Deep Neural Network-Based Digital Pre-distortion for High Baudrate Optical Coherent Transmission

Bajaj, Vinod; Buchali, Fred; Chagnon, Mathieu; Wahls, Sander; Aref, Vahid

DOI
10.1109/JLT.2021.3122161

Publication date
2022

Document Version
Final published version

Published in
Journal of Lightwave Technology

Deep Neural Network-Based Digital Pre-Distortion for High Baudrate Optical Coherent Transmission

Vinod Bajaj, Fred Buchali, Mathieu Chagnon, *Member, IEEE*, Sander Wahls, *Senior Member, IEEE*, and Vahid Aref

(Top-Scored Paper)

***Abstract*—High-symbol-rate coherent optical transceivers suffer more from the critical responses of transceiver components at high frequency, especially when applying a higher order modulation format. We recently proposed a neural network (NN)-based digital pre-distortion (DPD) technique trained to mitigate the transceiver response of a 128 GBaud optical coherent transmission system. In this paper, we further detail this work and assess the NN-based DPD by training it using either a direct learning architecture (DLA) or an indirect learning architecture (ILA), and compare performance against a Volterra series-based ILA DPD and a linear DPD. Furthermore, we deliberately increase the transmitter nonlinearity and compare the performance of the three DPDs schemes. The proposed NN-based DPD trained using DLA performs the best among the three contenders. In comparison to a linear DPD, it provides more than 1 dB signal-to-noise ratio (SNR) gains at the output of a conventional coherent receiver DSP for uniform 64-quadrature amplitude modulation (QAM) and PCS-256-QAM signals. Finally, the NN-based DPD enables achieving a record 1.61 Tb/s net rate transmission on a single channel after 80 km of standard single mode fiber (SSMF).**

***Index Terms*—Artificial neural networks, digital pre-distortion, digital signal processing, machine learning and optical fiber communication.**

I. Introduction

THE exponential increase in the internet traffic due to the emergence of bandwidth-hungry services such as cloud-based applications and video on demand is pushing the existing optical transport network to its limit. To increase the aggregate bit rate carried by a singlefiber strand, one must find ways to best utilize the available optical spectrum while minimizing the number of components required to do so. The three main avenues to attain this objective are to increase the symbol rate and the average number of bits conveyed per symbol on a carrier, and decrease the spectral guard band between multiplexed carriers. Thus, it is desirable to operate such systems at high symbol rates on a tight spectral grid using high-order modulation formats to maximize the information rate [1]–[4]. Therefore, the signal-to-noise ratio (SNR) should be as high as possible, a necessary condition for operating on a high order format.

The generation of signals with high integrity is challenging due to the impairments stemming from different sources along the information transmission system. At the transmitter side, these impairments include the limited bandwidth and the nonlinear characteristics of components. A common practice to mitigate these distortions is by digitally pre-compensating them in the digital signal processing (DSP) stack, a technique usually termed "digital pre-distortion" (DPD).

A linear DPD is generally employed to compensate for linear inter-symbol interference stemming from the limited bandwidth and/or the imperfect spectral response of the transmitter components [5], [6]. It is a common practice to limit the amplitude of the signals applied to transmitter components exhibiting a nonlinear response (e.g. driver amplifier (DA), electro-optic modulator, etc.) when using a linear DPD, which in turn limits the SNR because of the small signal power. A larger signal swing can improve the SNR, but may require to be accompanied by a nonlinear DPD to pre-compensate the increased nonlinear distortions. To increase the information rate, transmitters will require DPDs that can compensate for both the linear and nonlinear responses. The most common nonlinear DPDs are based on Volterra series which have been investigated for both radio frequency (RF) amplifiers [7]–[11] and coherent optical transmitters [12]–[17].

Another type of DPD is based on neural networks (NNs) whose application dates back to 1980s [18], [19]. Recently, NN-based DPDs have received more attention [20]–[27]. A simple feed-forward NN (FFNN) was used in [22] to mitigate the response of RF amplifier, without considering any memory effects. The memory effects were included in the DPDs based on time-delay NNs (TDNNs) [20], [25], [28] and on convolutional NNs (CNNs) [26]. Recently, some of the above schemes have been compared in [21] and shown experimentally that

Manuscript received February 26, 2021; revised July 1, 2021, August 31, 2021, and October 9, 2021; accepted October 13, 2021. Date of publication October 26, 2021; date of current version February 1, 2022. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant 766115. *(Corresponding author: Vinod Bajaj.)*

Mathieu Chagnon is with Nokia Bell Labs, Stuttgart 70435, Germany (e-mail: mathieu.chagnon@nokia-bell-labs.com).

Vinod Bajaj is with Nokia Bell Labs, 70435 Stuttgart, Germany, and also with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: v.bajaj-1@tudelft.nl).

Sander Wahls is with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: s.wahls@tudelft.nl).

Fred Buchali and Vahid Aref are with the Nokia Solutions and Networks GmbH und Co KG, 70435 Stuttgart, Germany (e-mail: fred.buchali@nokia.com; vahid.aref@nokia.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2021.3122161.

Digital Object Identifier 10.1109/JLT.2021.3122161

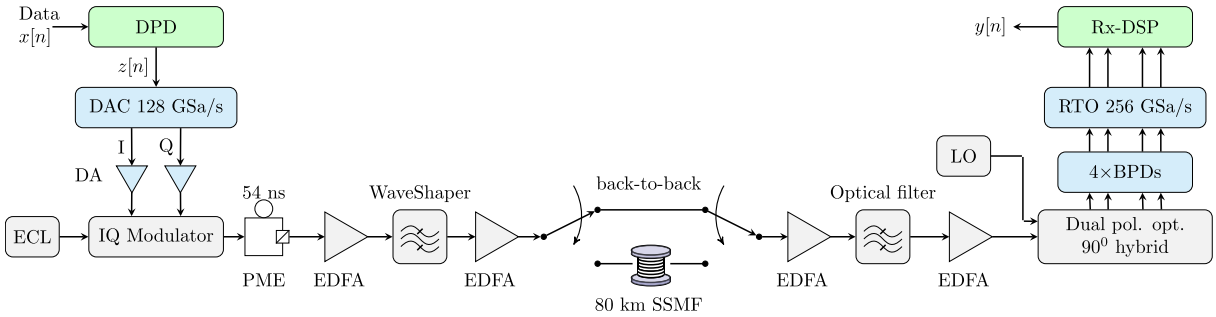

Fig. 1. A schematic of the 128 GBaud experimental setup configured either in back-to-back or in an 80 km fiber transmission arrangement.

adding residual neural network (ResNet) structure improves the nonlinearity mitigation of RF amplifiers. However, NN-based DPDs for optical coherent transmitters are so far not well explored. Memoryless FFNNs were proposed to pre-compensate for Mach-Zehnder modulator (MZM) responses [29] and a simulated low-resolution digital to analog converter (DAC) response [24]. In [27], a DPD based on recurrent NNs (RNNs) is applied to the simulated aggregate response of a optical coherent transmitter. In another paper [30], a NN-based DPD was designed by considering the collective response of a coherent transmitter as a Wiener-Hammerstein (WH) model.

