Low Complexity Joint Impairment Mitigation of I/Q Modulator and PA Using Neural Networks

Citation for the original published paper (version of record):

Wu, Y., Gustavsson, U., Graell I Amat, A. et al (2022). Low Complexity Joint Impairment Mitigation of I/Q Modulator and PA Using Neural Networks. IEEE Journal on Selected Areas in Communications, 40(1): 54-64. http://dx.doi.org/10.1109/JSAC.2021.3126024

Low Complexity Joint Impairment Mitigation of I/Q Modulator and PA Using Neural Networks

Yibo Wu, *Student Member, IEEE*, Ulf Gustavsson,
Alexandre Graell i Amat, *Senior Member, IEEE*, and Henk Wymeersch, *Senior Member, IEEE*

***Abstract*—Neural networks (NNs) for multiple hardware impairments mitigation of a realistic direct conversion transmitter are impractical due to high computational complexity. We propose two methods to reduce the complexity without significant performance penalty. First, propose a novel NN with shortcut connections, referred to as shortcut real-valued time-delay neural network (SVDEN), where trainable neuron-wise shortcut connections are added between the input and output layers. Second, we implement a NN pruning algorithm that gradually removes connections corresponding to minimal weight magnitudes in each layer. Simulation and experimental results show that SVDEN with pruning achieves better performance for compensating frequency-dependent quadrature imbalance and power amplifier nonlinearity than other NN-based and Volterra-based models, while requiring less or similar complexity.**

I. Introduction

Radio frequency (RF) direct conversion transceivers suffer from multiple hardware impairments due to analog hardware imperfections [2] such as non-ideal digital-to-analog converters (DACs), nonlinear active lowpass filters (LPFs), imperfect local oscillators (LOs), and nonlinear power amplifiers (PAs). These impairments induce various signal distortions which degrade the quality of the transmitted signal, leading to reduced performance in terms of throughput [3]. These impairments can be mitigated separately by different algorithms, but separate optimization of each algorithm makes their combination not globally optimal.

PA nonlinearity is one of the major hardware impairments [4]. In the frequency domain, PA nonlinearity materializes as in-band errors and out-of-band emissions due to intermodulation and harmonic products [5]. PAs further exhibit memory effects during operation over large bandwidths [6], i.e., past input signals have nonlinear effects on the instantaneous output of the PA. To linearize the PA, it is customary to apply digital predistortion (DPD) [7], which compensates for the signal distortion caused by the PA nonlinearity, so that the cascade of the DPD and the PA is a linear system. Quadrature (I/Q) imbalance is another major impairment [8], which commonly reflects as gain and phase mismatches, where the gain mismatch is introduced by the gain difference of DACs and LPFs between the in-phase (I) and quadrature (Q) branches, and the phase mismatch is caused by the LO imperfection during up- and down-conversions. Similar to the PA, the I/Q imbalance introduces nonlinear distortions with memory effects due to the nonlinear LPFs and DACs. Separate impairment mitigation of the PA and I/Q modulator has some shortcomings, as nonlinear mixing of the individual effects occurs. While some methods have been proposed to mitigate both impairments jointly, they suffer from either limited performance or high computational complexity [9], [10].

Several methods have been proposed to mitigate the I/Q imbalance and PA nonlinearity: Volterra series-based [8], [9], [11]–[16], NN-based [1], [10], [17]–[22], and finite impulse response (FIR) filter-based [23], [24] methods. The works [8], [23], [24] only focus on I/Q imbalance, while [11]–[13], [15], [16], [25] propose simplified versions of Volterra series [26] focusing only on the PA nonlinearity. Their performance is limited when both impairments occur [9]. Joint impairment mitigation of both the I/Q modulator and PA is investigated in [9], which extends the parallel Hammerstein (PH) method [11] by the FIR I/Q imbalance model so that the extended PH allows to jointly mitigate both I/Q modulator and PA impairments. Its performance, however, is limited for highly nonlinear PAs and I/Q modulators due to the simplification of the Volterra series and the linearity of FIR filters. All above mentioned Volterra-based models can improve performance by increasing the nonlinear order and memory length, but at the expense of an exponentially increasing complexity, which limits their utilization in practice [14].

As an alternative to Volterra-based methods, NNs for I/Q-PA impairments mitigation are studied in [1], [10], [17]–[22]. Among them, the multilayer perceptron (MLP) is mostly chosen due to easy deployment and training. Based on the MLP, the real-valued time-delay neural network (RVTDNN) was proposed for PA behavioral modeling [17]. It allows to learn nonlinearities with memory effects by feeding real-valued I and Q components of the original complex-valued signal with time-delays. Various variants of the RVTDNN have been later proposed [1], [10], [19]–[22]. The works [1], [19], [20] only focus on the PA nonlinearity, while [10], [21] and [22] consider both frequency-flat I/Q imbalance and PA nonlinearity in single-input single-output (SISO) and multiple-input multiple-output (MIMO) transmitters, respectively. Specifi-

This work was presented in part at the IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan [1].

Y. Wu is with Ericsson Research and Chalmers University of Technology, Gothenburg, Sweden (email: yibo@chalmers.se).

U. Gustavsson is with Ericsson Research, Gothenburg, Sweden (e-mail: ulf.gustavsson@ericsson.com).

A. Graell i Amat and H. Wymeersch are with Chalmers University of Technology, Gothenburg, Sweden (alexandre.graell@chalmers.se; henkw@chalmers.se).

This work was supported by the Swedish Foundation for Strategic Research (SSF), grant no. I19-0021.

cally, our recent work [1] combines residual learning with RVTDNN, which is demonstrated to improve performance for PA nonlinearity mitigation as well as reduce complexity compared with other RVTDNN variants. Similar performance improvements of using residual learning NNs are also shown in [27] for DPD, and [28] for compensating nonlinearities of a fiber-optic link. None of these NN-based models consider the mitigation of nonlinear frequency-dependent I/Q imbalance, which is considerable in practice [8]. More importantly, the high-complexity problem of NN-based models is not tackled except in our previous work [1], which limits their usages in practice.

In this paper, we investigate the performance and complexity of impairment mitigation models for the direct conversion transceiver with multiple hardware impairments. Particularly, we consider the joint mitigation of nonlinear frequency-dependent I/Q imbalance and PA nonlinearity. Our contributions are summarized as follows:

- We propose a shortcut connection NN based on the RVTDNN [17], referred to as shortcut real-valued time-delay neural network (SVDEN), to compensate for signal distortions caused by multiple hardware impairments, including PA nonlinearity and nonlinear frequency-dependent I/Q imbalance. Experimental results show that SVDEN yields better performance compared to state-of-the-art methods, while simultaneously exhibiting less complexity.
- We find that during the training of SVDEN, neurons in the first hidden layer fed by shorter lag input signals contribute more to the output with larger weight magnitudes.
- We propose and analyze a NN connection pruning algorithm to reduce complexity. Unimportant neural connections, i.e., those with weights with small magnitude, are gradually removed during the pruning process. Results show that pruning allows SVDEN to achieve better mitigation performance with less complexity.
- We evaluate the mitigation performance of different methods for a large complexity range. Experimental results illustrate that SVDEN with proper pruning factor performs the best over all complexity levels.

This paper extends [1] by generalizing to a multiple hardware impairments system including the PA and I/Q modulator. The weighted shortcut connections and pruning algorithm are novel.

II. System Model

The block diagram of a direct conversion transmitter is shown in Fig. 1. The hardware impairments of the DACs, LPFs, LO, and the PA introduce I/Q imbalance and PA nonlinearity, which the DPD placed before the hardware components tries to compensate. We now describe I/Q imbalance, PA nonlinearity, and DPD in detail.

A. I/Q Imbalance

As shown in Fig. 1, considering a discrete-time baseband signal $x(n)$ to be modulated by the I/Q modulator, its real and imaginal parts, $x_{\mathrm{I}}(n)$ and $x_{\mathrm{Q}}(n)$ are sent to the I and Q branches of the modulator, respectively. We consider both wideband and frequency-dependent I/Q imbalances. The wideband I/Q imbalance is due to memoryless nonlinearities of non-ideal DACs caused by quantization noise and clipping, while the frequency-dependent I/Q imbalance is due to nonlinearities with memory effects of imperfect and non-equal LPFs. The combination of DAC and LPF is represented by the nonlinear function $f_{\mathrm{I}} : \mathbb{R}^{L_1+1} \to \mathbb{R}$ and $f_{\mathrm{Q}} : \mathbb{R}^{L_1+1} \to \mathbb{R}$ for the I and Q branches, respectively, where L_1 is the memory length. Denote the output of the DAC-LPF for the I and Q branches as $s_{\mathrm{I}}(n)$ and $s_{\mathrm{Q}}(n)$, respectively. Their input-output relations can be expressed as

$$s_{\mathrm{I}}(n) = f_{\mathrm{I}}(x_{\mathrm{I}}(n), \ldots, x_{\mathrm{I}}(n-L_1)) = f_{\mathrm{I}}(\boldsymbol{x}_{\mathrm{I}}^{L_1}), \quad (1)$$

$$s_{\mathrm{Q}}(n) = f_{\mathrm{Q}}(x_{\mathrm{Q}}(n), \ldots, x_{\mathrm{Q}}(n-L_1)) = f_{\mathrm{Q}}(\boldsymbol{x}_{\mathrm{Q}}^{L_1}), \quad (2)$$

where $\boldsymbol{x}_{\mathrm{I}}^{L_1} = [x_{\mathrm{I}}(n), \ldots, x_{\mathrm{I}}(n-L_1)]^{\mathsf{T}}$, and $\boldsymbol{x}_{\mathrm{Q}}^{L_1} = [x_{\mathrm{Q}}(n), \ldots, x_{\mathrm{Q}}(n-L_1)]^{\mathsf{T}}$.

