21]. The area of detector adopting MMSE or MPD algorithm maintains unchanged with the change of modulation order because the hardware implementations of these two algorithm have little to do with modulation order.

Table V summarizes the comparison results. The modulation for all designs is normalized to 16-QAM. Compared to MMSE-based detectors [49], [50], the proposed detector achieves about 3× throughput and much higher area efficiency than [50] while the area efficiency is comparative with [49]. The advantage of our detector is BER? .... The EPD [51] can yield near-optimal BER performance. However, it requires the exact matrix inversion with high computational complexity of $O(M^3)$. Hence, the resulting area efficiency is about 43.5% of our design.

The CHEMP detector in [21] attains higher throughput than the proposed detector. , buzhidao zen me xie .... However, it's worth pointing out that the design in [22] can only produce hard-output. But soft-output is adopted in the proposed DNN detector to enhance the performance when integrated with decoders, which usually has higher complexity. Also, the modified CHEMP algorithm proposed in [21] is hardware implementation-oriented with various simplifications and approximations especially for QPSK modulation. Compensation factors selected by Monte Carlo-like simulations are used in [21] to enhance the performance similar to the proposed sMPD. Therefore, the BER performance of [21] is similar to the original CHEMP, and the good performance is limited to systems with small loading factors. However, as shown in the above simulation results, DNN-sMPD can outperform CHEMP with various antenna configurations, which also achieves enhanced robustness. Furthermore, the proposed framework to design DNN detectors based on MPDs in this paper can indeed be applied to the algorithm in [21] to optimize the factors and enhance the BER performance.

VII. Conclusion

In this paper, we present a novel framework to design DNN massive MIMO detectors. Three DNN detectors including DNN-dBP, DNN-MS, and DNN-sMPD are then proposed by unfolding modified BP and CHEMP algorithms. The architecture of the DNN detectors and the training strategies are discussed. Numerical results with different antenna configurations and various channel conditions are illustrated to show the advanced performance of the proposed methods. An efficient hardware architecture for DNN-sMPD is also presented with implementation results. The future work will be directed towards further optimization of the DNN structure and efficient training methods. Also, this framework can be applied to improve various other MPDs.

References

[1] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, 2010.

[2] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, 2013.

[3] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[4] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine Learning Paradigms for Next-Generation Wireless Networks," *IEEE Commun. Mag.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.

[5] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-Driven Deep Learning for Physical Layer Communications," *arXiv:1809.06059 [cs, math]*, Sep. 2018.

[6] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang, "Deep Learning in Physical Layer Communications," *arXiv:1807.11713 [cs, math]*, Jul. 2018.

TABLE V
COMPARISON OF IMPLEMENTATION RESULTS ON ASIC

Detector	TCAS-I'18 [49]	VLSI'16 [22][a]	TCAS-I'19 [21][a]	ISSCC'17 [50]	ISSCC'18 [51]	This Work[a]
Algorithm	MMSE	Hard-output CHEMP	Soft-output MPD[b]	ZF/MMSE	EPD	Soft-output sMPD
$M \times N$	8×128	32×128	8×128	8×128	16×128	8×128
Modulation	64-QAM	256-QAM	QPSK	256-QAM	256-QAM	16-QAM
Soft Output						Yes
Technology [nm]	65	40	40	28	28	65
Area [mm^2]	2.57	0.58	0.076	1.1	2.0	0.80
Frequency [MHz]	680	425	500	300	569	340
Throughput [Gb/s]	1.02	2.76	8.0	0.3	1.8	0.18
Norm. T/P [Gb/s][c]	0.68[d]	1.42[d]	9.84[d]	0.06[d]	0.52[d]	0.18
Norm. Area Eff. [Gb/s/mm^2][c]	0.26[d]	11.09[d]	16.35[d]	0.01[d]	0.10[d]	0.23

[a] Preprocessing unit for computing $\mathbf{J}$ and $\mathbf{z}$ is not included.

[b] A modified CHEMP is used in [21], in which various modifications are proposed to reduce the complexity for QPSK modulation.

[c] Normilized to 65nm based on: frequency $\propto s$, area $\propto \frac{1}{s^2}$, where s is the technology factor.