Very recently, we proposed an NN-based DPD designed using simple FFNNs and CNNs for a high-baud rate (12 GBaud) coherent optical transmitter [31]. In this paper, we demonstrate that the considered NN-based DPD leads to a record 1.61Tb/s data rate over a 80km fiber link, detail the proposed NN-based DPD and further investigate its performance by training it using the two well-known learning architectures, namely the indirect learning architecture (ILA) [7], [18] and the direct learning architecture (DLA) [8], [9]. In addition, we also consider a Volterra series-based ILA DPD and a linear DPD in our investigation. All considered DPDs are disjoint-IQ implemented using real values i.e. two separate DPDs each for I and Q channel. The DPDs are applied to two modulation formats, namely 64-quadrature amplitude modulation (QAM) and probabilistic constellation shaped (PCS)-256-QAM, and trained and evaluated for varying transmitter nonlinearity. Our results show that NN-based DPD trained using DLA performs the best among the considered candidates and obtains gains of 1.6 dB and 1.2 dB in received SNR with respect to the linear DPD for uniform 64-QAM and PCS-256-QAM, respectively. In addition, we compare the complexity of DPDs considered in the study. We reduce the complexity of the proposed NN-based DPD by applying a pruning method.

The outline of the paper is as follows: in Section II the experimental setup is described. The DPD techniques considered in this work and their implementation are discussed in Section III. The proposed NN-based DPD is detailed in Section IV, and performance assessments are presented in Section V. A complexity comparison is added to Section VI. The paper is concluded in Section VII,

II. High Baud Rate Coherent Optical Transmitter

A schematic of our experimental setup is shown in Fig. 1. The transmitter (Tx) comprises three major components: digital to analog converters, driver amplifiers and an optical modulator (external IQ modulator). Each of the components have linear and nonlinear characteristics. In addition, the signal reflections originating by the radio frequency (RF) cables/connections add to the linear effects and further spread the impulse response. Overall, analog signals leaving each DAC flow through a chain of linear and nonlinear responses.

We used uniform 64-QAM and probabilistic constellation shaped 256-QAM (PCS-256-QAM) signals. The PCS 256-QAM format was shaped using the Maxwell-Boltzmann distribution with an entropy of 7.5 bits/symbol which was previously determined as a good choice for 19dB SNR, the SNR limit in the setup. In the transmitter, the data consisting of a 2^{15} symbol sequence passes through the DPD block. The predistorted sequence is then loaded into the DACs after clipping and quantization. The DACs sample the signal at 128 GSa/s and operate at 1 sample per symbol (sps). The DACs have an effective number of bits (ENOB) of 4 at 64 GHz and 24 GHz 3-dB bandwidth. The two DACs produce the two electrical tributaries feeding a single-polarization optical IQ modulator. The outputs of the DACs are first amplified using DAs with 60 GHz 3-dB bandwidth. The DAs' outputs are then fed to the lithium-niobate ($LiNbO_3$) IQ modulator which has around 41 GHz 3-dB bandwidth. The optical carrier is generated by an external cavity laser (ECL) at 193.5 THz with a line-width of $<$100 kHz. The optical carrier is fed into the IQ modulator, where it gets modulated by the signals from the DAs' outputs. A polarization multiplexing emulator (PME) with a decorrelation delay of 54 ns is used to generate a polarization multiplexed signal. The polarization multiplexed signal is amplified using an Erbium doped fiber amplifier (EDFA). A Finisar Waveshaper is used to compensate the low pass response of the IQ modulator by increasingly attenuating frequencies closer to the carrier in order to flatten the optical spectrum at its output. The WaveShaper is configured once when a linear DPD was employed at the transmitter and kept fixed. The signal is then amplified using another EDFA and either sent directly to the coherent receiver or transmitted through 80 km of standard single mode fiber (SSMF) before coherent reception. The coherent receiver is preceded by an EDFA, an optical filter of 128 GHz 3-dB bandwidth to remove the amplified spontaneous emission noise, and a second EDFA. The resulting optical signal beats with a local oscillator through a dual-polarization 90° hybrid. Four balanced photo-diodes (BPDs) detects the signal. A Keysight high bandwidth real time oscilloscope (RTO) is used to sample

and record the four detected waveform at 256 GSa/s. The RTO has a nominal resolution of 10 bits.

An offline DSP for symbol recovery after optical coherent detection is applied. Note that, we intend to compensate the transmitter impairments using a DPD. While it is difficult to isolate the transmitter and the receiver impairments in an experimental setup, methods like homodyne detection to mitigate certain receiver impairments can be used. Also, some of the receiver impairments such as low pass response of photo-detectors can be separately determined and compensated by employing static filters. However, such scenarios are challenging for integrated transceivers. To train the DPD parameters in the transmitter DSP, we chose to always apply the same DSP stages at the receiver. Applying coherent receiver DSP allows to convert the inherently dynamic channel response (e.g.: rotation of the state of polarization followed by a polarization beam splitter at the receiver, beating of the incoming optical signal with a free running, non-phase locked laser source) into a stationary channel response as also applied in previous DPD research works [12]–[14].

The receiver DSP first re-samples the signal at 2 samples per symbol. Then chromatic dispersion is removed and timing errors are corrected. The polarizations of the signal are de-multiplexed by using a 2×2 complex-valued multiple-input multiple-output (MIMO) equalizer updated by a multi-modulus algorithm (MMA). Intermediate frequency offset and phase noise are then compensated. The residual signal distortions are compensated by another MIMO equalizer, operated as 4×4 on real values. Note that some of the impairments that originate at the transmitter may get corrected by this adaptive 4×4 real-valued MIMO equalizer used in the receiver DSP. Consequently, the transmitter may not compensate for some of the impairments that actually occurred at the transmitter because they are always handled by the receiver DSP. As transmitter pre-distortion is the main focus of this work, we try to compensate most of the impairments at the transmitter. There could be different ways to do this, such as training the DPDs by excluding the 4×4 real-valued MIMO equalizer from the training loop [32]. In this work, the DPDs are trained in a step by step procedure by changing the length of the receiver 4×4 real-valued MIMO equalizer.

The transmission quality is measured in terms of SNR, mutual information (MI) and generalized MI (GMI). For decoding, we used a family of 130 optimized spatially coupled LDPC codes [1], [33] with variable overheads ranging from 3% to 100%. For each channel, the code with the smallest overhead capable of decoding the bits error-free is chosen.

III. Review of Various DPD Techniques

In our work, we evaluated the performance of a linear DPD as well as nonlinear DPDs based on either Volterra series or the proposed neural network architecture. In the following, we first describe the two general training methods well-known in literature namely direct learning architecture (DLA) and indirect learning architecture (ILA). Then, we briefly review the Volterra series-based DPD. The proposed NN-based DPD is described in the next section.

A. Direct Vs. Indirect Learning Architecture

An example schematic of the DLA is shown in Fig. 2(a). In DLA, the "communication channel" is modelled by a differentiable auxiliary channel model S with the help of which the DPD is determined in the following two steps. In the first step, the auxiliary channel model is trained by minimizing the objective function $J_1 = \Sigma_n \frac{1}{2}(e_1[n])^2 = \Sigma_n \frac{1}{2}(y[n] - y_e[n])^2$. Here, $y[n]$ is the soft symbols output of the receiver DSP and $y_e[n]$ is the output of the auxiliary channel model. Both sequences $y[n]$ and $y_e[n]$ are corresponding to a sequence $z[n]$ injected into both the communication channel and the auxiliary channel model. In the second step, once the auxiliary channel model S cannot further minimize the objective J_1, S is fixed to its current state and only the DPD G is updated to minimize $J_2 = \Sigma_n \frac{1}{2}(e_2[n])^2 = \Sigma_n \frac{1}{2}(x[n] - y_e[n])^2$. The gradients of the loss function are back-propagated through S in order to train G; the parameterized digital pre-distortion function. The input to the DPD is $x[n]$ which consists of ideal QAM symbols. The DPD obtained using the second step changes the statistics of the input signal $z[n]$; consequently, changing the response of the communication channel. So, the auxiliary channel model needs to be retrained. Thus, using the DLA architecture, the parametrized functions S and G are iteratively trained until no more gains are obtained.