The DAC-LPF outputs are up-converted by mixers, where a phase imbalance ϕ is introduced, caused by LO imperfection. The output of the I/Q modulator is

$$z(n) = z_{\mathrm{I}}(n) + \jmath z_{\mathrm{Q}}(n), \quad (3)$$

where $z_{\mathrm{I}}(n) = s_{\mathrm{I}}(n) - \sin(\phi) s_{\mathrm{Q}}(n)$ and $z_{\mathrm{Q}}(n) = \cos(\phi) s_{\mathrm{Q}}(n)$. Equation (3) can be rewritten as

$$\begin{aligned} z(n) &= s_{\mathrm{I}}(n) - \sin(\phi) s_{\mathrm{Q}}(n) + \jmath \cos(\phi) s_{\mathrm{Q}}(n) \\ &= s_{\mathrm{I}}(n) + \jmath e^{\jmath\phi} s_{\mathrm{Q}}(n) \\ &= f_{\mathrm{I}}(\boldsymbol{x}_{\mathrm{I}}^{L_1}) + \jmath e^{\jmath\phi} f_{\mathrm{Q}}(\boldsymbol{x}_{\mathrm{Q}}^{L_1}), \end{aligned} \quad (4)$$

Due to the difference between DACs and LPFs of the I and Q branches, f_{I} and f_{Q} present different nonlinearities and memory effects, which leads to I/Q imbalances with both frequency-independent and frequency-dependent components. For ease of notation (4), we use a single function $f_{\mathrm{IQ}} : \mathbb{C}^{L_1+1} \to \mathbb{C}$ with memory length L_1 to represent the I/Q modulator system, so (4) can be rewritten as

$$z(n) = f_{\mathrm{IQ}}(x(n), \ldots, x(n-L_1)) = f_{\mathrm{IQ}}(\boldsymbol{x}^{L_1}), \quad (5)$$

where $\boldsymbol{x}^{L_1} = [x(n), \ldots, x(n-L_1)]^{\mathsf{T}}$. Note that for an ideal I/Q modulator $\phi = 0$, $L_1 = 0$, and $z(n) = x(n)$.

B. PA Nonlinearity

The modulated signal $z(n)$ is amplified by the PA, which behaves as a nonlinear system with memory effects, i.e., the PA output at any time instant depends on the current instantaneous input and previous inputs. Memory effects are mainly due to the frequency-dependent behavior of the PA and thus more considerable for wideband signals. We define the PA as a function $f_{\mathrm{PA}} : \mathbb{C}^{L_2+1} \to \mathbb{C}$ with input $z(n)$ and output $y(n)$, and memory length L_2,

$$y(n) = f_{\mathrm{PA}}(z(n), \ldots, z(n-L_2)) = f_{\mathrm{PA}}(\boldsymbol{z}^{L_2}), \quad (6)$$

where $\boldsymbol{z}^{L_2} = [z(n), \ldots, z(n-L_2)]^{\mathsf{T}}$. For an ideal PA, $L_2 = 0$ and $y(n) = Gz(n)$, G being the PA gain.

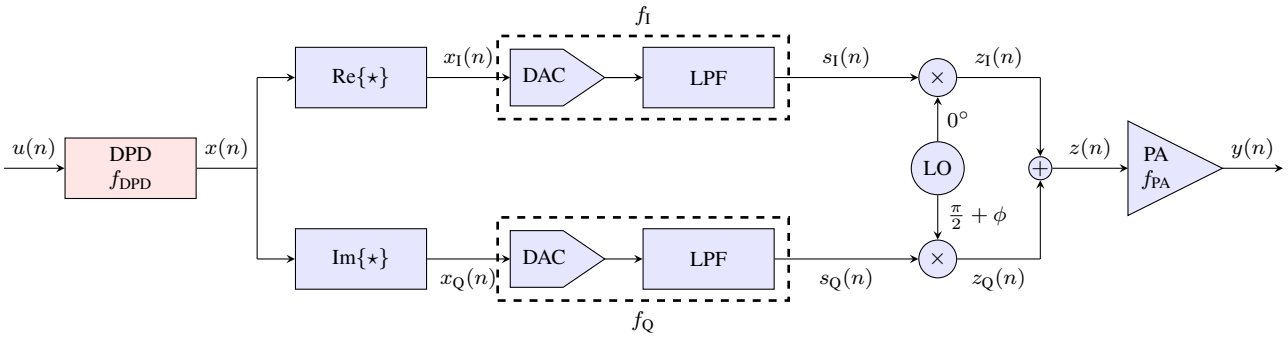

Fig. 1: Block diagram of the DPD-I/Q-PA system. The DPD block compensates for signal distortions caused by multiple hardware impairments in the direct conversion transmitter including non-ideal DACs, nonlinear LPFs, imperfect LO, and nonlinear PA.

C. Digital Predistortion

The DPD is represented by the function $f_{\mathrm{DPD}} : \mathbb{C}^{L_3+1} \to \mathbb{C}$ with memory length L_3 and input signal $u(n)$,

$$x(n) = f_{\mathrm{DPD}}(u(n), \ldots, u(n-L_3)) = f_{\mathrm{DPD}}(\boldsymbol{u}^{L_3}), \quad (7)$$

where $\boldsymbol{u}^{L_3} = [u(n), \ldots, u(n-L_3)]^{\mathsf{T}}$.

Substituting (5) into (6), we can rewrite $y(n)$ as

$$\begin{aligned} y(n) =& f_{\mathrm{PA}}(f_{\mathrm{IQ}}(\boldsymbol{x}^{L_1}), \ldots, f_{\mathrm{IQ}}(\boldsymbol{x}^{L_1}_{n-L_2})) \\ =& f_{\text{IQ-PA}}(x(n), \ldots, x(n-L_1-L_2)) \\ =& f_{\text{IQ-PA}}(\boldsymbol{x}^{L_1+L_2}), \end{aligned} \quad (8)$$

where the function $f_{\text{IQ-PA}} : \mathbb{C}^{L_1+L_2+1} \to \mathbb{C}$ represents the I/Q-PA system. The system resulting from the cascade of the I/Q modulator and the PA has memory length $(L_1 + L_2)$.

The input-output relation of the whole system is obtained by substituting (7) into (8) as

$$\begin{aligned} y(n) =& f_{\text{IQ-PA}}(f_{\mathrm{DPD}}(\boldsymbol{u}_n^{L_3}), \ldots, f_{\mathrm{DPD}}(\boldsymbol{u}^{L_3}_{n-L_1-L_2})) \\ =& f_{\text{DPD-IQ-PA}}(\boldsymbol{u}^{L_1+L_2+L_3}), \end{aligned} \quad (9)$$

where the function $f_{\text{DPD-IQ-PA}} \in \mathbb{C}^{L_1+L_2+L_3} \to \mathbb{C}$ denotes the DPD-I/Q-PA system with memory length $(L_1 + L_2 + L_3)$.

Ideally, the DPD would make the cascade DPD-I/Q-PA linear, in which case (9) would reduce to a linear function. Unfortunately, this is infeasible in practice due to the presence of hardware impairments such as PA clipping and thermal noise, which can not be compensated for. DPD methods aim, therefore, to make the DPD-I/Q-PA system as linear as possible by minimizing the mean squared error (MSE) between the PA output $y(n)$ and DPD input $u(n)$,

$$\hat{f}_{\mathrm{DPD}} = \arg\min_{f_{\mathrm{DPD}}} \mathbb{E}[|f_{\text{DPD-IQ-PA}}(\boldsymbol{u}^{L_1+L_2+L_3}) - u(n)|^2], \quad (10)$$

where $\mathbb{E}[\cdot]$ denotes expectation.

Example (I/Q imbalance)**.** *If we only consider frequency-independent I/Q imbalance introduced by a phase imbalance ϕ and ignore nonlinearities, memory effects, and the PA gain in the I/Q-PA system, $f_{\text{IQ-PA}}$ reduces to a linear function as*

$$y(n) = x_I(n) - \sin(\phi)x_Q(n) + \jmath\cos(\phi)x_Q(n),$$

and its inverse, $f^{-1}_{\text{IQ-PA}}$, can be explicitly calculated as

$$x(n) = y_I(n) + \frac{\sin(\phi)}{\cos\phi}y_Q(n) + \jmath\frac{1}{\cos(\phi)}y_Q(n).$$

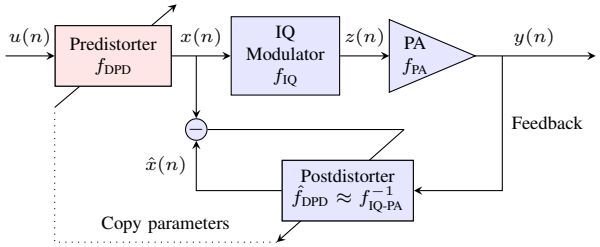

Fig. 2: Block diagram of the ILA. To learn the inverse behavior of the I/Q-PA system, $f^{-1}_{\text{IQ-PA}}$, the postdistorter is learned by minimizing the error between its output, $\hat{x}(n)$, and the input of the IQ modulator, $x(n)$. The learned postdistorter is then utilized as the predistorter.

