

# Advanced Convolutional Neural Networks for Nonlinearity Mitigation in Long-Haul WDM Transmission Systems

Oleg Sidelnikov, Alexey Redyuk, Stylianos Sygletos, Mikhail Fedoruk, and Sergei Turitsyn, *Fellow, OSA*

**Abstract**—Practical implementation of digital signal processing for mitigation of transmission impairments in optical communication systems requires reduction of the complexity of the underlying algorithms. Here, we investigate the application of convolutional neural networks for compensating nonlinear signal distortions in a 3200 km fiber-optic 11x400-Gb/s WDM PDM-16QAM transmission link with a focus on the optimization of the corresponding algorithmic complexity. We propose a design that includes original initialization of the weights of the layers by a filter predefined through the training a single-layer convolutional neural network. Furthermore, we use an enhanced activation function that takes into account nonlinear interactions between neighbouring symbols. To increase learning efficiency, we apply a layer-wise training scheme followed by joint optimization of all weights applying additional training to all of them together in the large multi-layer network. We examine application of the proposed convolutional neural network for the nonlinearity compensation using only one sample per symbol and evaluate complexity and performance of the proposed technique.

**Index Terms**—Convolutional neural networks. Nonlinearity mitigation in fiber-optic links.

## I. INTRODUCTION

Capacity demand in communication networks follows a stable increasing trend over the recent decades due to the continuing expansion of current and emerging digital applications and services. Assuming that this trend will maintain, the potential disparity between growth rates of future traffic demand and available network capacity is expected to create a what is known as “capacity crunch” problem. This fact calls for new approaches to improve the transmission performance of optical fiber links.

In general, there are two important questions related to optical networking: What is the best way to design new high-capacity transmission systems? and How to manage the existing systems in the most efficient way? The key approach to contend with the future demand is parallelization - i.e. to increase the number of communication channels in the spectral or spatial dimension. These new designs can be used in next

generation optical communication systems. However, in the already installed fiber links, possibilities are limited by the existing infrastructure, requiring different technical approaches for optimizing the performance. Overcoming fiber nonlinearity is one of the most challenging tasks in those systems and it is a major limiting factor for extending their capacity.

A nonlinear fiber channel differs substantially from the classical linear additive white Gaussian noise channel by the complexity of the link between the output and input signal. The output signal is given by the solution of a nonlinear stochastic partial differential equation(s) with the input signal defining the initial conditions of the problem. It is well understood nowadays, that nonlinear fiber communication channels require the development of conceptually new digital processing methods capable to deal with the nonlinear transmission impairments (see e.g. [1], [2], [3], [4], [5], [6] and references therein). One of those methods is digital backward propagation (DBP) [7], [8] that digitally mimics the propagation of a signal through a fiber in the reversed direction at the receiver. Coincidentally, backward propagation methodology is a central building block in many machine learning (ML) approaches such as neural networks. The basics of back-propagation were introduced in 1960s in the context of control theory [9] and then applied in the field of machine learning in [10]. In the context of neural networks the back-propagation algorithm is used to identify the optimum layer weights for a specific training set.

Machine learning methods are generally well suited for applications in complex nonlinear systems. Therefore, it is rather natural that ML techniques emerged as a promising tool to improve performance of complex modern fiber-optic networks ([11], [12], [13], [14] and references therein), with the number of publications in the area following an explosive growth. Neural networks (NN), in particular, are extremely popular in this field due to the high classification accuracy they can achieve. However, it is also well-known that there is a lack of clear and well-defined rules for designing an efficient neural network architecture to address a particular applications. The number and size of hidden layers, the type of activation function and other design options are often addressed by a reasonable enumeration of possible configurations, and heuristic approaches. Any apriori knowledge of the system’s behaviour can be extremely useful in the NN training to achieve fast convergence at a better optimum operating point.

Although DBP-based equalization has been supported by robust mathematical models, their associated complexity has

O. Sidelnikov is with the Novosibirsk State University, Novosibirsk, 630090, Russia (e-mail: o.sidelnikov@g.nsu.ru).

A. Redyuk and M. Fedoruk are with the Novosibirsk State University, Novosibirsk 630090, Russia, and also with the Institute of Computational Technologies, Siberian Branch, Russian Academy of Sciences, Novosibirsk 630090, Russia (e-mail: alexey.redyuk@gmail.com; rector@nsu.ru).