In contrasts to DLA, the ILA architecture does not require an auxiliary channel model, as is shown in Fig. 2(b). With ILA, the DPD G is trained at the output of the communication channel as a post-equalizer, while a copy of the pre-distorter G (obtained from the previous iteration) is used at the input of the transmitter. The new DPD G is trained by minimizing the objective function $J_1[n] = (e_1[n])^2 = \Sigma_n \frac{1}{2}(z[n] - z_e[n])^2$. The input to the DPD (training block) is the soft symbols output of the Rx-DSP $y[n]$. The "communication channel" response changes with signal statistics, hence, several iterations are needed to achieve good convergence. As ILA does not require an auxiliary model, its computational complexity in the training phase is almost halved compared to DLA. However, it suffers from a bias caused by nonlinear operations of DPD on the transmitter output which is often noisy, as explained in [34]. Further, the nonlinear DPD trained using ILA may not be the optimum as nonlinear blocks may not be commutative.

B. Linear DPD

The output of a M_1 memory linear DPD $z_e[n]$ for inputs $y[n]$ is given by

$$z_e[n] = \sum_{\tau_1=-M_1}^{M_1} g[\tau_1] y[n-\tau_1], \tag{1}$$

where $g[\tau_1]$ are filter coefficients. The above relation can be written as

$$\vec{z}_e = \boldsymbol{Y}\vec{g}, \tag{2}$$

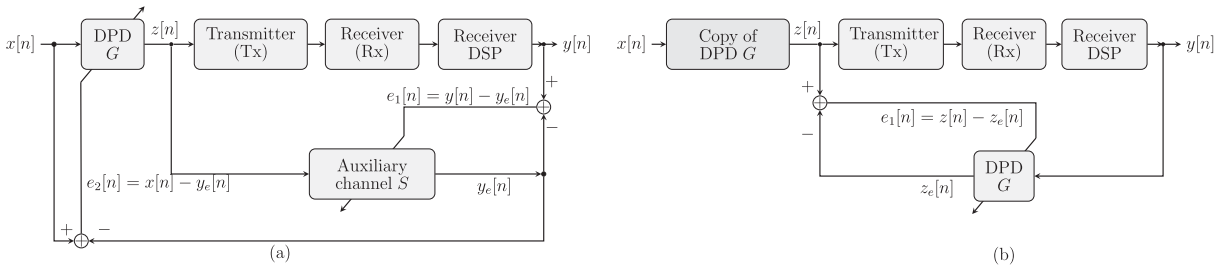

Fig. 2. An example representation of training a DPD G using (a) direct learning architecture (DLA) and (b) indirect learning architecture (ILA). S is the auxiliary channel model.

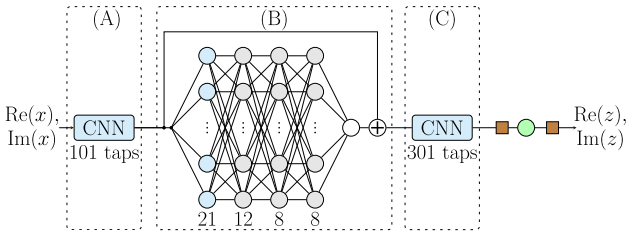

Fig. 3. Architecture of the proposed NN-based DPD.

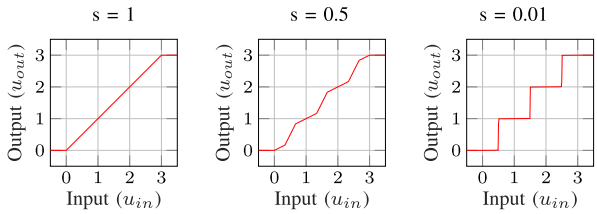

Fig. 4. A 2-bit Soft-DAC activation unit for different softening factor (s).

where $\vec{(.)}$ denotes vector quantities and $\vec{g}$ is the vector of the linear filter coefficients $g[\tau_1]$. The matrix $\boldsymbol{Y}$ is made of columns of shifted vectors $\vec{y}_k = [y[k], y[k+1], \ldots, y[k+n]]^T$. Here, $(.)^T$ denotes the transpose operation. The matrix $\boldsymbol{Y}$ is represented mathematically as $\boldsymbol{Y} = [\vec{y}_{-M_1} \ldots \vec{y}_{+M_1}]$. The coefficients vector $\vec{g}$ can be obtained using the Moore-Penrose inverse by the relation

$$\vec{g} = \left(\boldsymbol{Y}^H \boldsymbol{Y}\right)^{-1} \boldsymbol{Y}^H \vec{z_e} \tag{3}$$

where $(.)^H$ represents complex-conjugate transpose and $\vec{z_e} = [z_e[0], z_e[1] \ldots, z_e[n]]^T$. Two separate disjoint linear DPD models were determined for I and Q tributaries.

C. Volterra Series-Based DPD

Volterra series are well-known to model nonlinear systems with memory [35]. The output $z_e[n]$ of a nonlinear system up to a nonlinearity order of three with memory can be written in terms of its input $y[n]$ in the form of

$$z_e[n] = g_0 + \sum_{\tau_1=-M_1}^{M_1} g_1[\tau_1] y[n-\tau_1] + \tag{4}$$

$$\sum_{\tau_2=-M_2}^{M_2} \sum_{d_2=0}^{D_2} g_2[\tau_2, d_2] y[n-\tau_2] y[n-\tau_2-d_2] + \tag{5}$$

$$\sum_{\tau_3=-M_3}^{M_3} \sum_{d_2=0}^{D_2} \sum_{d_3=d_2}^{d_2+D_3} g_3[\tau_3, d_2, d_3]. \tag{6}$$

$$y[n-\tau_3] y[n-\tau_3-d_2] y[n-\tau_3-d_3] \tag{7}$$

where M_p is the memory length and g_p are Volterra kernel coefficients in the p^{th} order. We considered a Volterra series-based DPD trained using ILA. The Volterra kernel (pre-distorter) coefficients g_p that map y to z_e can be represented in the matrix form like (2). Here, $\vec{g}$ is the vector of the Volterra kernel coefficients g_p and the matrix $\boldsymbol{Y}$ is made of columns of shifted vectors $\vec{y}_k = [y[k], y[k+1], \ldots, y[k+n]]^T$ and columns generated by element-wise multiplications of shifted versions of $\vec{y}$. The input $y[n]$ is normalized to have a unit variance prior to generating the matrix $\boldsymbol{Y}$ is represented mathematically as $\boldsymbol{Y} = [\vec{y}_{-M_1} \ldots \vec{y}_{+M_1} \ldots \vec{y} \odot \vec{y} \ldots \vec{y}_{M_2} \odot \vec{y}_{M_2-D_1} \ldots]$. In the above relation, the element-wise multiplication operation is denoted by $\odot$. The coefficients g_p can be obtained by using the Moore-Penrose inverse with relation (3). In this paper, we do not consider IQ cross-talk compensation at the transmitter, hence, two separate disjoint Volterra-series based DPD models were determined each for I and Q tributaries which were implemented using real values.

IV. Neural Network-Based DPD

First, we modelled the optical coherent transmitter using experimentally acquired linear responses of the DACs and the DAs and simulated nonlinear responses of DACs, DAs and MZMs. Then, we tested different architectures in numerical simulations. The linear memory was accounted using convolutional neural networks (CNNs) which are easy to interpret. A total memory of around 400 taps was needed for the CNNs due the signal reflections with a large time delay present in the experimental setup. These reflections are shown in Fig. 5 and explained in the next section.