III. Preliminaries

A. DPD-parameter Identification by ILA

In practice, estimating the parameters of the DPD function f_{DPD} through (10) is troublesome as the I/Q-PA system is generally a combination of black boxes, i.e., unknown $f_{\text{IQ-PA}}$. The direct learning architecture (DLA) [29] solves this problem by approximating $f_{\text{IQ-PA}}$ as a differential model, which allows to iteratively identify DPD parameters through a gradient-based method. However, the accuracy of the identified DPD is seriously affected by the accuracy of the approximated $f_{\text{IQ-PA}}$, and the identification process of DLA is highly complex due to numerous updating iterations.

Instead, the indirect learning architecture (ILA) [26] indirectly estimates the DPD parameters by learning the inverse behavior of the I/Q-PA system, i.e., $f^{-1}_{\text{IQ-PA}}$, referred to as the *postdistorter*, which is then used as the *predistorter* for DPD [30]. The block diagram of the ILA is depicted in Fig. 2. The parameters of the postdistorter are optimized by minimizing the MSE between the postdistorter output signal, $\hat{x}(n)$, and the IQ modulator input signal, $x(n)$. The optimization problem becomes

$$\hat{f}_{\mathrm{DPD}} = \arg\min_{f^{-1}_{\text{IQ-PA}}} \mathbb{E}[|f^{-1}_{\text{IQ-PA}}(\boldsymbol{y}^{L_1+L_2}) - x(n)|^2], \quad (11)$$

where $\boldsymbol{y}^{L_1+L_2} = [y(n), ..., y(n-L_1-L_2)]^{\mathsf{T}}$. ILA is the most used identification method due to simple implementation and excellent performance [31]. Therefore, we consider ILA as the identification method for DPD in this paper.

B. Shortcut Connections

Shortcut connections for NNs have been widely used to address vanishing/exploding gradients for deep networks [32]–[34]. Consider a stack of NN layers without shortcut connections to fit an underlying mapping $f(\cdot)$ with input $\boldsymbol{x}$ and output $\boldsymbol{y}$, i.e., $\boldsymbol{y} = f(\boldsymbol{x})$. Adding shortcut connections allows the input of the first layer, $\boldsymbol{x}$, to bypass several layers and directly contribute to the output of the last layer, $\boldsymbol{y}$, with matrix weight $\boldsymbol{W}$. This input-output relation of the shortcut network can be expressed as

$$\boldsymbol{y} = \underbrace{\boldsymbol{W}\boldsymbol{x}}_{f_{\text{short}}(\boldsymbol{x})} + \underbrace{f(\boldsymbol{x}) - \boldsymbol{W}\boldsymbol{x}}_{f_{\text{remain}}(\boldsymbol{x})}, \tag{12}$$

where we refer to the shortcut component $\boldsymbol{W}\boldsymbol{x}$ as the *shortcut function*, denoted by f_{short}, and the remaining component $f(\boldsymbol{x}) - \boldsymbol{W}\boldsymbol{x}$ as the *remaining function*, denoted by $f_{\text{remain}}(\boldsymbol{x})$.

Many variants of shortcut connections have been studied [27], [28], [32]–[34]. Works [27], [28] considered an identity mapping between the network input and output layers, while the authors in [32] used a similar identity mapping for intermediate layers. These works reduce $\boldsymbol{W}$ to an identity matrix or a linear projection matrix for matching dimension. In this case, (12) corresponds to *residual learning*, and the corresponding NN is known as a *residual network*. Similarly, [33] proposed shortcut connections for intermediate layers with two gate parameters that control both f_{short} and f_{remain} to allow flexible residual learning, where $\boldsymbol{W}$ reduces to a diagonal matrix for gating. The work [34] used the *inception layer* consisting of a shortcut branch and other deep learning branches.

IV. Shortcut Neural Networks for DPD

In this section, we introduce the proposed SVDEN to mitigate impairments of the I/Q-PA system and NN pruning to reduce computational complexity.

A. Shortcut Connections for I/Q-PA System

Using the ILA, the postdistorter learns the inverse behavior of the IQ-PA system, i.e., $f_{\text{IQ-PA}}^{-1}$. We apply the shortcut connection to $f_{\text{IQ-PA}}^{-1}$, in which case $f_{\text{IQ-PA}}^{-1}$ is decomposed into a linear and a nonlinear part. Consider $f_{\text{IQ-PA}}^{-1}$ as the underlying mapping f in (12) for a NN-based postdistorter to learn. Considering shortcut connections, the mapping f to be fitted can be reformulated as

$$x(n) = \boldsymbol{W}_{\text{s}}\boldsymbol{y}^{L_1+L_2} + f_{\text{IQ-PA}}^{-1}(\boldsymbol{y}^{L_1+L_2}) - \boldsymbol{W}_{\text{s}}\boldsymbol{y}^{L_1+L_2}, \tag{13}$$

where $\boldsymbol{W}_{\text{s}}$ denotes the complex-valued *shortcut matrix weight* and (13) is complex-valued.

For the choice of $\boldsymbol{W}_{\text{s}}$, we propose to only bypass the memoryless part of $\boldsymbol{y}^{L_1+L_2}$, i.e., shortcut connecting $y(n)$, in which case $\boldsymbol{W}_{\text{s}}$ reduces to a complex scalar w_{s}. This choice is substantiated by practical considerations of the inverse of the I/Q-PA system, in which case only the linear relation between $y(n)$ and $x(n)$ is directly passed. This linear input-output relation exists in many other behavioral models such as the memory polynomial model, where this relation can be found by the first order polynomial with memoryless input [13, Eq. (19)].

Substituting the reduced shortcut weight w_{s} into (13), we have

$$x(n) = \underbrace{w_{\text{s}}y(n)}_{f_{\text{short,IQ-PA}}^{-1}(y(n))} + \underbrace{f_{\text{IQ-PA}}^{-1}(\boldsymbol{y}^{L_1+L_2}) - w_{\text{s}}y(n)}_{f_{\text{remain,IQ-PA}}^{-1}(\boldsymbol{y}^{L_1+L_2})}, \tag{14}$$

where $f_{\text{short,IQ-PA}}^{-1}$ and $f_{\text{remain,IQ-PA}}^{-1}$ denote the shortcut and remaining functions for the inverse of the I/Q-PA system $f_{\text{IQ-PA}}^{-1}$.

With shortcut connections, learning the original unknown $f_{\text{IQ-PA}}^{-1}$ reduces to learning the remaining nonlinear behavior $f_{\text{remain,IQ-PA}}^{-1}$. We remark that the use of shortcut connections is different from the residual learning in [32], as we only extract a specific part of the input signal, i.e., the current input signal $y(n)$, instead of the whole input sequence. Note that the remaining function $f_{\text{remain,IQ-PA}}^{-1}$ has a physical meaning: it represents the remaining linear and nonlinear behaviors of the inverse of the I/Q-PA system (whereas the residual function [32] does not). We only apply the shortcut connections between the input and output of the network, whereas the work [32] uses it between several intermediate layers.

Example (I/Q imbalance)**.** *Considering only phase imbalance as in* (II-C) *and* (II-C)*, we can calculate w_s by setting the right hand side of* (II-C) *equal to $w_s y(n)$ as*

$$w_s = \frac{1 + \sin(\phi) + \cos(\phi)}{2\cos(\phi)} + \jmath\frac{1 - \sin(\phi) - \cos(\phi)}{2\cos(\phi)}. \tag{15}$$

Similarly, considering no hardware impairments in the I/Q-PA system and ignoring the PA gain would lead to $w_s = 1$. These values of w_s can be used for parameter initialization depending on the prior knowledge of the I/Q-PA system.

B. SVDEN Architecture

Based on the MLP, we propose a novel NN by considering shortcut connections (14) for the inverse of the I/Q-PA system, referred to as SVDEN, and Fig. 3 shows the block diagram of SVDEN with arbitrary connections pruned by the NN pruning algorithm (dotted lines), which is described in Section IV-C. SVDEN consists of K fully connected layers with $(K-2)$ hidden layers. The number of neurons in layer k is denoted by D_k. The input vector of layer k is denoted by $\boldsymbol{s}_k \in \mathbb{R}^{D_{k-1}}$ for $k > 1$. The input and output vectors of the input and output layers are $\boldsymbol{s}_1$ and $\boldsymbol{s}_{K+1}$. Depending on the use of SVDEN as the postdistorter for parameter estimation or the predistorter for DPD deployment, the input-output of SVDEN will be interchanged. As shown in Fig. 2, during the DPD parameter estimation, $y(n)$ corresponds to $\boldsymbol{s}_1$ and $x(n)$ corresponds $\boldsymbol{s}_{K+1}$, and vice versa when SVDEN is deployed as DPD.