S. Sygletos is with the Aston Institute of Photonic Technologies, Aston University, Birmingham B4 7ET, U.K. (e-mail: s.sygletos@aston.ac.uk).

S. Turitsyn is with the Novosibirsk State University, Novosibirsk 630090, Russia, and also with the Aston Institute of Photonic Technologies, Aston University, Birmingham B4 7ET, U.K. (e-mail: s.k.turitsyn@aston.ac.uk).

prevented any real time implementation in optical communication systems [15]. On the other hand, it has been recently shown that deep neural networks can provide a good approximation of DBP at lower computational cost [16]. The alternating layers of the proposed architecture corresponded to the linear and nonlinear signal transformations of the split-step Fourier method (SSFM). The resulting method was referred to as learned DBP (LDBP). However, contrary to the conventional DBP, which requires exact knowledge of all the transmission parameters to be effective, the parameters of the LDBP equalizer can be jointly optimized through a supervised training process allowing a "blind" operation even on a totally unknown channel.

In this work we develop a new design of a deep convolutional neural network (DCNN) for mitigating the nonlinear signal distortions in a long-haul fiber communication system. To adjust the proposed DCNN to the channel nonlinearity we applied an activation function based on enhanced SSFM [17] that takes into account nonlinear interaction of the symbol under consideration with neighbouring symbols both from the same and from surrounding spectral channels. We divide nonlinear layers into groups of filters depending on the distance between the processed spectral channels, which allows us to find a trade-off between the computational complexity of the proposed scheme and its performance. We demonstrated here that nonlinearity compensation is possible using sampling with just one sample per symbol (SpS). We examined the performance of the proposed scheme in a 3200 km 11x400-Gb/s RRC WDM PDM-16QAM transmission system when processing single channel or 5 neighboring channels simultaneously. We also conducted extensive analysis of computational complexity of the equalizer based on deep convolutional neural network and showed the superiority of the proposed scheme over conventional DBP methods.

The paper is organized as follows. In Section II, we briefly present the theoretical background of the design of the convolution neural network and conceptual connection between DCNN and DBP. Next, we introduce a detailed description of the proposed DCNN-based nonlinear equalizer, including complexity analysis. Section III provides detailed description of the particular transmission system and numerical modeling parameters. Section IV presents the results of numerical modeling and the comparison between DCNN and DBP performance. Section V concludes the paper.

## II. CNN-BASED PROCESSING AT THE RECEIVER

Based on information theory, an optical communication system can be considered as a nonlinear channel with memory defined by the interplay of chromatic dispersion (CD) and Kerr effect. A conventional optical signal launched in the fiber link presents an analog function  $A(z=0, t) = \sum_k a_k f(t - kT)$  of time  $t$ , where  $a_k$  are complex transmitted symbols,  $k$  is a number of time slot,  $T$  is a symbol interval, and  $f(t)$  is a waveform of a carrier pulse. This offers a natural discrete representation of the signal  $A(z=0, t)$  in the form of an infinite (in  $k$ ) discrete-time series of vectors  $\xi_k = (\xi_k^1, \dots, \xi_k^n)$ , where  $\{\xi_k^l\}$  is a set of regularly spaced signal samples for

$j = 1, \dots, n$  and  $n/T$  is a sampling rate. Obviously, at one sample per symbol, such a vector has just one component and represent a scalar, that is a particular case of the considered approach. In a similar manner we can represent received signal  $A(z=L, t)$  at the receiver side as a time series of vectors  $\eta_k = (\eta_k^1, \dots, \eta_k^n)$ . Due to channel memory a finite set of input signals  $\xi_{k-M}, \dots, \xi_{k+M}$  has an impact on the output signal  $\eta_k$ , where  $M$  is a memory parameter.

Convolutional neural networks (CNN) are suitable tools for time series analysis by processing the elements of the series in blocks sliding along the input data. A CNN of  $N$  layers transforms an input vector  $\mathbf{x}$  to an output vector  $\bar{\mathbf{y}}$  by alternating between convolution with vectors  $\mathbf{w}^{(i)}$  and point-wise nonlinear activation functions  $f(\mathbf{x})$ :  $\mathbf{x}^{(i)} = f(\mathbf{w}^{(i)} * \mathbf{x}^{(i-1)} + \mathbf{b}^{(i)})$ , where  $i = 1, \dots, N$  is an index of a layer,  $\text{sign} *$  denotes convolution product,  $\mathbf{b}^{(i)}$  is a bias vector,  $\mathbf{x}^{(0)} = \mathbf{x}$  and  $\mathbf{x}^{(N)} = \bar{\mathbf{y}}$ . In contrast to fully-connected neural networks, where  $\mathbf{w}^{(i)}$  is typically represented as a dense matrix describing connection between all neurons from neighboring layers, in convolutional neural networks  $\mathbf{w}^{(i)}$  is usually a matrix or vector of specific length and it is known as filter or kernel. The elements of the  $\mathbf{w}^{(i)}$  and  $\mathbf{b}^{(i)}$  vectors are considered as learning parameters, whereas  $f(\mathbf{x})$  is a fixed function. During the training process the learning parameters are updated in a way that minimizes the difference between the estimated output vector  $\bar{\mathbf{y}}$  and the target vector  $\mathbf{y}$  of the training set.