To account for nonlinearity, we used fully connected layers with leaky ReLU activation functions which can be evaluated with a few simple operations. Nonlinearity mixed with memory was introduced in the NN by adding a convolution layer before the fully connected layers. A sufficient number of layers and neurons were then determined over the simulation setup by different trials and by observing the performance (in terms of SNR) of a given architecture. It was known that DACs have

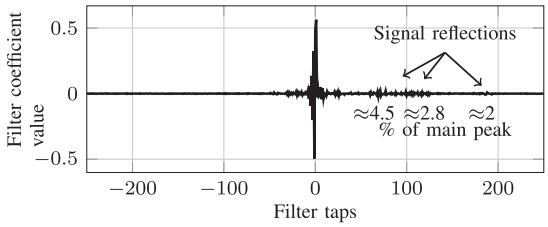

Fig. 5. The experimentally obtained coefficients of the "I" tributary Linear DPD filter. A filter with long memory is required to capture the reflections of the signal delayed by ≈ 200 symbol periods.

strong low pass response (8dB attenuation at 64 GHz [2]) and a non-negligible degradation of the signal quality at the transmitter stems from the power amplifiers nonlinearity. Hence, we design an NN-based DPD architecture by inspiring from the so-called Wiener-Hammerstein (WH) structure [35] i.e. by keeping the nonlinear fully connected FFNN layers in between the two linear memory (CNNs). This architecture gave better performance than others. Note that, although the structure of the proposed DPD is similar to a WH system, it does not belong to this class. The reason is that the nonlinear part is not static, but has memory. We found that as the linear effects are more dominant, an additional shortcut bypass (ResNet connection [36]) to the nonlinear FFNN improves the performance and speeds up the training process. Later, the number of layers and neurons was further reduced over the experimental setup to reach a final architecture described as follows.

The architecture of the DPD is shown in Fig. 3. The Sections (A) and (C) are uni-dimensional (1-D) linear CNNs, which are equivalent to finite impulse response (FIR) filters that compensate the linear responses while Section (B) in the middle mainly corrects for the nonlinear responses. In Section (B), the first layer (in cyan color) consists of short 1-D linear convolutions of 11 taps feeding to a layer with 21 neurons. The following three layers are fully connected FFNN layers with leaky rectified linear unit (Leaky ReLU) activation functions. The last layer in the FFNN has a single linear unit. The size of each layer is detailed in Fig. 3 and in Table I.

The NNs were implemented using real values. The complex signal was processed separately using two disjoint NNs. We used both DLA and ILA based training for the NN-based DPD and refer them as "NNDLA" and "NNILA" in the rest of the paper, respectively. For NNDLA based training, we used another NN serving as an auxiliary channel model S. Its architecture was designed as a mirrored version "(C)→(B)→(A)" of the DPD architecture shown in Fig. 3.

In order to model the DAC in our NN-based DPD, an approximation of the DAC was used to avoid the vanishing gradient problem. We call this customized unit as Soft-DAC and describe it in the following subsection.

A. Soft-DAC Activation Unit

Soft-DAC unit models the DAC with resolution of m bits and should quantize its input uniformly to 2^m discrete levels. As activation functions in NNs should have a non-zero derivative to pass gradients back, using a staircase activation function is not possible as its derivative is zero everywhere.

TABLE I
TABLE OF NETWORK SIZE HYPERPARAMETERS OF THE NN-BASED DPD

DPD-NN layer	Weight	Bias	Weight initialization	Bias initialization	Activation function
First CNN layer	101	1	Impulse	Zero	-
FFNN layer 1 (CNN)	11×21 = 231	21	Impulse	Zeros	-
FFNN layer 2	21×12 = 252	12	Kaiming uniform	Kaiming uniform	Leaky ReLU (0.1)
FFNN layer 3	12×8 = 96	8	Kaiming uniform	Kaiming uniform	Leaky ReLU (0.1)
FFNN layer 4	8×8 = 64	8	Kaiming uniform	Kaiming uniform	Leaky ReLU (0.1)
FFNN layer 5	8×1 = 8	1	Kaiming uniform	Kaiming uniform	-
Last CNN layer	301	1	Impulse	Zero	-
BN layer 1	1	1	One	Zero	-
BN layer 2	1	1	One	Zero	-
Auxiliary-NN layer	**Weight**	**Bias**	**Weight initialization**	**Bias initialization**	**Activation function**
First CNN layer	301	1	Impulse	Zero	-
Last CNN layer	101	1	Impulse	Zero	-

The output of the Soft-DAC unit, u_{out}, is defined by

$$f(u_{\text{in}};s)=\begin{cases}\lfloor u_{\text{in}}\rfloor+sr & r\le th\\ \lfloor u_{\text{in}}\rfloor+0.5+(r-0.5)/s & th<r<1-th\\ \lfloor u_{\text{in}}\rfloor+1+s(r-1) & r\ge 1-th,\end{cases}$$

$$u_{\text{out}}=\max\{\min\{f(u_{\text{in}};s),0\},2^m-1\},$$

where u_{in} is the input, $\lfloor.\rfloor$ is the floor function, $r=u_{\text{in}}-\lfloor u_{\text{in}}\rfloor$, s is the softening factor and $th=0.5/(1+s)$. The Soft-DAC unit is implemented as a piece-wise linear function with $2^{m+1}+1$ linear pieces. The slope of pieces is alternatively s and $1/s$. The behavior of the Soft-DAC can be changed from clipping-only operation to clipping and $m-$bits quantization operation by varying the slope s from 1 to 0, as shown in Fig. 4. The input to the Soft-DAC should be scaled and shifted properly such that it fits around the range of 0 to 2^m-1. A batch normalization (BN) layer is used prior to the Soft-DAC. The scaling parameter in the BN layer optimizes the clipping and is optimized by manually decreasing s from 1 towards 0 during the training of the pre-distorter. Finally, when $s=0$ the outputs of Soft-DAC are discrete levels and any preceding NN layers to the Soft-DAC cannot be trained.

The Soft-DAC unit is one way to apply quantization in the NN framework. There are alternative ways available in the literature to implement quantization in the NN framework. We refer to [37] and references there in the paper. The problem at hand is not the quantization of the weights (NN model parameters) but the activations. Some of the popular methods of activation quantization are by using approximation methods. In these methods, the forward pass through NN usually has an ideal quantization

TABLE II
TABLE OF TRAINING HYPERPARAMETERS OF THE NN-BASED DPD

Hyperparameter	Value
Optimization	Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate (DPD-NN)	1×10^{-4}
Learning rate (Auxiliary-NN)	5×10^{-4}
Batch size (DPD-NN)	4096
Batch size (Auxiliary-NN)	2048
Data length	2^{18}
Training epochs	30

and the corresponding backward pass is implemented by using some approximated function such as a smoothened version of the ideal quantization function or a straight through estimator. One challenge with these approximation methods is that they have a gradient mismatch problem as described in [38]. The soft-DAC avoids gradient mismatch as it is a differentiable function.

B. Initialization and Training of the NN-Based DPDs

The linear CNN layers in Fig. 3 are equivalent to FIR filters. The tap values of the CNN layer is initialized using "impulse initialization" i.e. all weights are set to zeros except for the center weight whose value is set to one. The weights and biases of the remaining FFNN part were initialized using Kaiming uniform initialization i.e. by randomly sampling a uniform distribution $U(-1, 1)\sqrt{k}$, where $k = 1/(\text{number of weights or biases})$. The network size hyperparameters and initialization are summarized in Table I. It is also possible to initialize the first and the last 1-D CNNs using the known linear DPD response. We observed that by doing so the overall performance did not change, however, convergence is achieved quicker.

For both NN-based DPDs i.e. NNILA and NNDLA, we used gradient descent back-propagation with the mean square error loss function and the Adam optimizer [39]. We used sequences of 2^{18} symbols to train the NNs. The batch size should be large enough to capture the transmitter memory. We used a larger batch size to minimize the fluctuation in the mean and the variance of individual batches because of the BN layer operation. The learning rates were determined by doing a grid search on a logarithmic scale. The training hyperparameters are summarized in Table II.