Define a complex-valued signal with sample $s_{\text{in}}(n) = s_{\text{in}}^{\text{I}}(n) + \jmath s_{\text{in}}^{\text{Q}}(n)$ at time instant n as the input of SVDEN. The real-valued input vector $\boldsymbol{s}_1$ is formed by concatenating the current and previous time instants of the input signal,

$$\boldsymbol{s}_1 = [s_{\text{in}}^{\text{I}}(n), s_{\text{in}}^{\text{Q}}(n), \ldots, s_{\text{in}}^{\text{I}}(n-M), s_{\text{in}}^{\text{Q}}(n-M)]^{\mathsf{T}}, \tag{16}$$

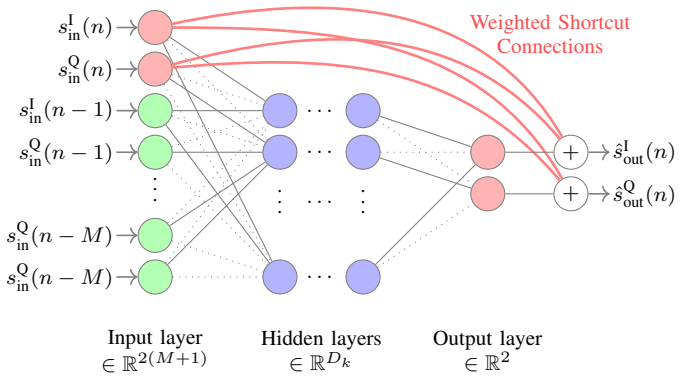

Fig. 3: Architecture of the proposed SVDEN with arbitrary connections being pruned. Dotted and solid lines between neurons represents pruned and remained connections. Fed by the real-valued I and Q components of the current and historical time instant signals, SVDEN returns estimations of the real-valued I and Q components of the current time instant output signal. When using ILA to estimate DPD parameters, $s_{\mathrm{in}}(n) = y(n)$ and $s_{\mathrm{out}}(n) = x(n)$.

where M denotes the number of time delays for the input signal. The time-delayed inputs allow SVDEN to capture the memory effects of the transmitter, and separating the real-valued signals allows the use of a simple real-valued training algorithm. In total, the number of neurons for the input layer is $D_1 = (2M + 2)$.

We denote the weight matrix that connects layer $k-1$ and k by $\boldsymbol{W}_k \in \mathbb{R}^{D_k \times D_{k-1}}$, for $k > 1$, the jth column weight vector of $\boldsymbol{W}_k$ by $[\boldsymbol{W}_k]_j \in \mathbb{R}^{D_k}$, and the corresponding bias vector by $\boldsymbol{b}_k \in \mathbb{R}^{D_k}$. For $k > 1$, layers $k-1$ and k are fully connected as

$$\boldsymbol{s}_{k+1} = \sigma(\boldsymbol{W}_k \boldsymbol{s}_k + \boldsymbol{b}_k), \tag{17}$$

where σ denotes the element-wise activation function. To output a full range of values, the output layer is a linear layer, i.e., function σ for the output layer is an identity mapping function, with number of neurons $D_K = 2$ corresponding to the I and Q output signals.

The shortcut connections in (14) are implemented in SVDEN by weighted *shortcut connections* between the instantaneous input and output signals as

$$f^{-1}_{\text{short,IQ-PA}} = \tilde{\boldsymbol{W}}_{\mathrm{s}} [s^{\mathrm{I}}_{\mathrm{in}}(n), s^{\mathrm{Q}}_{\mathrm{in}}(n)]^{\mathsf{T}}, \tag{18}$$

where the trainable weight matrix $\tilde{\boldsymbol{W}}_{\mathrm{s}} \in \mathbb{R}^{2\times 2}$ corresponds to the complex-valued w_{s} in (14). Without any prior knowledge of the I/Q-PA system impairments, we can initialize $\tilde{\boldsymbol{W}}_{\mathrm{s}}$ as an identity matrix $\mathbb{I} \in \mathbb{R}^{2\times 2}$ since $w_{\mathrm{s}} = 1$. Assuming we know a prior, the phase imbalance ϕ, of the I/Q-PA system, we can initialize

$$\tilde{\boldsymbol{W}}_{\mathrm{s}} = \begin{bmatrix} 1 & \sin\phi/\cos\phi \\ 0 & 1/\cos\phi \end{bmatrix} \tag{19}$$

with w_{s} given in (15). The shortcut connections are shown by red lines in Fig. 3. The remaining function in (14) is implemented by the hidden layers of SVDEN. Thus, the output of the output layer can be expressed as

$$\hat{\boldsymbol{s}}_{\mathrm{out}} \triangleq \boldsymbol{s}_{K+1} = \underbrace{\tilde{\boldsymbol{W}}_{\mathrm{s}} [s^{\mathrm{I}}_{\mathrm{in}}(n), s^{\mathrm{Q}}_{\mathrm{in}}(n)]^{\mathsf{T}}}_{f^{-1}_{\text{short,IQ-PA}}} + \underbrace{\boldsymbol{W}_K \boldsymbol{s}_K + \boldsymbol{b}_K}_{f^{-1}_{\text{remain,IQ-PA}}}, \tag{20}$$

where $\hat{\boldsymbol{s}}_{\mathrm{out}} \in \mathbb{R}^2$ consists of the I and Q output signal estimations $\hat{s}^{\mathrm{I}}_{\mathrm{out}}(n)$ and $\hat{s}^{\mathrm{Q}}_{\mathrm{out}}(n)$ of the complex-valued output signal $s_{\mathrm{out}}(n)$ at time instant n, respectively.

Denote all weight matrices and bias vectors as $\check{\boldsymbol{W}} = \{\boldsymbol{W}_1, \ldots, \boldsymbol{W}_K, \tilde{\boldsymbol{W}}_{\mathrm{s}}\}$ and $\check{\boldsymbol{b}} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K\}$. $\check{\boldsymbol{W}}$ and $\check{\boldsymbol{b}}$ can be learned through gradient descent by minimizing the MSE between the estimation $\hat{\boldsymbol{s}}_{\mathrm{out}}$ and observation $\boldsymbol{s}_{\mathrm{out}}$,

$$(\check{\boldsymbol{W}}, \check{\boldsymbol{b}}) = \arg\min_{\check{\boldsymbol{W}}, \check{\boldsymbol{b}}} \mathbb{E}[|\boldsymbol{s}_{\mathrm{out}} - \hat{\boldsymbol{s}}_{\mathrm{out}}|^2]. \tag{21}$$

Algorithm 1 : Magnitude-based pruning for layer k.

Input: Total training step N. Pruning interval ΔN.

for $n = 1 \to N$ **do**
  if $n/\Delta N =$ integer **then**
    Calculate η_n using (23)
    Calculate N_{p} using (24)
    Zero N_{p} weights of smaller magnitude in $\boldsymbol{W_k}$
    Zero the corresponding N_{p} masks in $\boldsymbol{M_k}$
  else
    Update weights in $\boldsymbol{W}_k$ with non-zero masks via back-propagation
  end if
end for
Remove $\boldsymbol{M}_k$

C. Neural Network Pruning

NNs have been shown to achieve good performance in many tasks. However, the high computation cost makes the deployment of NNs challenging in resource-constrained scenarios where the resource overhead for each chain, and thus for each DPD, is limited. Hence, it is crucial to reduce the computation cost of NNs for DPD.

To reduce the computational cost of NNs for inference, one popular technique that has been studied in recent years is NN pruning [35]–[38], which sets small valued weights to zero and removes corresponding connections. The resulting network is commonly known as sparse NN, where the *sparsity*, $\eta \in [0, 1]$, is defined as the proportion of NN weights that are zero valued. With pruning, the weight matrix becomes sparse, which brings two main practical benefits: (i) sparse NNs can be compressed to use less storage [35], and (ii) the resulting sparse matrix-vector multiplication can be accelerated by specialized hardware [39]–[42].

We apply the pruning method in [37] to increase the sparsity of a pre-trained SVDEN. This pruning method is a variant of the magnitude-based weight pruning scheme that prunes less important weights according to their magnitude. Pruning works on each layer by adding a binary mask with the same size as the layer's weight matrix, in which a zero indicates that the weight is pruned. Let $\boldsymbol{M}_k \in \mathbb{R}^{D_k \times D_{k-1}}$ denote the

binary mask matrix of layer k. The connection between layer $(k-1)$ and k in (17) with pruning can be rewritten as

$$\boldsymbol{s}_{k+1} = \sigma((\boldsymbol{M}_k \odot \boldsymbol{W}_k)\boldsymbol{s}_k + \boldsymbol{b}_k), \tag{22}$$

where $\odot$ denotes the Hadamard product operator. Note that shortcut connections in SVDEN are not pruned so as to keep the shortcut function $f_{\text{short}}^{\text{IQ-PA}}$ always active.

Given a total number of training steps N, weights are pruned every ΔN steps, referred to as *pruning process*. Denote the sparsity η at step n as η_n, which is gradually increased to the desired sparsity η_{d} by [37]

$$\eta_n = \eta_{\text{d}} - \eta_{\text{d}} \left(1 - \frac{\lfloor n/\Delta N \rfloor}{N}\right)^3. \tag{23}$$

The intuition behind (23) is to prune rapidly at the beginning and gradually prune less weights when the sparsity grows high. After each pruning step, non-pruned weights are retrained for $N-1$ training steps, referred to as *retraining process*, to alleviate the loss caused by pruning.

The pruning for layer k of SVDEN is in Algorithm 1. During each pruning step, η_n is calculated using (23). To meet η_n for the layer with a total number of weights N_{w}, the number of weights N_{p} needed to be pruned is calculated by

$$N_{\text{p}} = N_{\text{w}} \times (1 - \eta_n). \tag{24}$$

Then, weights in $\boldsymbol{W}_k$ are sorted by magnitudes, and the N_{p} weights of smaller magnitude are masked to zero by setting the corresponding values in $\boldsymbol{M}_k$ to zero. During each retraining step, weights in $\boldsymbol{W}_k$ are updated through $N-1$ back-propagation steps. Once pruning is done, $\boldsymbol{M}_k$ is removed.