Here, we consider a deep convolutional neural network with structure that inherits from the digital back-propagation concept [7], see Fig. 1. It is an alternation of linear (convolutional) and nonlinear (activation function) layers, with the linear layers performing the compensation of the accumulated chromatic dispersion and the nonlinear layers undertaking to compensate the response of the medium. We consider a sampling rate of one sample per symbol. As input  $\mathbf{x}$  we consider the vector of received samples  $\{\eta_k\}$ ,  $\{\xi_k\}$  represents the estimated symbols at the output of the architecture, target vector  $\mathbf{y}$  is the same as the vector of the transmitted symbols  $\{a_k\}$ . The parameters of all layers are jointly optimized after layer-wise training during the NN learning stage.

In our study, we investigate WDM-signal transmission. To take into account inter-channel interactions the proposed DCNN simulate the DBP method based on coupled nonlinear Schrödinger equations [18], [19]:

$$\begin{aligned} \frac{\partial A_c^{x/y}}{\partial z} = & -d_c \frac{\partial A_c^{x/y}}{\partial t} - i \frac{\beta_2}{2} \frac{\partial^2 A_c^{x/y}}{\partial t^2} + \\ & + i \frac{8\gamma'}{9} \left( |A_c^x|^2 + |A_c^y|^2 + 2 \sum_{s \neq c} \left( |A_s^x|^2 + |A_s^y|^2 \right) \right) A_c^{x/y}, \end{aligned} \quad (1)$$

where  $A_c^{x/y}$  is the complex field envelopes for x- and y-polarization,  $c$  is the number of a spectral channel,  $d_c = c\beta_2\Delta\omega$  is the dispersion coefficient corresponding to the walk-off effect between spectral channels,  $\Delta\omega$  is the channel spacing,  $\beta_2$  represents the second-order dispersion,  $\gamma' = \gamma L_{eff}$ , where  $\gamma$  is the Kerr coefficient, and  $L_{eff}$  is the effective length  $L_{eff} = (1 - e^{-\alpha L})/\alpha$  that accounts for averaging over periodic loss and gain,  $L$  is the amplifier span length

and  $\alpha$  is the fiber loss coefficient. It should be noted, that the neural network architecture based on coupled NLSEs have been proposed in [20] for single channel processing of optical signals. We use this model as the basis of the proposed scheme because it allows us to process multiple channels at low sampling in parallel. Note, that DCNN designed this way accounts for self-phase modulation (SPM) and cross-phase modulation (XPM) between spectral channels, but not four-wave mixing (FWM).

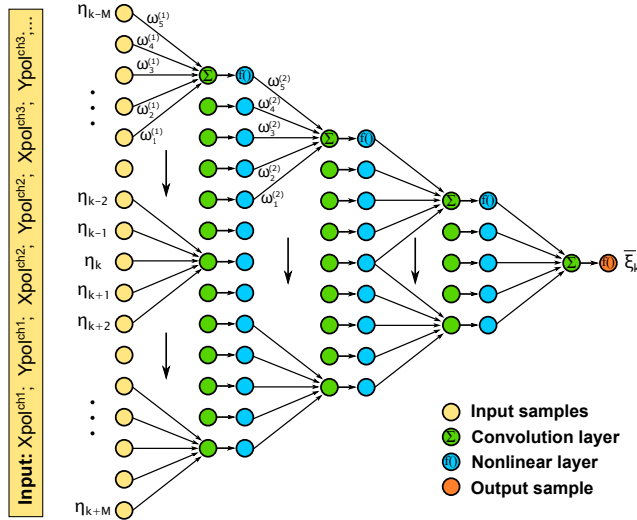


Fig. 1. Deep convolutional neural network architecture.