For NNILA, the training data consists of the received signal $\boldsymbol{y}[n] = [\text{Re}(y[n]), \text{Im}(y[n])]$ and pre-distorted signal $\boldsymbol{z}[n] = [\text{Re}(z[n]), \text{Im}(z[n])]$ as inputs and targets, respectively. In the first step of NNDLA, the auxiliary channel model is trained by using pre-distorted signal $\boldsymbol{z}[n] = [\text{Re}(z[n]), \text{Im}(z[n])]$ and the soft symbols output of the receiver DSP $\boldsymbol{y}[n] = [\text{Re}(y[n]), \text{Im}(y[n])]$ as inputs and targets. While in the second step, the cascaded NN i.e. pre-distorter followed by auxiliary channel model, uses $\boldsymbol{x}[n] = [\text{Re}(x[n]), \text{Im}(x[n])]$ as its input and target. Note that, in the second step, only the pre-distorter part of the cascaded NN is updated and the auxiliary channel model part is kept fixed.

V. RESULTS

In this section, we explain the procedure and the results of our experimental study. The training of the considered DPDs was done by using either the DLA or the ILA described in Section III. In the first iteration of the training, a predistorted signal z obtained by a linear static pre-distortion filter was used. This linear pre-distortion filter was already known from a characterization of the DACs and the DAs in the electrical domain. In the following subsection, we describe how we determined the memory needed in the considered DPDs.

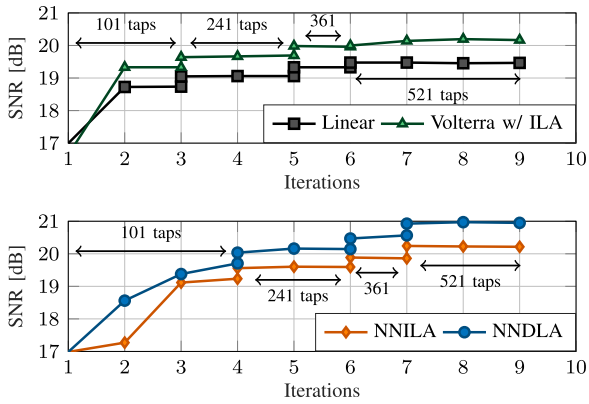

Fig. 6. Training of different DPDs at 460 mV DACs' voltage by increasing the memory length of the 4×4 real-valued MIMO equalizer in the receiver DSP.

A. Required Filter Length

A linear DPD with very long memory was trained in order to determine the memory required in the pre-distorter. The impulse response of the linear DPD filter after convergence of the adaptive training algorithm is shown in the Fig. 5. We observe that signal reflections are present even around 200 symbol duration delay. These reflections were possibly generated by the RF cables/connections. Thus, we set the length of linear DPD as well as the first order coefficients of Volterra to 441 taps. In NN-based DPDs (NNDLA, NNILA), memory (the sum of tap-lengths in CNNs) was set to around 440 taps.

The required memory for the second and the third order Volterra kernels for the DPD are determined in a similar way. The memory order and depth for the second order terms are $M_2 = 10$ and $d_2 = 4$, respectively. For the third order terms, we used $M_3 = 5$, $d_2 = 2$ and $d_3 = 3$, respectively. In total, the Volterra series-based DPD uses 105 s order and 99 third order coefficients along with 441 linear coefficient and one bias coefficient.

B. Training Procedure

As explained previously, we use the following training procedure so that most of the transmitter impairments are compensated at the transmitter side via the DPD and not by the adaptive 4×4 real-valued MIMO equalizer at the receiver DSP. We first train the DPD when the MIMO equalizer has a memory length of 101 taps. Then, after convergence, the memory length of the 4×4 real-valued MIMO equalizer is increased, and the DPD is trained again. More specifically, the MIMO filter lengths is increased from 101 to 241, 361 and 521. In Fig. 6, we show the training of the considered DPDs over the iterations for the case of uniform 64-QAM signal. We observe that all DPDs converged within 9 iterations. In these results, the DACs output voltages were set to

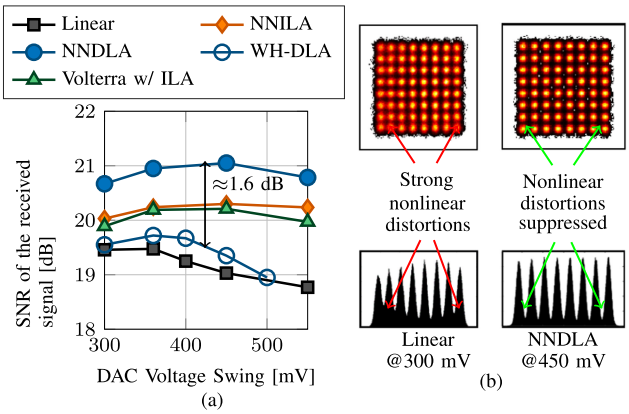

Fig. 7. (a) Performance of the considered DPDs at different DACs' voltages with uniform 64-QAM. (b) Constellations and histograms of the soft symbol outputs of the receiver DSP y after the convergence, for the Linear and the NNDLA DPD.

optimum values for the corresponding DPD as described in the following subsection.

C. DAC Voltage Variation

We vary the output voltage of the DACs to change the strength of the transmitter nonlinearity. We re-train each of the DPDs for every value of the DACs' output voltage. In Fig. 7(a), the SNR values of each DPD technique are plotted against the variation in the DACs' output voltage. We observe that with a linear DPD employed at the transmitter, the SNR decreases with increasing the DACs' output voltage due to the increased nonlinear distortions from the transmitter. For linear DPD, the optimum operating point is around 300 mV.

Further, applying nonlinear DPDs at the transmitter gives improvements in the SNR even at a lower DAC output voltage of 300 mV. This shows the presence of significant nonlinear distortions at the transmitter even at low voltage. Furthermore, we see that unlike linear DPD, increasing the DACs' output voltage beyond 300 mV improves the SNR. The optimum DACs' voltage for nonlinear DPDs is approximately 450 mV which is 50% higher than that of the linear DPD. The NN-based and the Volterra series-based DPD, both trained using the ILA architecture, attain almost similar performance. Although the SNR gains obtained by using the former is slightly higher. Moreover, the NNDLA (i.e. the NN-based DPD trained using DLA) provides the highest gain in SNR which is 1.6 dB with respect to the linear DPD, after 9 iterations of Tx DPD training. NNDLA DPD obtains better SNR values than NNILA mainly because of the difference in the training architectures: DLA and ILA. A detailed explanation of this point can be found in [40].

At this point, we show how the proposed NN-DPD is different from WH architecture. For this purpose, the WH-DPD is obtained from reconfiguration of the proposed NN-DPD as follows: The Section (A) and (C) and the short-cut connection between them is kept as it is. While, we remove the memory from the Section (B) as the nonlinear part of a typical WH structure is memoryless. In detail, the Section (B) was modified by removing the CNN layer (FFNN layer 1) i.e. the layer with 21 11-tap convolutions such that the output from the Section (A) fans out directly to the second layer (FFNN layer 2, the layer with 12 neurons) of the Section (B). Overall, the fully connected FFNN part has layers with 12, 8, 8, 1 neurons with leaky ReLU activation function. The WH-DPD was trained using DLA by using an auxiliary channel whose architecture is identical to the one used for NN-DLA. The SNR performance at different DAC voltages is shown in the Fig. 7. We see that WH-DPD only adds up to 0.2dB gain in the SNR to the linear DPD. This also shows that the nonlinearity is mixed with memory in the system and the proposed NN-DPD architecture captures this nonlinearity mixed with memory.