D. Computational Complexity

There are various ways to measure the complexity of an algorithm, such as the Bachmann-Landau measure $\mathcal{O}(\cdot)$, running time, number of parameters, and number of floating point operationss (FLOPs). While the Bachmann-Landau measure describes the asymptotic complexity of the algorithm, it is not accurate enough for the use of DPD in practice. The running time for different DPD models highly depends on both the software and hardware setups and is not a reproducible measure. For instance, graphic processing units (GPUs) can offer an order of magnitude improvement in accelerating NN computation. The number of parameters has been widely used to measure the complexity of Volterra series-based models because it also reflects the number of memories for hardware implementation [14]. However, it is still an approximation of the true computational complexity, as it does not account for any nonlinear operations in the algorithm. In contrast, the number of FLOPs is a commonly used measure that can accurately measure every addition, subtraction, and multiplication operation, which are the dominating operations for both Volterra series-based models and NNs. Hence, the number of FLOPs has been widely utilized to measure computational complexity for Volterra series-based models [14]–[16] and various types of NNs [35]–[42]. When it comes to the hardware implementing of DPD, the number of FLOPs is also usually used to measure the throughput [14]. In this paper, we therefore consider the number of FLOPs as an approximate measure of the complexity as in [14]–[16], [35]–[42].

We focus on the *running complexity* [14] of the DPD, which is defined as the number of calculations required for the inference of each output sample. Unlike the *training complexity* for estimating DPD parameters that is usually done off-line, the running complexity is a real-time cost, which heavily limits the system overhead. It can be quantified by the number of multiplications and additions operated, where each real-valued multiplication or addition accounts for one FLOP [14, Table. I].

By applying pruning to NNs, both the storage requirement and the computational cost are accordingly reduced as the network sparsity increases, where the computational cost is customarily represented by the remaining number of FLOPs, see, e.g., [35]–[38]. Ideally, multiplications with zero-valued weights can be skipped [38], which can reduce the computation cost considerably as multiplications dominate the computational cost. In practice, this computational cost improvement can be realized by specialized hardware [39]–[42]. Specifically, the designed chip in [39] implements sparse vector-matrix multiplication and achieves substantial speedups and energy savings compared to a non-sparse NN running on a central processing unit (CPU) and a GPU. We measure the computational cost of SVDEN in terms of the number of FLOPs as[1]

$$C_{\text{SVDEN}} = 2(1 - \eta_{\text{d}}) \sum_{k=1}^{K-1} D_k D_{k+1} + 8, \tag{25}$$

where the factor 8 corresponds to the number of FLOPs introduced by the shortcut connections, and the number of FLOPs decreases accordingly with the desired network sparsity η_{d}. The computational complexity of other RVTDNN-based methods can be calculated in a similar way.

V. Experimental results

A. Setup

In this section, we describe the measurement setup of the I/Q-PA system, two performance metrics, and detailed training settings.

1) Measurement Setup: The measurement setup is based on the RF WebLab [43], which can be remotely accessed at www.dpdcompetition.com. Its block diagram is shown in Fig. 4. The block MATLAB includes all digital signal processing steps such as the DPD identification, DPD deployment, and artificial I/Q imbalance generation. In the transmission stage, digital signals generated by MATLAB are converted into analog signals by a vector signal transceiver (VST) PXIe-5646R VST, and then transmitted to the Gallium Nitride PA DUT (Cree CGH4006-TB) with a 40 dB linear driver. In the receiving stage, through a 30 dB attenuator, analog PA

[1]Depending on the hardware platform, the actual number of FLOPs may vary. For example, it is larger on CPUs and GPUs while it can be much smaller on specialized chips [39]–[42].

output signals are collected by the VST and then sent back to MATLAB.

An artificial I/Q imbalance is added before sending the signal to the RF WebLab. The gain imbalance is 1 dB, and the phase imbalance is $\phi = 8°$. Frequency-dependent I/Q imbalance is introduced using two 5-th order FIR lowpass elliptic filters in the I and Q branches with different filter parameters: minimum stopband attenuation of 60 dB (I) and 50 dB (Q), the peak-to-peak ripples of 0.1 dB (I) and 0.12 dB (Q), the normalized passband edge frequencies of 0.8 (I) and 0.85 (Q). For more details of the frequency response difference between these two filters, refer to [44]. The measured saturation point and measurement noise variance of the PA in RF WebLab are 24.1 V ($\approx$ 37.6 dBm of a 50 Ω impedance) and 0.0032, respectively. The output signal of the PA has an average power of 24.93 dBm, which corresponds to a theoretical normalized mean square error (NMSE) minimum [45] of -39.56 dB and a simulated adjacent channel power ratio (ACPR) minimum [45] of -49.92 dBc.

2) Metrics: We measure performance in terms of NMSE and ACPR. The NMSE is defined as

$$\text{NMSE} = 10\log_{10}\frac{\mathbb{E}[|y(n)-u(n)|^2]}{\mathbb{E}[|u(n)|^2]}, \tag{26}$$

and gives the all-band error in time-domain between the PA output signal and the DPD input signal. The ACPR is defined as

$$\text{ACPR} = 10\log_{10}\frac{\int_{\text{adj.}}|Y(f)|^2\mathrm{d}f}{\int_{\text{ch.}}|Y(f)|^2\mathrm{d}f}, \tag{27}$$

where $Y(f)$ denotes the Fourier transform of the PA output signal. The integration in the numerator and denominator is performed over one adjacent channel (the one with larger integration between the lower and upper adjacent channel) and the main channel, respectively. The ACPR evaluates the amount of out-of-band emission.

3) Benchmarks: All models use the ILA for DPD identification. For a fair comparison, we consider the extended PH [9] because it is designed to jointly mitigate frequency-dependent I/Q imbalance and PA nonlinearity. Other referred Volterra-based models [11]–[13] fail to address both impairments. The extended PH [9] uses the least squares algorithm for parameter identification, and its computation complexity is given in Appendix A.

We also consider four other RVTDNN-based models for comparison, namely RVTDNN [17], real-valued focused time-delay neural network (RVFTDNN) [21], augmented real-valued time-delay neural network (ARVTDNN) [10], and residual real-valued time-delay neural network (R2TDNN) [1]. All RVTDNN-based models including the proposed SVDEN use the ReLU activation function, the MSE loss function (20), and the Adam optimizer [46] for stochastic gradient descent with a fixed learning rate 0.001 and a mini-batch size of 256. All weights and biases are randomly initialized, while the shortcut weight matrix $\boldsymbol{W}_{\text{s}}$ in SVDEN is initialized as in (19) given $\phi = 8°$. All NN-based models are trained until convergence and then tested by independent data. The training and testing data

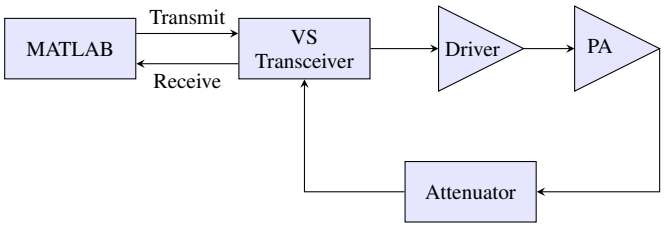

Fig. 4: Block diagram of the RF WebLab. Digital signals are transmitted and received by the block of MATLAB.

TABLE I: Number of neurons D_k in each hidden layer for RVTDNN, RVFTDNN, ARVTDNN, R2TDNN, and the proposed SVDEN. Note that $K = 5$ represents three hidden layers plus the input and output layers, and $D_2 = D_3 = D_4$. Varying D_k changes the number of FLOPs. Each value of D_k corresponds to a marker in Fig. 5 and Fig. 6.

	K	Varied D_k
RVTDNN [17]	5	$\{2, 4, 6, 8, 10, 12, 18, 27, 36, 48, 60\}$
RVFTDNN [21]	4	$\{2, 4, 7, 10, 13, 16, 24, 36, 50, 66, 83\}$
ARVTDNN [10]	5	$\{2, 3, 5, 7, 9, 11, 16, 25, 34, 46, 57\}$
R2TDNN [1]	5	$\{2, 4, 6, 8, 10, 12, 18, 27, 36, 48, 60\}$
SVDEN, $\eta_{\text{d}} = 0$	5	$\{2, 4, 6, 8, 10, 12, 18, 27, 36, 48, 60\}$
SVDEN, $\eta_{\text{d}} = 0.5$	5	$\{2, 4, 6, 8, 10, 12, 18, 27, 36, 48, 60\}$

are obtained by sending independently generated orthogonal frequency division multiplexing (OFDM) signals to the RF WebLab with the same settings including sampling frequency 200 MHz, signal length 10^6, and bandwidth 10 MHz. For all DPD schemes, we choose a memory length 3. For all NN-based DPD schemes, we set the number of hidden layers to 3, i.e., $K = 5$, except for RVFTDNN [21], which has only two hidden layers by construction. For each NN, the number of neurons is the same for all hidden layers, i.e., $D_2 = D_3 = D_4$. Note that modifying the number of neurons in the hidden layer changes the number of FLOPs.

B. Results

In this section, we investigate how the performance changes with respect to the complexity, memory length, and pruning rate.

1) Performance versus Complexity: Fig. 5 and Fig. 6 show the NMSE and ACPR as a function of the number of FLOPs for the extended PH, RVTDNN, RVFTDNN, ARVTDNN, R2TDNN [1], the proposed non-pruned SVDEN, and SVDEN with a pruning factor $\eta_{\text{d}} = 0.5$. The marks for NN-based schemes correspond to different numbers of neurons in the hidden layers, which are given in Table I. Pruned SVDEN is based on the same structure of non-pruned SVDEN. The ARVTDNN contains three augmented envelope terms of the input signal (amplitude and its square and cube) [10, Tab. II entry 11] at the input layer. For PH, the best results are selected with respect to the number of FLOPs through an exhaustive search of different values of its nonlinear order and filter length.