[d] Scaled to $M = 8$ and 16-QAM.

[7] X. Yuan, L. Ping, C. Xu, and A. Kavcic, "Achievable rates of MIMO systems with linear precoding and iterative LMMSE detection," *IEEE Trans. Inf. Theory*, vol. 60, no. 11, pp. 7073–7089, 2014.

[8] A. K. Sah and A. Chaturvedi, "An MMP-based approach for detection in large MIMO systems using sphere decoding," *IEEE Wireless Commun.*, vol. 6, no. 2, pp. 158–161, 2017.

[9] P. Li and R. D. Murch, "Multiple output selection-LAS algorithm in large MIMO systems," *IEEE Commun. Lett.*, vol. 14, no. 5, 2010.

[10] N. Srinidhi, S. K. Mohammed, A. Chockalingam, and B. S. Rajan, "Low-complexity near-ML decoding of large non-orthogonal STBCs using reactive tabu search," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2009, pp. 1993–1997.

[11] Z. Wu, C. Zhang, Y. Xue, S. Xu, and X. You, "Efficient architecture for soft-output massive MIMO detection with Gauss-Seidel method," in *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2016, pp. 1886–1889.

[12] J. Yang, C. Zhang, X. Liang, S. Xu, and X. You, "Improved symbol-based belief propagation detection for large-scale MIMO," in *Proc. of IEEE Workshop on Signal Processing Systems (SiPS)*, 2015, pp. 1–6.

[13] C. Jeon, R. Ghods, A. Maleki, and C. Studer, "Optimality of large MIMO detection via approximate message passing," in *2015 IEEE International Symposium on Information Theory (ISIT)*. Hong Kong, Hong Kong: IEEE, Jun. 2015, pp. 1227–1231.

[14] T. L. Narasimhan and A. Chockalingam, "Channel Hardening-Exploiting Message Passing (CHEMP) Receiver in Large-Scale MIMO Systems," *arXiv:1310.3062 [cs, math]*, Oct. 2013, cites: narasimhanChannelHardeningExploitingMessage2013.

[15] J. Yang, W. Song, S. Zhang, X. You, and C. Zhang, "Low-complexity belief propagation detection for correlated large-scale MIMO systems," *Journal of Signal Processing Systems*, pp. 1–15, 2017.

[16] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proc. of the 15th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

[17] Q. Su and Y.-C. Wu, "On convergence conditions of gaussian belief propagation." *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1144–1155, 2015.

[18] Y. Gao, H. Niu, and T. Kaiser, "Massive MIMO detection based on belief propagation in spatially correlated channels," in *Proc. of 11th International ITG Conference on Systems, Communications and Coding (SCC)*, 2017, pp. 1–6.

[19] Y. Zhang, L. Ge, X. You, and C. Zhang, "Belief propagation detection based on max-sum algorithm for massive MIMO systems," in *Proc. of 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, 10 2017, pp. 1–6.

[20] Y. Chen, C. Cheng, T. Tsai, W. Sun, Y. Ueng, and C. Yang, "A 501mW 7.6lGb/s integrated message-passing detector and decoder for polar-coded massive MIMO systems," in *Proc. of IEEE Symposium on VLSI Circuits*, Jun. 2017, pp. C330–C331.

[21] Y. Chen, W. Sun, C. Cheng, T. Tsai, Y. Ueng, and C. Yang, "An Integrated Message-Passing Detector and Decoder for Polar-Coded Massive MU-MIMO Systems," *IEEE Trans. Circuits Syst. I*, vol. 66, no. 3, pp. 1205–1218, Mar. 2019.

[22] W. Tang, C. Chen, and Z. Zhang, "A 0.58mm2 2.76Gb/s 79.8pJ/b 256-QAM massive MIMO message-passing detector," in *Proc. of IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, Jun. 2016, pp. 1–2.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[24] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[25] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[26] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. of 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016, pp. 341–346.

[27] W. Xu, Z. Wu, Y.-L. Ueng, X. You, and C. Zhang, "Improved polar decoder based on deep learning," in *Proc. of IEEE Workshop on Signal Processing Systems (SiPS)*, 2017, pp. 1–6.