Remark: We also tried a Volterra-based DPD trained using DLA. In order to have a fair comparison, we used the auxiliary channel-NN of NNDLA as a surrogate for training the Volterra-DLA-DPD. The Volterra-DLA was implemented within the NN-framework. This has advantage as the Volterra-DLA-DPD-NN can be trained easily by using the auxiliary channel-NN in the same manner as done for the NN-DLA. The Volterra-DLA-DPD NN takes all possible Volterra terms as its input features and learns the required weights and bias in order to produce the output. A batch normalization is applied at the output. Surprisingly, we did not observe significant gains with Volterra-DLA-DPD-NN. At 460 mV DACs' voltage, the Volterra-DLA-DPD-NN added only around 0.1dB gain in the SNR in comparison to Volterra-ILA. The Volterra-DLA needs more investigation.

Next, we visualize the impact of applying the NNDLA at the transmitter by carrying out a spectral analysis and by plotting the signal constellation after the entire Rx-DSP. In Fig. 7(b), we plot the constellation diagram of one of the polarizations of the experimentally obtained signals along with corresponding histogram of one dimension. It can be clearly seen that when only a linear DPD is applied, the received signal is contaminated with power dependent distortions. In contrast, these distortions are suppressed and not visible in the received signal when the NNDLA DPD is applied at the transmitter. Fig. 8 presents a spectral analysis of the output signal of the auxiliary NN when there is a test signal is applied as input. The test signal is a 64 Gbaud 64-QAM signal upsampled to 2 sample per symbol (sps) by zero-insertion and filtering using a brick-wall filter of 64 GHz bandwidth. We see that when the test signal is applied, without any DPD, to the trained auxiliary NN model, the auxiliary NN output has a distorted in-band spectrum along with out-of-band ($> |32|$ GHz) spectral components resulted from the nonlinearity. When the linear DPD is applied to the test signal before feeding it into the auxiliary NN, the output signal of the auxiliary NN has only its in-band spectrum corrected while the out-of-band spectrum generated by nonlinearity stays as is. On the other hand, when the test signal is passed through the NN-DPD and then fed to the auxiliary NN, the auxiliary NN output has a flat in-band spectrum together with a suppressed out-of-band spectrum.

We have also trained and tested a look-up table (LUT)-based DPD using this auxiliary NN model of the transceivers. We consider a the LUTs of memory 3 i.e. a correction to a symbol is based on that symbol and its adjacent symbols. The correction coefficients of the LUTs were learned by the method given in [41], [42] while the trained "Linear" DPD of 401 taps was

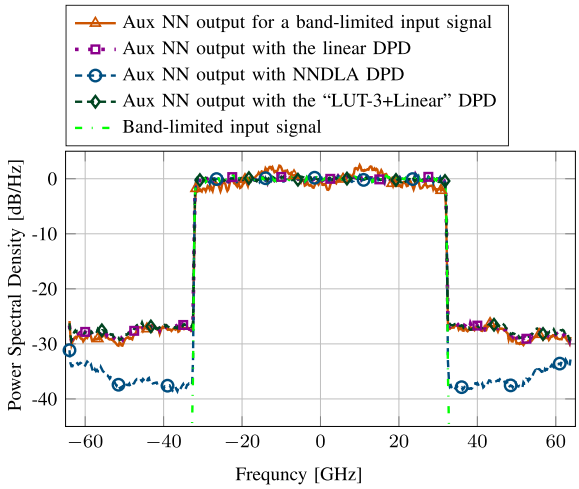

Fig. 8. Noiseless spectral analysis of a brick-wall band-limited 64 Gbaud 64-QAM (at 2 sps) signal using experimentally trained NN models shows the transmitter distortions and its compensation. With the band-limited 64-QAM signal (dashed, green curve) as input, there is out-of-band noise at the output of auxiliary NN due to the nonlinearity (solid, orange curve). With NNDLA DPD (densely dash-dotted, blue curve) this nonlinear noise is suppressed.

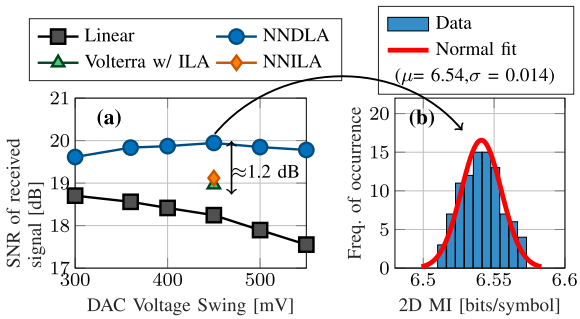

Fig. 9. (a) Received signal SNR over a variation in DACs' voltage for PCS 256 QAM. (b) A histogram of the 2-D MI values obtained by transmitting 100 statistically independent symbol sequences.

applied in order to compensate for the large channel memory. This combined DPD is referred as "LUT-3+Linear" DPD. Two LUTs were trained each for the in-phase and the quadrature-phase tributary at 1 sps for a 128 Gbaud 64-QAM signal. We compared the performance of the "LUT-3+Linear" DPD with that of the Linear and the NNDLA DPD in terms of the normalized mean square error (NMSE) between the transmit and the auxiliary NN output symbols. We observed that "LUT-3+Linear" gave 1.6dB smaller NMSE than the Linear DPD. The NNDLA gave about 10dB smaller NMSE than the "LUT-3+Linear" DPD. The NMSE difference between the "LUT-3+Linear" and the NNDLA shows that nonlinearity is mixed with the memory. One may expect more compensation gains by increasing the LUT memory. However, the current LUT has already $8^3 = 512$ entries and its size grows exponentially with memory if implementation is not optimized. The spectrum of the auxiliary output signal when the test signal with "LUT-3+Linear" DPD is fed at its input is also shown in Fig. 8.

In the next experiment, we test the DPDs using the PCS-256-QAM format and quantify the SNR performance. Fig. 9 shows the SNR values obtained for different DAC voltages. We

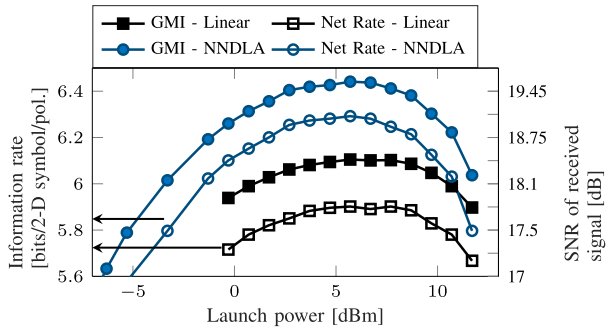

Fig. 10. SNR, GMI and net information rate at different launch powers for 80 km SSMF transmission of PCS 256-QAM format signal pre-distorted using the Linear or the NNDLA DPD.

observed a trend similar to the uniform 64-QAM format. The SNR gain that NNDLA provides with respect to the linear DPD is approximately 1.2 dB.

D. Verification of Pattern Independence

A common concern in the NN-based techniques is dependence on patterns. We applied our proposed NNDLA predistortion on 100 statistically independent symbol sequences which were not used in the training. The corresponding 100 predistorted waveform were transmitted through the experimental setup in back-to-back configuration with 450 mV DAC voltage and their performances were evaluated. In Fig. 9(b), a histogram of the observed 2-D MI values is plotted. A fitted curve with Gaussian approximation shows that the standard deviation is very small of around 0.014 bits/symbol/polarization indicating that the NNDLA is nearly pattern independent.