The proposed SVDEN with and without pruning achieves lower NMSE and ACPR results than all other DPD schemes for all number of FLOPs. Specifically, the PH has limited mitigation performance, flattens around a NMSE of -29.9 dB and an ACPR of -37.2 dBc, whereas SVDEN achieves a

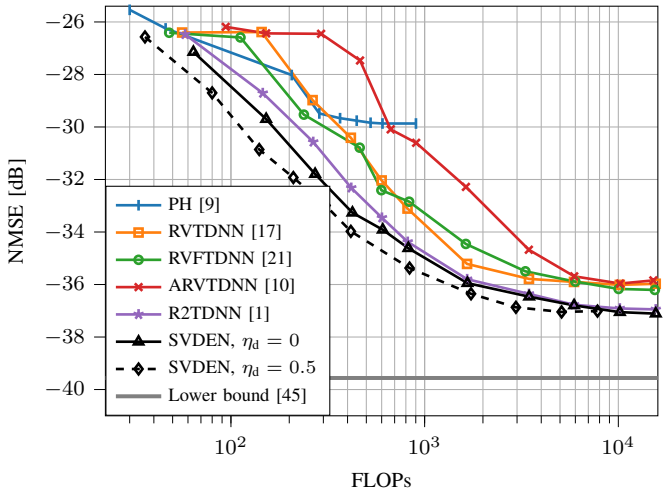

Fig. 5: NMSE as a function of the number of FLOPs for a DPD of memory length 3. The markers for PH [9] correspond to different sets of nonlinear order. The markers for RVTDNN [17], RVFTDNN [21], ARVTDNN [10], and SVDEN correspond to different numbers of neurons in the hidden layers, which are given in TABLE I. For SVDEN, $K = 5$ and $D_2 = D_3 = D_4$.

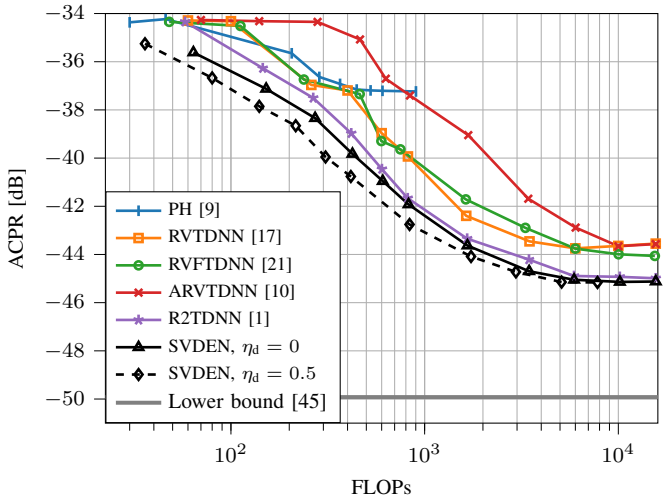

Fig. 6: ACPR as a function of the number of FLOPs for a DPD of memory length 3. The markers for PH [9] correspond to different sets of nonlinear order. The markers for RVTDNN [17], RVFTDNN [21], ARVTDNN [10], and SVDEN correspond to different numbers of neurons in the hidden layers, which are given in TABLE I. For SVDEN, $K = 5$ and $D_2 = D_3 = D_4$.

NMSE of -37.0 dB and an ACPR of -45.1 dBc. Compared with the R2TDNN [1], SVDEN yields sizable NMSE and ACPR gains for a number of FLOPs smaller than 500, which verifies the effectiveness of the shortcut weights in the shortcut connections. Furthermore, SVDEN with a pruning factor $\eta_d = 0.5$ requires even less number of FLOPs to achieve the same NMSE and ACPR compared with the non-pruned SVDEN, though this advantage vanishes as the size of SVDEN becomes large (FLOPs$>$ 3000).

2) Complexity-Restricted Scenario: We compare the mitigation performance of different DPD schemes in a limited complexity scenario for a number of FLOPs around 400. Fig. 7 shows the error spectrum of the PA output without DPD, with

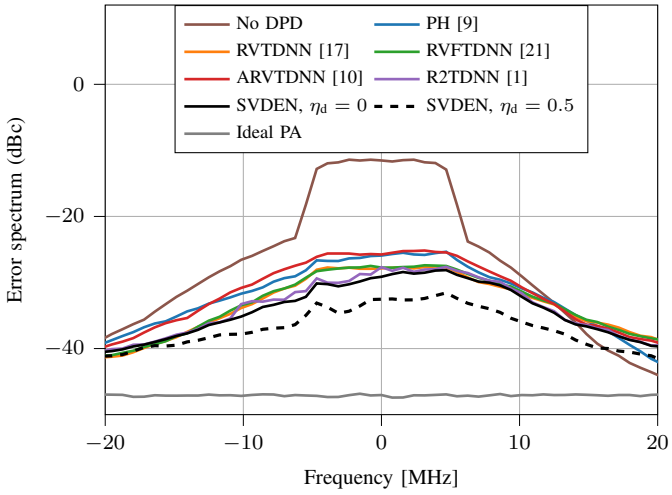

Fig. 7: Error spectrum between the actual and desired PA output signals for different models in a computation-restricted scenario with around 400 FLOPs.

TABLE II: NMSE and ACPR results of the PH [9], RVFTDNN [21] RVTDNN [17], ARVTDNN [10], and proposed SVDEN in Fig. 7. Note that $D_2 = D_3 = D_4$. The lower bound results are the minimum that can be achieved at an average output power 25.19 dBm.

	FLOPs	K, D_2	NMSE [dB]	ACPR [dBc]
No DPD	—	—	-17.84	-34.40
PH [9]	446	—	-29.69	-36.76
RVTDNN [17]	416	5, 8	-31.13	-37.91
RVFTDNN [21]	462	4, 10	-30.93	-37.54
ARVTDNN [10]	464	5, 7	-27.46	-35.08
R2TDNN [1]	418	5, 8	-32.06	-38.01
SVDEN, $\eta_d = 0$	424	5, 8	-33.26	-38.97
SVDEN, $\eta_d = 0.5$	416	5, 12	$\mathbf{-34.58}$	$\mathbf{-41.82}$
Lower bound [45]	—	—	-39.56	-49.92

DPD via PH, RVTDNN, RVFTDNN, ARVTDNN, SVDEN, pruned SVDEN, and of an ideal linear PA. The corresponding number of FLOPs, NMSE, and ACPR results are given in Table II. The pruned SVDEN is based on an original SVDEN with $C_{\text{svden}} = 818$ FLOPs and a pruning factor $\eta_d = 0.5$. For a fair comparison, the memory length for all DPD schemes is set to 3, and the number of FLOPs for each scheme is ≈ 400 by adjusting the number of neurons in the hidden layers for NN-based schemes and the nonlinear order for PH. Fig 8 shows the NMSE as a function of memory length, M, for the DPD schemes with the same structures in Fig. 7 and Table II but different memory length.

As shown in Fig 7, without DPD, there are considerable in-band and out-of-band distortions, which are not fully compensated by any of the DPD schemes due to residual unrecoverable distortions in the I/Q-PA system. The pruned SVDEN with $\eta_d = 0.5$ achieves the best performance with NMSE of -34.58 dB and ACPR of -41.82 dB, while requiring a similar number of FLOPs. As shown in Fig. 8, increasing the memory length yields NMSE improvements for most DPD schemes except for the ARVTDNN, in which case its performance is limited by the number of neurons in the hidden layer instead of the memory length. The NMSE improvement is more substantial for the proposed SVDEN,

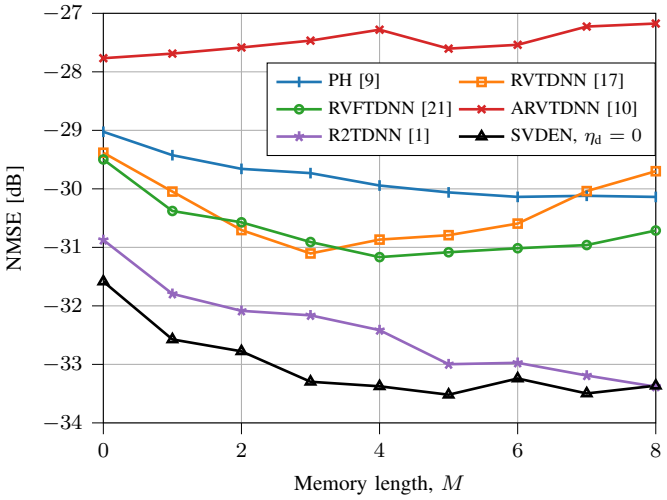

Fig. 8: NMSE as a function of the memory length, M. The nonlinear order for PH [9] is fixed by 9. The structures of all NN-based DPD schemes are the same as in TABLE II and Fig. 7, except that D_1 increases as M increases. Specifically, the markers for $M = 3$ correspond to the results in TABLE II.