It is common practice for NN design that all involved quantities such as network input/output and layer weights are real-valued rather than complex valued. However, since the propagation of a telecommunication signal in an optical fiber is described by the evolution of a complex field envelope and the constellation symbols are also complex, it is worthwhile to implement the NN with complex-valued arithmetic. We implemented complex numbers and certain arithmetic operations presenting input complex data as pairs of real numbers corresponding to its real and imaginary parts. In addition, the individual polarizations and WDM channels are processed in parallel, and we considered them as additional “feature columns” in the data array. Thus, input sequences of complex symbols of size  $L$  for  $N_{ch}$  spectral channels and both polarizations are represented as a real-valued array of size  $(L, 4 \cdot N_{ch})$ .

The implementation of the proposed DCNN is performed in MXNet using the Adam optimizer [21] with adaptive learning rate. Mean squared error between the transmitted and recovered symbols is used as a loss function. We average the resulting error over all spectral channels and polarizations.

Aiming at the reduced complexity we consider a DCNN architecture with each layer corresponding to one span propagation and the input vector sampled at 1 sample per symbol.

### A. Recovering of Signal Dispersion Broadening

The DBP method simulates a signal propagation through a fiber in the reversed direction. Therefore, a signal with

accumulated dispersion, corresponding to the entire length of the fiber, is used as the input of the approach. Similarly, samples of the received signal are used as input data for the developed DCNN. Moreover, the signal should be sampled at 1 SpS. In our study, we transmit root raised cosine (RRC) pulses with a roll-off factor of 0.1, and therefore the signal bandwidth is wider than  $1/T$ . As a result, if we downsample the received signal to 1 SpS directly, we will lose some of the useful information. To avoid this we perform the following procedure: first compensate accumulated chromatic dispersion for the received signal downsampled to 2 SpS, next downsampling to single sample per symbol takes place, and then we recover the signal dispersion broadening by the inverse procedure of the accumulated CD compensation in the frequency domain. This approach allows us to correctly take into account the interplay of linear and nonlinear effects for signals with 1 sample per symbol.

### B. Convolution Layers for Chromatic Dispersion Compensation

Although each linear step of the DBP method can be implemented either in time [22] or in frequency domain [7], a time domain implementation using finite impulse response (FIR) filters [22], [23] is more efficient in real time applications. Furthermore, it is consistent with the one-dimensional (1D) convolution operation which can be equivalently executed by the linear layer of a convolutional neural network. In our case each 1D convolution layer undertakes to compensate an equal part of chromatic dispersion, although a non-uniform compensation scheme can be also applied.

Neural network training is a rather long and complex procedure. Nevertheless, we can improve its efficiency and achieve fast convergence by creating favourable initial conditions using any preliminary knowledge about our problem, e.g. we can initialize the layer weights with the coefficients of an equivalent FIR-based CD compensation (CDC) filters [7], [24]. Specifically in our case, the weights of the convolution layer were trained first to adapt itself as one of the DBP linear steps. Such approach allows to obtain CD filters of small length with acceptable accuracy compared to filters based on frequency-domain sampling [16]. The resulting convolutional filters with 61 and 151 coefficients are depicted in the insets of Fig. 3(a).

To initialize the weights of the entire neural network, and thus compensate for the total accumulated dispersion of the link, one could replicate the previously identified coefficients for the remaining convolution layers of the NN architecture. However, the repeated use of single span optimized coefficients will likely lead to sub-optimal performance [16]. Therefore, instead, before training the deep NN simulating DBP, a joint weight optimization for the convolution layer cascade was performed. The training was achieved by omitting the in-between nonlinear activation functions and initializing each linear layer of the cascade with the single-step weight solution identified by the above process. During the joint optimization process, we require that, in addition to the filter sequence compensating for the entire accumulated CD, each filter should still compensate for the corresponding part of the dispersion

similar to [16]. The joint optimization was implemented as follows: the first layer should compensate chromatic dispersion of one span, at the same time the first two layers compensate CD corresponding for two spans, the first three convolutional layers used for CD compensation of three spans, and so on. The application of a joint filter optimization procedure can significantly reduce the resulting error in compensating chromatic dispersion and reduce substantially the training time. To show this we compare different techniques of pre-termination of convolution layers, including: (i) the proposed optimization process described above; (ii) joint optimization of all convolutional layers, but without additional requirement for individual filters; (iii) initialization of all filters with optimized coefficients for single span (without joint optimization) and (iv) initialization of all convolutional layers with random values.