E. Evaluation in the Fiber Transmission Scenario

In further investigations, we apply our trained NNDLA and the linear DPD at the transmitter with PCS 256-QAM and test it over a link of 80 km SSMF. Fig. 10 shows the SNR, GMI and net rate of the received signal over different transmit powers. The figure indicates that the optimum launch power is around 6 dBm for both DPDs. Furthermore, applying NNDLA at the transmitter results into a significant SNR gain in comparison to the linear DPD. At the optimum launch power, the SNR gain is around 1.2 dB which is the same as observed in experiments with the back-to-back configuration for PCS 256-QAM. The NNDLA transmission achieves GMI of 6.44 bits/symbol/polarization in comparison to 6.1 bits/symbol/polarization obtained by applying the linear DPD. Furthermore, the net rate increases from 5.9 bits/symbol/polarization for the linear DPD to 6.3 bits/symbol/polarization for the case of the NNDLA. Moreover, we observed that the FEC decoding loss is slightly less when the NNDLA is applied. This is attributed to the more Gaussian-like distribution of the soft symbols at the Rx-DSP output (y in Fig. 7(b)) when using the NNDLA instead of the linear DPD. The lower decoding loss is due to the assumption of conventional FEC decoding algorithms that received symbols follow a Gaussian likelihood. Overall, our proposed NNDLA increases the net bit rate to a record 1.61 Tb/s over a single-channel of 80 km SSMF.

TABLE III
REAL-VALUED MULTIPLICATIONS PER CHANNEL NEEDED FOR NN-BASED DPD

NN layer	Real-valued multiplications
First CNN layer	101
Last CNN layer	301
CNN layer before FFNN	11×21 = 231
FFNN layer 1	21×12 + 12 = 264
FFNN layer 2	12×8 + 8 = 104
FFNN layer 3	8×8 + 8 = 72
FFNN layer 4	8 ×1= 8
Total	1081

TABLE IV
REAL-VALUED MULTIPLICATIONS NEEDED PER CHANNEL FOR EACH OF THE CONSIDERED DPDS

DPD type	Real-valued multiplications
Linear	441
Volterra	659
NNDLA/NNILA	1081

VI. COMPUTATIONAL COMPLEXITY AND PRUNING

In this section, we compare the compuational complexity of each of the considered DPDs. As a figure of merit, we compute the required number of real-valued multiplications to implement each DPDs. The number of real-valued multiplications per stage of the NN-based DPDs is provided in Table III. The batch normalization layers and soft-DAC are not accounted for in the computation as similar processing is required for the other DPDs in order to generate integers prior to loading the DACs.

The linear DPD has 441 coefficients, thus, require 441 real-valued multiplications for each channel (I/Q). The Volterra filter has three kernel orders requiring 441, 105 and 99 coefficients for the first, second and third order respectively. Additionally, 14 real-valued multiplications are needed to generate the second and third order terms. By following [43], we consider that already computed lower-order Volterra terms are used to generate other possible higher-order terms so that the real-valued multiplications for Volterra DPD are not over-counted. In total Volterra implementation requires 659 multiplications. Table IV summarizes the complexity. We see that our proposed NN-DPD requires about 64% more real-valued multiplications than the Volterra DPD.

To understand performance-complexity trade-off, we reduce the complexity of NN-based DPD by pruning it after the NNDLA has converged in the experiments. We prune only the middle FFNN structure (ie. the Section (B) from Fig. 3) as the other layers are linear and are common to other DPDs as well. The pruning method proposed in [44] was applied on each channel (I/Q) separately. We used L1-norm as pruning criteria such that the smallest weights and biases in the FFNN layer are forced to zero after pruning. A target or final pruning factor s_f is achieved in N training steps or epochs by pruning with a s_c factor every ΔN epochs. The pruning factor for a given epoch s_c is given by the following relation

$$s_c = s_f + s_f \left(1 - \frac{\lfloor n/\Delta N \rfloor}{N}\right)^3, \tag{8}$$

We used $N = 20$ epochs to achieve a final pruning factor where pruning by s_c was applied every ΔN epochs. Pruning reduces the complexity by avoiding several multiplications. It is even possible that all the weights feeding a neuron are zeroed by the pruning process, thereby further reducing the complexity as the computation of the activation of that neuron can then also be omitted. For our case, the decrease in the complexity due to deactivated neurons is very small, and thus, ignored. The decrease in the number of real-valued multiplications in the middle FFNN structure as a result of pruning is considered as $651 \times s_f + 28$.

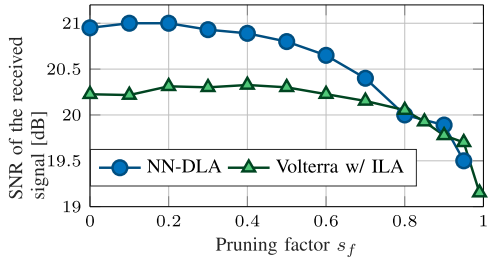

Fig. 11. SNR of the received signal for different pruning factor s_f applied on nonlinear parts of NNDLA and Volterra w/ ILA DPDs. The number of real-valued multiplication reduces by $651 \times s_f + 28$ and $218 \times s_f$ for NNDLA and Volterra DPD, respectively.

We also pruned the nonlinear part of the Volterra w/ ILA DPD kernels i.e. the coefficients of the second and the third order. The pruning was again done by forcing the smallest magnitude coefficients to zero. In Fig. 11, we plot the performance of NNDLA and Volterra DPD over the experimental setup at various pruning factors. For Volterra DPD, a pruning up to a factor of 0.6 does not add any penalty, instead, we observe some improvement (0.1dB) in the SNR when pruning factor is around 0.3. This is attributed to the fact that a reduced number of Volterra kernel increases the accuracy of the least squares based Volterra DPD. At pruning factor s_f of 0.6, Volterra DPD has around 530 kernel coefficients.

For NNDLA, we see that a pruning by a factor of 0.2 can be applied without causing performance degradation, while larger pruning factors add penalty to the received SNR. At pruning factor of 0.8 both DPDs have similar performance while it requires 560 and 485 real-valued multiplications for NNDLA and Volterra DPD, respectively. A pruning by a factor of 0.4 still gives good performance while reducing the overall per channel complexity of the NNDLA to around 820 real-valued multipliers.

VII. CONCLUSION

In this paper, we reported on a new record transmission of 1.61Tb/s data rate over a single channel of 80km of standard single mode fiber that was achieved using a novel neural network-based digital pre-distorter. The proposed DPD has been compared with a Volterra series-based ILA-DPD and a linear DPD. In addition, we evaluated the performance of the proposed DPD by training it using direct learning and indirect learning architecture. The NN-based DPD trained using DLA adds SNR gain of around 1.6 dB and 1.2 dB with respect to a linear DPD for uniform 64-QAM and PCS 256-QAM formats, respectively. Further, we show that by applying pruning the computation complexity of the proposed DPD can be reduced significantly with no or only minor losses in the SNR.

REFERENCES

[1] F. Buchali *et al.*, "DCI field trial demonstrating 1.3-Tb/s single-channel and 50.8-Tb/s WDM transmission capacity," *J. Lightw. Technol.*, vol. 38, no. 9, pp. 2710–2718, May 2020.

[2] F. Buchali *et al.*, "1.52 Tb/s single carrier transmission supported by a 128 GSa/s SiGe DAC," in *Proc. Opt. Fiber Commun. Conf.*, 2020, pp. 1–3, Paper Th 4C-2.

[3] H. Sun *et al.*, "800G DSP ASIC design using probabilistic shaping and digital sub-carrier multiplexing," *J. Lightw. Technol.*, vol. 38, no. 17, pp. 4744–4756, Sep. 2020.

[4] "Ciena wavelogic 5, 800G," Accessed: Feb. 20, 2019. [Online]. Available: https://www.ciena.com/insights/articles/Ciena-unveils-WaveLogic-5-800G-and-so-much-more.html

[5] D. Rafique, A. Napoli, S. Calabro, and B. Spinnler, "Digital preemphasis in optical communication systems: On the DAC requirements for terabit transmission applications," *J. Lightw. Technol.*, vol. 32, no. 19, pp. 3247–3256, Oct. 2014.