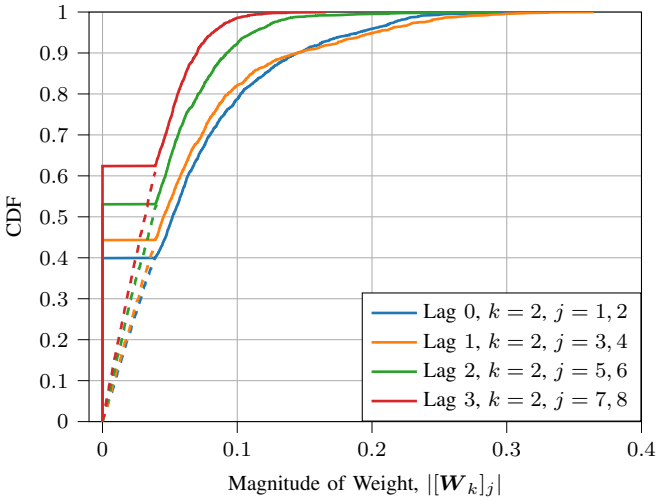

Fig. 9: CDFs of the weight magnitudes for neurons in the input layer of SVDEN fed with input signals with lag 0, 1, 2, and 3 before and after pruning (dashed and solid lines), respectively. $K = 3$, $D_2 = 512$, and $\eta_d = 0.5$. The final remaining weights have a minimal magnitude around 0.04.

with around 2 dB, than for other DPD schemes. We note that the performances of both RVTDNN and RVFTDNN start to degrade for $M \geq 4$, while this degradation is not noticed for shortcut connection networks, SVDEN and the R2TDNN. The degradation can be explained, by the fact that more irrelevant memory inputs are sent into NNs as M increases, which actually harms the NN accuracy.

3) Interpretation of Pruning: Considering SVDEN with $K = 3$, $M = 3$, $D_2 = 512$, and $\eta_d = 0.5$, Fig. 9 shows the cumulative distribution functions (CDFs) of the magnitude of weights in the connections between the first and second layers, i.e., $|\boldsymbol{W}_2|$, before and after pruning. Each CDF corresponds to the the magnitude of weights in the connections for every two neurons in the first layer fed with input signals of lag 0, 1, 2, and 3, i.e., $|[\boldsymbol{W}_2]_{1,2}|$, $|[\boldsymbol{W}_2]_{3,4}|$, $|[\boldsymbol{W}_2]_{5,6}|$, and $|[\boldsymbol{W}_2]_{7,8}|$, respectively. The dashed and solid lines correspond to before and after pruning, respectively. The remaining weights have a minimal magnitude around 0.04.

Note that the weights in the connections for neurons fed with shorter lag input signals have larger magnitudes (> 0.04) than for neurons fed with longer lag input signals, especially for lag 0. Thus, despite the pruning factor $\eta_d = 0.5$ for the second layer, more weights ($> 50\%$) are masked to zero for neurons with longer lags than for neurons of shorter lags ($< 50\%$). This indicates that SVDEN learns to assign more large valued weights to its memoryless input.

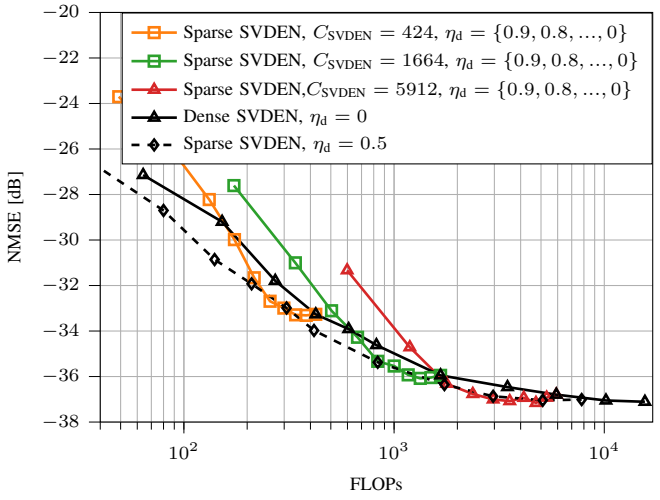

Fig. 10: NMSE as a function of the number of FLOPs for dense and sparse SVDENs with fixed $\eta_d = \{0, 0.5\}$ and varied $\eta_d = \{0.9, 0.8, ..., 0\}$. The markers for dense and sparse SVDENs correspond to different number of neurons in the hidden layers as in Fig. 5 and different η_d, respectively.

4) Large Sparse versus Small Dense: The performance of sparse and dense SVDENs is compared in Fig. 10. It illustrates the NMSE as a function of the number of FLOPs for three sparse SVDENs with a varied $\eta_d = \{0.9, 0.8, ..., 0\}$ but different non-pruned complexity $C_{\text{SVDEN}} = \{424, 1664, 5912\}$, and two SVDENs with fixed $\eta_d = \{0, 0.5\}$ from Fig. 5. All the SVDENs have the same number of layers $K = 5$.

For a given number of FLOPs, sparse SVDENs allow to outperform the dense SVDENs. A larger size SVDEN ($C_{\text{SVDEN}} = 5912$) allows a larger $\eta_d = 0.7$ than that of a smaller size SVDEN ($C_{\text{SVDEN}} = 424$) with $\eta_d = 0.5$. This indicates that there is an optimal pruning factor for a given sized SVDEN. The SVDEN with a fixed $\eta_d = 0.5$ performs nearly always the best over all complexities, which suggests that it is better to train a $2\times$ larger size dense SVDEN and prune it to the desired complexity than using the best dense SVDEN.

VI. Conclusion

We proposed a novel shortcut connection NN, referred to as SVDEN, for low-complexity mitigation of multiple hardware impairments in direct conversion transmitters. SVDEN keeps the instantaneous linear input-output relation of the transmitter by adding two trainable neuron-wise shortcut connections

between the corresponding neurons of the input and output layers. Furthermore, we proposed and analyzed a NN connection pruning algorithm, which allows to gradually remove weights of minimum magnitude in each layer. Experimental results show that SVDEN with a pruning factor of 0.5 achieves a NMSE gain > 2.5 dB and an ACPR gain > 2 dBc compared to other RVTDNN-based models and a Volterra-based model proposed in the literature, with less or similar complexity.

APPENDIX A
COMPUTATION COMPLEXITY OF THE PH

The extended PH [9] is based on the PH model [11] given by the polynomials

$$\psi_p(x(n)) = \sum_{k \in I_p} a_{k,p} |x(n)|^{k-1} x(n), p \in I_p, \qquad (28)$$

where p is the polynomial order, $I_p = \{1, 3, ..., p\}$ for only odd orders, and $a_{k,p}$ are the polynomial weights. The polynomial (28) and its conjugate $\psi_p(x^*(n))$ are filtered by FIR filters $h_p(n)$ and $h_q(n)$ of length L_p and L_q, respectively. The output of the extended PH is

$$y(n) = \sum_{p=1}^{P} h_p(n) \circledast \psi_p(x^*(n)) + \sum_{q=1}^{Q} h_q(n) \circledast \psi_q(x^*(n)), \qquad (29)$$

where P and Q are the polynomial orders for the non-conjugate and conjugate branches and $\circledast$ denotes convolution.

The number of complex-valued weights in (28) is

$$N_{\text{PH, poly}} = \left(1 + \frac{P+1}{2}\right) \frac{P+1}{4} + \left(1 + \frac{Q+1}{2}\right) \frac{Q+1}{4}. \qquad (30)$$

8 FLOPs are required for each weight, where 6 FLOPs are for the complex multiplication and 2 FLOPs for the complex summation [14]. Similarly, the number of complex-valued filter parameters is [9]

$$N_{\text{PH, filter}} = \sum_{p \in I_P} L_p + \sum_{q \in I_Q} L_q + 1, \qquad (31)$$

which also require 8 FLOPs each. In total, the number of FLOPs required for the PH is

$$C_{\text{PH}} = 8(N_{\text{PH, poly}} + N_{\text{PH, filter}}) - 4 + 3 + (\max(P, Q) - 1). \qquad (32)$$

REFERENCES

[1] Y. Wu, U. Gustavsson, A. Graell i Amat, and H. Wymeersch, "Residual neural networks for digital predistortion," in *Proc. IEEE GLOBECOM'20*, Dec. 2020, pp. 1–6.

[2] A. A. Abidi, "Direct-conversion radio transceivers for digital communications," *IEEE Journal of solid-state circuits*, vol. 30, no. 12, pp. 1399–1410, Dec. 1995.

[3] U. Gustavsson, C. Sanchéz-Perez, T. Eriksson, F. Athley, G. Durisi, P. Landin, K. Hausmair, C. Fager, and L. Svensson, "On the impact of hardware impairments on massive MIMO," in *IEEE GLOBECOM Workshops*, Dec. 2014, pp. 294–300.

[4] S. C. Cripps, *RF power amplifiers for wireless communications*. Artech house Norwood, MA, 2006, vol. 2.

[5] J. C. Pedro and N. B. Carvalho, *Intermodulation distortion in microwave and wireless circuits*. Artech House, 2003.

[6] H. Ku and J. S. Kenney, "Behavioral modeling of nonlinear RF power amplifiers considering memory effects," *IEEE Trans. Microw. Theory Tech*, vol. 51, no. 12, pp. 2495–2504, Dec. 2003.

[7] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electron. Lett.*, vol. 37, no. 23, pp. 1417–1418, Nov. 2001.

[8] H. Cao, A. S. Tehrani, C. Fager, T. Eriksson, and H. Zirath, "I/Q imbalance compensation using a nonlinear modeling approach," *IEEE Trans. Microw. Theory Tech.*, vol. 57, no. 3, pp. 513–518, Mar. 2009.

[9] L. Anttila, P. Händel, and M. Valkama, "Joint mitigation of power amplifier and I/Q modulator impairments in broadband direct-conversion transmitters," *Trans. Microw. Theory Tech.*, vol. 58, no. 4, pp. 730–739, Mar. 2010.

[10] D. Wang, M. Aziz, M. Helaoui, and F. M. Ghannouchi, "Augmented real-valued time-delay neural network for compensation of distortions and impairments in wireless transmitters," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 30, no. 1, pp. 242–254, Jun. 2018.