TABLE I  
COMPARISON OF CONVOLUTIONAL LAYER OPTIMIZATION TECHNIQUES

Optimization technique	Achieved $Q^2$ -factor	Number of epochs
Layer-by-layer (i)	14.48	1570
Joint optimization (ii)	14.45	2180
Single step (iii)	14.37	2510
Random (iv)	12.86	10000*

Table I shows the achieved  $Q^2$ -factor and number of performed epochs for the considered techniques after training the entire DCNN when processing single channel with an input power of 2 dBm. By epoch, we mean one pass of the entire data set through the neural network. It can be seen that all methods except Random provide a similar level of performance. At the same time, the method proposed in this work requires much less epochs, and in all cases, the joint optimization process takes little time. It should be noted that in the case of random optimization we stopped the training process when 10000 epochs were reached.

With the propagation of WDM signals, the chromatic dispersion leads to a group delay difference between spectral channels [19]. Therefore, a neural network processed multiply channels simultaneously should take this into account in the architecture design. In the proposed DCNN each channel and polarization are processed separately in a linear layer. We assume here that signals at each channel propagate at the carrier frequency and after the linear step group delay corresponding to channel frequency and distance propagated is compensated. To account for the group delay difference, a real-valued fractional delay (FD) filter for each spectral channel can be used after convolutional layer at each linear step [25], [26]. Furthermore, to reduce the computational complexity, we set the step length so that the time delay for each channel is divisible by the duration of the symbol interval  $T$ , as it was suggested in [20]. Thereby, we can shift the resulting data array by the corresponding number of symbols to compensate for the walk-off between the channels. It should be noted that in general it is necessary to use additional FD filter after the last step, since the steps chosen by the method described above may not completely cover the full transmission length [20].

Simple physical considerations [7], [24] show that the CD compensating FIR filters are symmetrical. Therefore, to reduce the computational complexity we also required that linear convolution layers are symmetric. This requirement slightly degrades algorithm performance, but almost halves the complexity.

### C. Activation Function for Fiber Nonlinearity Compensation

Selection of the appropriate nonlinear activation function is an important design issue in the NN design. Its main task is to create advanced mappings between the network's inputs and outputs, which are essential for the processing of complex data. On the other hand, in fiber-optic communications the Kerr nonlinearity has a well defined form, creating for each sub-step of the DBP method the following transfer function :

$$A(z+h) = e^{-i\gamma_{DBP}h|A(z)|^2} A(z), \quad (2)$$

where  $\gamma_{DBP}$  is the effective (that includes losses through the  $L_{eff}$ ) fiber nonlinear parameter and  $h$  is the sub-step length. As mentioned earlier, when designing a neural network it is beneficial to make use of any pre-existing knowledge of the underlying physical effects it is asked to address. Therefore, it is natural to consider the activation function mimicking the nonlinear DBP step. There are many approaches that focus on reducing complexity and improving performance of the DBP method, for instance, the enhanced split-step Fourier method [17], [27], where neighboring samples are also used at a nonlinear step. In this case, the nonlinear sub-step of the DBP method (2) can be rewritten as:

$$A_k(z+h) = e^{-i\gamma_{DBP}h \sum_{j=-R_l}^{R_r} \alpha_j |A_{k+j}(z)|^2} A_k(z), \quad (3)$$

where  $A_k(z) \equiv A(z, t = k/f)$ ,  $f$  - sampling rate,  $R = R_l + R_r + 1$  is a number of included symbols and  $\alpha_j$  are real-valued coefficients. In a similar manner we introduce a nonlinear activation function for the neural network as follows:

$$f(z_k) = e^{-i \sum_{j=-R_l}^{R_r} \alpha_j |z_{k+j}|^2} z_k, \quad (4)$$

where  $\alpha_j$  are real-valued trainable weights.

Note that the sum in the exponent is similar to the formula for an one-dimensional convolution layer with the real-valued coefficients. Thus, the implementation of the enhanced SSFM in a nonlinear sub-step of a deep convolutional neural network is possible by using a convolution layer with filter  $\{\alpha_j\}$  to the squared symbol modules  $|z|^2$  and then calculating the nonlinear Kerr activation function.