[6] J. Zhang, H.-C. Chien, Y. Xia, Y. Chen, and J. Xiao, "A novel adaptive digital pre-equalization scheme for bandwidth limited optical coherent system with DAC for signal generation," in *Proc. Opt. Fiber Commun. Conf.*, 2014, pp. 1–3.

[7] 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.

[8] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Trans. Signal Process.*, vol. 46, no. 6, pp. 1726–1730, Jun. 1998.

[9] 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, 2001.

[10] 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.

[11] 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, Jan. 2007.

[12] P. W. Berenguer *et al.*, "Nonlinear digital pre-distortion of transmitter components," *J. Lightw. Technol.*, vol. 34, no. 8, pp. 1739–1745, Apr. 2016.

[13] H. Faig, Y. Yoffe, E. Wohlgemuth, and D. Sadot, "Dimensions-reduced Volterra digital pre-distortion based on orthogonal basis for band-limited nonlinear opto-electronic components," *IEEE Photon. J.*, vol. 11, no. 1, pp. 1–13, Feb. 2019.

[14] R. Elschner *et al.*, "Improving achievable information rates of 64-GBd PDM-64QAM by nonlinear transmitter predistortion," in *Proc. Opt. Fiber Commun. Conf.*, 2018, pp. 1–3.

[15] G. Khanna, B. Spinnler, S. Calabró, E. De Man, and N. Hanik, "A robust adaptive pre-distortion method for optical communication transmitters," *IEEE Photon. Technol. Lett.*, vol. 28, no. 7, pp. 752–755, Apr. 2016.

[16] Y. Yoffe *et al.*, "Low-resolution digital pre-compensation enabled by digital resolution enhancer," *J. Lightw. Technol.*, vol. 37, no. 6, pp. 1543–1551, Mar. 2019.

[17] A. Napoli *et al.*, "Digital pre-compensation techniques enabling high-capacity bandwidth variable transponders," *Opt. Commun.*, vol. 409, pp. 52–65, 2018.

[18] D. Psaltis, A. Sideris, and A. A. Yamamura, "A multilayered neural network controller," *IEEE control Syst. Mag.*, vol. 8, no. 2, pp. 17–21, Apr. 1988.

[19] A. Bernardini, M. Carrarini, and S. D. Fina, "The use of a neural net for copeing with nonlinear distortions," in *Proc. 20th Eur. Microw. Conf.*, 1990, vol. 2, pp. 1718–1723.

[20] T. Gotthans, G. Baudoin, and A. Mbaye, "Digital predistortion with advance/delay neural network and comparison with Volterra derived models," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 2014, pp. 811–815.

[21] Y. Wu, U. Gustavsson, A. G. i Amat, and H. Wymeersch, "Residual neural networks for digital predistortion," in *Proc. IEEE Glob. Commun. Conf.*, 2020, pp. 01–06.

[22] 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.*, 2019, pp. 296–301.

[23] G. Paryanti, H. Faig, S. L. Rokach, and D. Sadot, "A direct learning approach for neural network based pre-distortion for coherent nonlinear optical transmitter," *J. Lightw. Technol.*, vol. 38, no. 15, pp. 3883–3896, Aug. 2020.

[24] M. Abu-Romoh, S. Sygletos, I. D. Phillips, and W. Forysiak, "Neural-network-based pre-distortion method to compensate for low resolution DAC nonlinearity," in *Proc. 45th Eur. Conf. Opt. Commun.*, 2019, pp. 1–4.

[25] R. Hongyo, Y. Egashira, T. M. Hone, and K. Yamaguchi, "Deep neural network-based digital predistorter for Doherty power amplifiers," *IEEE Microw. Wireless Compon. Lett.*, vol. 29, no. 2, pp. 146–148, Feb. 2019.

[26] X. Hu *et al.*, "Convolutional neural network for behavioral modeling and predistortion of wideband power amplifiers," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, 2021, doi: 10.1109/TNNLS.2021.3054867.

[27] J. Sun, W. Shi, Z. Yang, J. Yang, and G. Gui, "Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10 348–10 356, Nov. 2019.

[28] S. Boumaiza and F. Mkadem, "Wideband RF power amplifier predistortion using real-valued time-delay neural networks," in *Proc. Eur. Microw. Conf.*, 2009, pp. 1449–1452.

[29] M. Schaedler, M. Kuschnerov, S. Calabró, F. Pittalá, C. Bluemm, and S. Pachnicke, "AI-based digital predistortion for IQ Mach-Zehnder modulators," in *Proc. Asia Comm. Photon. Conf.*, 2019, pp. 1–3.

[30] T. Sasai *et al.*, "Wiener-Hammerstein model and its learning for nonlinear digital pre-distortion of optical transmitters," *Opt. Exp.*, vol. 28, no. 21, pp. 30952–30963, Oct. 2020. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-28-21-30952

[31] V. Bajaj, F. Buchali, M. Changon, S. Wahls, and V. Aref, "Single-channel 1.61 Tb/s optical coherent transmission enabled by neural network-based digital pre-distortion," in *Proc. Eur. Conf. Opt. Commun.*, 2020, pp. 1–4, Paper Tu1D5.

[32] V. Bajaj, F. Buchali, M. Changon, S. Wahls, and V. Aref, "54.5 Tb/s WDM transmission over field deployed fiber enabled by neural network-based digital pre-distortion," in *Proc. Opt. Fiber Commun. Conf.*, 2021, pp. 1–3.

[33] L. Schmalen, V. Aref, J. Cho, D. Suikat, D. Rösener, and A. Leven, "Spatially coupled soft-decision error correction for future lightwave systems," *J. Lightw. Technol.*, vol. 33, no. 5, pp. 1109–1116, Mar. 2015.

[34] D. R. Morgan, Z. Ma, and L. Ding, "Reducing measurement noise effects in digital predistortion of RF power amplifiers," in *Proc. IEEE Int. Conf. Commun.*, 2003, vol. 4, pp. 2436–2439.

[35] T. Ogunfunmi, *Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches*. Berlin, Germany: Springer, 2007.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[37] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *CoRR*, 2021. [Online]. Available: https://arxiv.org/abs/2103.13630

[38] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave Gaussian quantization," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 5918–5926. [Online]. Available: http://arxiv.org/abs/1702.00953

[39] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learn. Representations*, Dec. 2017, *arXiv:1412.6980*.

[40] H. Paaso and A. Mammela, "Comparison of direct learning and indirect learning predistortion architectures," in *Proc. IEEE Int. Symp. Wireless Commun. Syst.*, 2008, pp. 309–313.

[41] S. Zhalehpour, J. Lin, W. Shi, and L. A. Rusch, "Reduced-size lookup tables enabling higher-order QAM with all-silicon IQ modulators," *Opt. Exp.*, vol. 27, no. 17, pp. 24 243–24 259, 2019.

[42] J. H. Ke, Y. Gao, and J. C. Cartledge, "400 Gbit/s single-carrier and 1 Tbit/s three-carrier superchannel signals using dual polarization 16-QAM with look-up table correction and optical pulse shaping," *Opt. Exp.*, vol. 22, no. 1, pp. 71–84, Jan. 2014. [Online]. Available: http://www.osapublishing.org/oe/abstract.cfm?URI=oe-22-1-71

[43] 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 Techn.*, vol. 58, no. 6, pp. 1510–1520, Jun. 2010.

[44] M. Zhu and S. Gupta, "To prune, or not to prune: Exploring the efficacy of pruning for model compression," 2017. [Online]. Available: https://openreview.net/forum?id=S1lN69AT-