[11] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

[12] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based Volterra behavioral modeling of RF power amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 12, pp. 4323–4332, Dec. 2006.

[13] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[14] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson, and C. Fager, "A comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 6, pp. 1510–1520, Jun. 2010.

[15] J. Moon and B. Kim, "Enhanced hammerstein behavioral model for broadband wireless transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 59, no. 4, pp. 924–933, Apr. 2011.

[16] S. Afsardoost, T. Eriksson, and C. Fager, "Digital predistortion using a vector-switched model," *IEEE Trans. Microw. Theory Tech.*, vol. 60, no. 4, pp. 1166–1174, Apr. 2012.

[17] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.

[18] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band dynamic modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[19] M. Rawat, K. Rawat, and F. M. Ghannouchi, "Adaptive digital predistortion of wireless power amplifiers/transmitters using dynamic real-valued focused time-delay line neural networks," *IEEE Trans. Microw. Theory Tech.*, vol. 58, no. 1, pp. 95–104, Jan. 2010.

[20] F. Mkadem and S. Boumaiza, "Physically inspired neural network model for RF power amplifier behavioral modeling and digital predistortion," *IEEE Trans. Microw. Theory Tech.*, vol. 59, no. 4, pp. 913–923, Apr. 2011.

[21] M. Rawat and F. M. Ghannouchi, "A mutual distortion and impairment compensator for wideband direct-conversion transmitters using neural networks," *IEEE Trans. Broadcast.*, vol. 58, no. 2, pp. 168–177, Jun. 2012.

[22] P. Jaraut, M. Rawat, and F. M. Ghannouchi, "Composite neural network digital predistortion model for joint mitigation of crosstalk, I/Q imbalance, nonlinearity in MIMO transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 66, no. 11, pp. 5011–5020, Nov. 2018.

[23] L. Anttila, M. Valkama, and M. Renfors, "Frequency-selective I/Q mismatch calibration of wideband direct-conversion transmitters," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 55, no. 4, pp. 359–363, Apr. 2008.

[24] L. Ding, Z. Ma, D. R. Morgan, M. Zierdt, and G. T. Zhou, "Compensation of frequency-dependent gain/phase imbalance in predistortion linearization systems," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 55, no. 1, pp. 390–397, Feb. 2008.

[25] G. M. Raz and B. D. Van Veen, "Baseband Volterra filters for implementing carrier based nonlinearities," *IEEE Trans. Signal Process.*, vol. 46, no. 1, pp. 103–114, Jan. 1998.

[26] C. Eun and E. J. Powers, "A new Volterra predistorter based on the indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 223–227, Jan. 1997.

[27] C. Tarver, A. Balatsoukas-Stimming, and J. R. Cavallaro, "Design and implementation of a neural network based predistorter for enhanced mobile broadband," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*, Oct. 2019, pp. 296–301.
[28] V. Bajaj, F. Buchali, M. Chagnon, S. Wahls, and V. Aref, "Single-channel 1.61 Tb/s optical coherent transmission enabled by neural network-based digital pre-distortion," *Proc. IEEE ECOC'20*, pp. Tu1D–5, Dec. 2020.
[29] D. Zhou and V. E. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *IEEE Trans. Signal Process.*, vol. 55, no. 1, pp. 120–133, Dec. 2006.
[30] M. Schetzen, "Theory of pth-order inverses of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. 23, no. 5, pp. 285–291, May. 1976.
[31] J. Chani-Cahuana, C. Fager, and T. Eriksson, "A new variant of the indirect learning architecture for the linearization of power amplifiers," in *European Microw. Conf. (EuMC)*, Sep. 2015, pp. 1295–1298.
[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR'16*, Jun. 2016, pp. 770–778.
[33] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *NIPS'15*, Dec. 2015.
[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE CVPR'15*, Jun. 2015, pp. 1–9.
[35] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *ICLR'16*, May. 2016.
[36] Y. Guo, A. Yao, and Y. Chen, "Dynamic network surgery for efficient DNNs," in *Adv. Neural Inform. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates Inc., 2016, pp. 1379–1387.
[37] M. Zhu and S. Gupta, "To prune, or not to prune: exploring the efficacy of pruning for model compression," *ICLR'18*, May. 2018.
[38] T. Gale, E. Elsen, and S. Hooker, "The state of sparsity in deep neural networks," *arXiv preprint arXiv:1902.09574*, 2019.
[39] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "EIE: efficient inference engine on compressed deep neural network," in *ACM/IEEE ISCA'16*, Jun. 2016, pp. 243–254.
[40] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. solid-state circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.
[41] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen, "Cambricon-X: An accelerator for sparse neural networks," in *IEEE/ACM MICRO'16*, Oct. 2016, pp. 20:1–20:12.
[42] S. Dey, K.-W. Huang, P. A. Beerel, and K. M. Chugg, "Pre-defined sparse neural networks with hardware acceleration," *IEEE J. Emerg. Sel. Topics Circuits and Syst.*, vol. 9, no. 2, pp. 332–345, Mar. 2019.
[43] P. N. Landin, S. Gustafsson, C. Fager, and T. Eriksson, "WebLab: A web-based setup for PA digital predistortion and characterization [application notes]," *IEEE Microw. Mag.*, vol. 16, no. 1, pp. 138–140, Feb. 2015.
[44] Z. Zhu, X. Huang, and H. Leung, "Joint I/Q mismatch and distortion compensation in direct conversion transmitters," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2941–2951, May. 2013.
[45] J. Chani-Cahuana, C. Fager, and T. Eriksson, "Lower bound for the normalized mean square error in power amplifier linearization," *IEEE Microw. Wireless Compon. Lett.*, vol. 28, no. 5, pp. 425–427, May. 2018.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR'15*, May. 2015.

Yibo Wu (S'20) obtained the Bachelor's degree in Communication Engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017, and the M.S.c degree in Electrical Engineering from Chalmers University of Technology, Gothenburg, Sweden, in 2019, where he is currently pursuing the Ph.D. degree in the Department of Electrical Engineering as an Industrial Ph.D. Candidate. He joined Ericsson AB in 2020. His main research interests lie in the hardware impairments mitigation in the communication system, with an emphasis on the usage of deep learning techniques.

Ulf Gustavsson received the M.Sc. degree in electrical engineering from Örebro University, Örebro, Sweden, in 2006, and the Ph.D. degree from Chalmers University of Technology, Gothenburg, Sweden, in 2011. His background ranges from power amplifier design to radio signal processing. He is currently a Senior Specialist with Ericsson Research where his research interests include novel radio signal processing techniques for hardware impairment mitigation and behavioral modeling of radio hardware for future advanced antenna systems and communication engineering. He is currently the Lead Scientist from Ericsson Research with the Marie Skodowska-Curie European Industrial Doctorate Innovative Training Net-work, SILIKA.

Alexandre Graell i Amat (S'01–M'05–SM'10) is a Professor at the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. He received the M.Sc. degree in Telecommunications Engineering from the Universitat Politècnica de Catalunya, Barcelona, Catalonia, Spain, in 2001, and the M.Sc. and Ph.D. degrees in Electrical Engineering from the Politecnico di Torino, Turin, Italy, in 2000 and 2004, respectively. From 2001 to 2002, he was a Visiting Scholar with the University of California San Diego, La Jolla, CA, USA. From 2002 to 2003, he held a visiting appointment at Universitat Pompeu Fabra, Barcelona, and the Telecommunications Technological Center of Catalonia, Barcelona. From 2001 to 2004, he held a part-time appointment at the STMicroelectronics Data Storage Division, Milan, Italy, as a consultant on coding for magnetic recording channels. From 2004 to 2005, he was a Visiting Professor with Universitat Pompeu Fabra, Barcelona. From 2006 to 2010, he was with the Department of Electronics, IMT Atlantique (formerly ENST Bretagne), Brest, France. Since 2019 he is also Adjunct Research Scientist at Simula UiB, Bergen, Norway. His research interests are in the field of coding theory with application to distributed computing, privacy and security, random access, and optical communications.

Prof. Graell i Amat received the Marie Skłodowska-Curie Fellowship from the European Commission and the Juan de la Cierva Fellowship from the Spanish Ministry of Education and Science. He received the IEEE Communications Society 2010 Europe, Middle East, and Africa Region Outstanding Young Researcher Award. He was the General Co-Chair of the 7th International Symposium on Turbo Codes and Iterative Information Processing, Sweden, 2012, and the TPC Co-Chair of the 11th International Symposium on Topics in Coding, Canada, 2021. He was an Associate Editor of the IEEE COMMUNICATIONS LETTERS from 2011 to 2013. He was Associate Editor and Editor-at-Large of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2011 to 2016 and 2017 to 2020, respectively. He is currently Area Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS.

Henk Wymeersch (S'01, M'05, SM'19) obtained the Ph.D. degree in Electrical Engineering/Applied Sciences in 2005 from Ghent University, Belgium. He is currently a Professor of Communication Systems with the Department of Electrical Engineering at Chalmers University of Technology, Sweden. He is also a Distinguished Research Associate with Eindhoven University of Technology. Prior to joining Chalmers, he was a postdoctoral researcher from 2005 until 2009 with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Prof. Wymeersch served as Associate Editor for IEEE Communication Letters (2009-2013), IEEE Transactions on Wireless Communications (since 2013), and IEEE Transactions on Communications (2016-2018). During 2019-2021, he is an IEEE Distinguished Lecturer with the Vehicular Technology Society. His current research interests include the convergence of communication and sensing, in a 5G and Beyond 5G context.