### D. Second Polarization and Neighboring Spectral Channels Accounting

Considering the interaction between the polarizations and neighboring spectral channels, we took into account only cross-phase modulation effects as described in the propagation equations (1). Then using the enhanced SSFM the nonlinear activation function has the form:

$$f(z_k^{x/y,c}) = e^{-i\Phi_k^{x/y,c}} z_k^{x/y,c}, \quad (5)$$

$$\Phi_k^{x/y,c} = \sum_{s=1}^{N_{ch}} \sum_{i=-R_{s,l}}^{R_{s,r}} \alpha_{s,i}^c (|z_{k+i}^{x,s}|^2 + |z_{k+i}^{y,s}|^2),$$

where  $z^{x/y,c}$  are data from the x- or y-polarization and  $c$ th channel and  $N_{ch}$  is the number of the processed spectral channels. It should be noted that we use the same weights for both polarization of each WDM-channel. At the same time, we divide the coefficients  $\alpha_{s,i}^c$  into groups depending on the distance between the spectral channels  $c$  and  $s$ . For example, if  $c = s$  the coefficients  $\alpha_{s,i}^c$  correspond to SPM effects and we refer the real-valued convolution layer determined by these weights as SPM filter. Accordingly, coefficients with  $c \neq s$  correspond to XPM effects and by analogy with SPM filters we call it XPM-k filters for the spectral channels spaced at  $k$  channel spacing. Such a division can be justified if we turn to the inset of Fig. 3(b), where coefficients for SPM, XPM-1 and XPM-2 filters of the central spectral channel with a width of 41 obtained after training DCNN processed 5 neighboring channels are presented. It should be noted that by XPM-k we denote a set of the filters, so for the central channel there are one XPM-1 and one XPM-2 filters corresponding to the left adjacent channels (XPM-1(+)) and XPM-2(+)) in the inset) and filters XPM-1(-) and XPM-2(-) for the right neighbors. Filters XPM-k(+) and XPM-k(-) have the similar shape but with reversed coefficients. We can see that depending on the desired scale of the coefficients taken into account, we can use SPM and XPM filters of different widths. Moreover, different numbers of left and right neighboring symbols can be used. We also can see that the resulting SPM filter is symmetrical. So by analogy with linear convolution layers to reduce complexity we also required that these filters are symmetric during the training.

### E. Complexity analysis

The estimate of computational complexity is performed in terms of the number of real multiplications (RMs) per transmitted symbol, the addition operations are not taken into account. In this work, we investigate DCNN that process single channel or five central channels from WDM channel grid and the complexity analysis will be devoted to these cases.

Let us first estimate the complexity of a one-dimensional real convolution layer. Such a layer of width  $S$  can be described by the formula:

$$y_i = \sum_{k=-(S-1)/2}^{(S-1)/2} w_k \cdot x_{i+k}, \quad (6)$$

and, therefore, requires  $S$  real multiplications. Complex form of a convolution can be written as

$$w = (d * z), \quad w_n = \sum_k d_k z_{n-k}, \quad (7)$$

where  $d = a + ib$ ,  $z = x + iy$ ,  $w = u + iv$  are complex numbers. If we split a result into real and imaginary parts

$$\begin{aligned} u_n &= \sum_k a_k x_{n-k} - \sum_k b_k y_{n-k}, & u &= (a * x) - (b * y), \\ v_n &= \sum_k b_k x_{n-k} + \sum_k a_k y_{n-k}, & v &= (b * x) + (a * y), \end{aligned} \quad (8)$$

it can be seen that single complex-valued convolution can be realised using four real-valued convolution and two addition operations. As a result, complex convolution layer of size  $S$  requires  $4 \cdot S$  real multiplications per transmitted symbol. It should be noted that since we use symmetric filters for linear layers, before the complex convolution we can add together the corresponding left and right symbols, thereby reducing the required number of real multiplications. Then the computational complexity of the complex convolution layer is  $2 \cdot (S + 1)$ .

Implementation of the nonlinear activation function (2) in complex-valued arithmetic is straightforward and can be performed as follows:

$$\begin{aligned} f^{Re}(z) &= \cos(\gamma_{DBP} P) \cdot x + \sin(\gamma_{DBP} P) \cdot y, \\ f^{Im}(z) &= \sin(\gamma_{DBP} P) \cdot x - \cos(\gamma_{DBP} P) \cdot y, \\ P &= |x|^2 + |y|^2. \end{aligned} \quad (9)$$

So the implementation of the Kerr function requires a calculation of the cosine and sine functions and 7 real multiplications. We assume here that  $\cos$  and  $\sin$  functions are defined by pre-computed tables, thus no additional multiplications are required for their calculation.