[28] L. Lugosch and W. J. Gross, "Neural offset min-sum decoding," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2017, pp. 1361–1365.

[29] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 399–406.

[30] M. Borgerding and P. Schniter, "Onsager-corrected deep learning for sparse linear inverse problems," in *Proc. of IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016, pp. 227–231.

[31] A. Mousavi and R. G. Baraniuk, "Learning to invert: Signal recovery via deep convolutional networks," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2272–2276.

[32] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *Proc. of IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2016, pp. 223–228.

[33] H. Ye, G. Y. Li, and B.-H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[34] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," *arXiv preprint arXiv:1706.01151*, 2017.

[35] C. Jin, Y. Zhang, S. Yu, R. Hu, and C. Chen, "Virtual MIMO blind detection clustered wsn system," in *Proc. of Asia-Pacific Microwave Conference (APMC)*, vol. 3, 2015, pp. 1–3.

[36] 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. PP, no. 99, pp. 1–15, 2017.

[37] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, “Signal detection of MIMO-OFDM system based on auto encoder and extreme learning machine,” in *Proc. of International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1602–1606.

[38] J. Guo, B. Song, Y. Chi, L. Jayasinghe, C. Yuen, Y. L. Guan, X. Du, and M. Guizani, “Deep neural network-aided Gaussian message passing detection for ultra-reliable low-latency communications,” *Future Generation Computer Systems*, vol. 95, pp. 629–638, Jun. 2019.

[39] X. Tan, Z. Zhang, X. You, and C. Zhang, “Low-Complexity Message Passing MIMO Detection Algorithm with Deep Neural Network,” in *Proc. of IEEE GlobalSiP*, Anaheim, CA, USA, Nov. 2018, p. 5.

[40] H. He, C.-K. Wen, S. Jin, and G. Y. Li, “A model-driven deep learning network for MIMO detection,” in *IEEE GlobalSiP*, Anaheim, CA, USA, Nov. 2018.

[41] J. Proakis, *Digital Communications*, ser. Electrical engineering series. McGraw-Hill, 2001.

[42] W. Fukuda, T. Abiko, T. Nishimura, T. Ohgane, Y. Ogawa, Y. Ohwatari, and Y. Kishiyama, “Low-complexity detection based on belief propagation in a massive MIMO system,” in *Proc. of IEEE 77th Vehicular Technology Conference (VTC Spring)*, 2013, pp. 1–5.

[43] A. Chockalingam and B. S. Rajan, *Large MIMO Systems*. New York, NY, USA: Cambridge University Press, 2014.

[44] H. Zhu, J. Lin, and Z. Wang, “Reduced complexity message passing detection algorithm in large-scale MIMO systems,” in *Proc. of 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, Oct. 2017, pp. 1–5.

[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.

[46] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[47] C. P. Schnorr and M. Euchner, “Lattice basis reduction: Improved practical algorithms and solving subset sum problems,” *Math. Programming*, vol. 66, no. 1, pp. 181–199, Aug. 1994.

[48] R. P. Brent and P. Zimmermann, *Modern Computer Arithmetic*. Cambridge University Press, Nov. 2010.

[49] G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, “A 1.58 Gbps/W 0.40 Gbps/mm2 ASIC Implementation of MMSE Detection for $128\times 8 64$ -QAM Massive MIMO in 65 nm CMOS,” *IEEE Trans. Circuits Syst. I*, vol. 65, no. 5, pp. 1717–1730, May 2018.

[50] H. Prabhu, J. N. Rodrigues, L. Liu, and O. Edfors, “A 60pJ/b 300Mb/s 128× 8 massive mimo precoder-detector in 28nm FD-SOI,” in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2017, pp. 60–61.

[51] W. Tang, H. Prabhu, L. Liu, V. Öwall, and Z. Zhang, “A 1.8 Gb/s 70.6 pJ/b 128× 16 link-adaptive near-optimal massive MIMO detector in 28nm UTBB-FDSOI,” in *IEEE International Solid-State Circuits Conference (ISSCC)*, 2018, pp. 224–226.