For the enhanced activation function after calculating of squared data modules (2 real multiplications per symbol) we need to apply one-dimensional real-valued convolution layer of size  $R$  that requires  $R$  real multiplications. It should be noted that since we use the same coefficients for different polarization, we actually perform convolution once for both polarizations. Therefore, a term with the number of multiplications required by such convolution is included to the complexity formula with a coefficient of 0.5. Next, implementation of the activation function in complex-valued arithmetic requires an additional 4 real multiplications in accordance with the formula (9). So one nonlinear activation function layer in the case of one channel requires  $NL_1 = 0.25 \cdot (R + 1) + 6$  RMs per transmitted symbol. This formula already takes into account the symmetry of SPM filter by analogy with linear convolution layers.

In the case of 5 channels, in total we use 5 SPM filters, 8 XPM filters for the closest adjacent spectral channels (XPM-1), 6 XPM filters for the channels spaced at two channel spacing (XPM-2), 4 XPM-3 filters and 2 XPM-4 filters. So one nonlinear activation function layer in case of 5 channels requires a total  $NL_5 = 0.25 \cdot (R + 1) + 0.8 \cdot R_1 + 0.6 \cdot R_2 + 0.4 \cdot R_3 + 0.2 \cdot R_4 + 6$  RMs per transmitted symbol, where  $R_k$  are width of XPM-k filters.

Let us consider a deep neural network used to process  $N_{ch}$  spectral channels with both polarization components, and consisting of  $N_s$  layers, and suppose that the first output

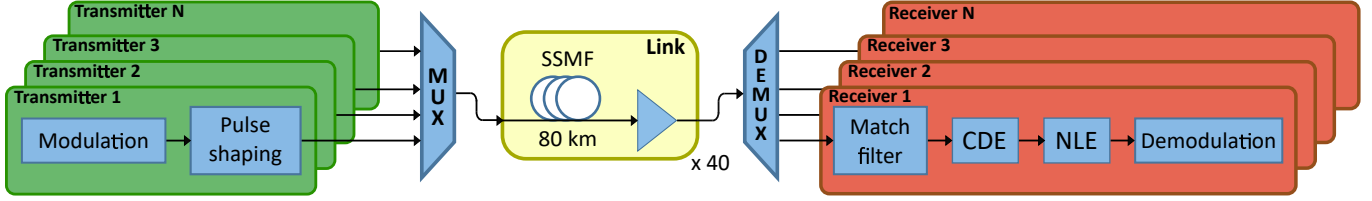


Fig. 2. Scheme of simulated transmission link.

symbol has already been calculated. Then, when calculating the second and subsequent DCNN output symbols, most of the necessary coefficients will be already calculated. In this case we need to compute only once a complex convolution and once an enhanced nonlinear activation function on each convolution layer. Thus, the total computational complexity of the proposed deep convolutional NN for the second and subsequent symbols is

$$C_{DCNN} = N_s \cdot (NL_i + 2 \cdot (S + 1)) + n \cdot 4 \cdot S_{FD}, \quad (10)$$

where  $i = 1$  or  $5$  depending on the number of processed channels,  $n = 1$  if the FD filter with  $S_{FD}$  coefficients is used and  $n = 0$  otherwise.

Thus, provided that the first output symbol has been calculated in advance or that a large number of symbols have been processed, that the complexity of first symbol computing becomes insignificant, we can use the expression (10) to estimate the required number of real multiplications per transmitted symbol for the entire deep convolutional neural network.

For an accurate comparison with other methods, we also take into account the computational complexity of the chromatic dispersion equalization (CDE) block. It includes the chromatic dispersion compensation and the recovering of the signal dispersion broadening, which is actually the same as the CDC, but in the opposite direction. So this block corresponds to two linear steps of the DBP method with 2 and 1 samples per symbol, respectively, and its computational complexity in terms of number of real multiplications per transmitted symbol is [15]:

$$C_{CDE} = 4 \cdot \left( \frac{2N(\log_2 N + 1)}{(N - N_{D_2} + 1)} + \frac{N(\log_2 N + 1)}{(N - N_{D_1} + 1)} \right), \quad (11)$$

where  $N$  is the FFT size and  $N_{D_q} = q\tau_D/T$ , where  $\tau_D$  corresponds to the dispersive channel impulse response. The factor 4 in the expression corresponds to the fact that one complex multiplication can be expressed through 4 real ones. We optimized FFT size  $N$  to get the minimum computational complexity. Finally, the complexity of the overall deep convolutional neural network equalization scheme can be calculated as the sum of  $C_{DCNN}$  and  $C_{CDE}$ .

We compare the performance of the proposed scheme with the digital back-propagation method processed one or five spectral channels. In the case of single channel the computational complexity of the DBP method in terms of the number of required real multiplications per transmitted symbol can be estimated as [15]:

$$C_{DBP-1ch} = 4N_{Sp}N_{StpSp} \left( \frac{N(\log_2 N + 1)q}{(N - N_{D_q} + 1)} + q \right), \quad (12)$$

where  $N_{Sp}$  is the total number of spans,  $N_{StpSp}$  is the number of propagation steps per span and  $q$  is the oversampling factor.

In the case of five WDM channels transmission, we consider the DBP method based on coupled NLSEs (1) similar to DCNN. It allows to compensate only SPM and XPM effects, but we can use a small number of samples per symbol to reduce the computational complexity. In this case, the linear step is the same as in the case of single channel DBP, and therefore it requires the same number of real multiplications per transmitted symbol. The nonlinear step has the following form:

$$A_c^{x/y}(z+h) = e^{-i\frac{8\gamma h}{9}(|A_c^x|^2 + |A_c^y|^2 + 2\sum_{s \neq c}(|A_s^x|^2 + |A_s^y|^2))} A_c^{x/y}(z). \quad (13)$$

For the single channel DBP it is assumed that the value of the nonlinear phase shift can be obtained using a lookup table [15], [28], and then nonlinear step (2) requires single complex multiplication per sample. In the case of 5-channel DBP, first we need to calculate the optical intensity for each channel and polarization that requires 2 RMs per sample and after summing in the exponent, we can also use the lookup table to obtain the phase shift. So, to calculate the nonlinear step we need to perform 2 real and 1 complex multiplication per sample or 6 RMs in total. Thus, the computational complexity for 5-channel DBP in terms of the number of required real multiplications per transmitted symbol can be estimated as

$$C_{DBP-5ch} = N_{Sp}N_{StpSp} \left( 4 \frac{N(\log_2 N + 1)q}{(N - N_{D_q} + 1)} + 6q \right). \quad (14)$$

### III. TRANSMISSION SYSTEM MODEL

The simulated transmission link is depicted in Fig. 2. Transmission of 11 WDM channels with polarization multiplexing has been studied. Each channel transmitter generates 16-QAM modulated root raised cosine pulses at symbol rate 64 GBaud, resulting in 512 Gb/s channel rate that includes 28% forward-error-correction overhead, making net information rate 400 Gb/s per channel. A Gray-coded constellation diagram, a roll-off factor of 0.1 and an oversampling factor of 32 has been used in the numerical modelling. The frequency spacing



between the channels was 75 GHz. The central wavelength of the emitted signal band was located  $\lambda = 1550$  nm. All system and signal parameters used in the modelling are summarized in Table II.

TABLE II  
TRANSMISSION MODEL PARAMETERS

Parameter	Value	Parameter	Value
Attenuation	0.2 dB/km	COI Wavelength	1550 nm
Dispersion	17 ps/nm/km	RRC Roll-off	0.1
Nonlinearity	1.2 1/W/km	Symbol Rate	64 GBaud
Distance	40×80 km	Channel Spacing	75 GHz
Noise Figure	4.5 dB	Channel Data Rate*	512 Gbit/s

\*including 28% FEC overhead

The generated signal is subsequently launched into a transmission link that consisted of 40 spans of 80 km single mode fiber each making total propagation distance of 3200 km. A standard EDFA with a 4.5 dB noise figure compensates the losses of each span. Signal propagation is modelled by the Manakov equations [29]:

$$\frac{\partial A_{x/y}}{\partial z} = \left[ -\frac{\alpha}{2} - i\frac{\beta_2}{2} \frac{\partial^2}{\partial t^2} + i\gamma \frac{8}{9} (|A_x|^2 + |A_y|^2) \right] A_{x/y}. \quad (15)$$

The propagation equations have been solved using a standard second-order symmetrical split-step Fourier method [18]. We didn't include polarization-mode dispersion (PMD) and principal states of polarization rotations caused by fiber birefringence in our simulation. To take into account these effects, real-valued FD filters and trainable 2x2 rotation matrix on each layer can be used as proposed in [25].

After transmission, the optical signal is coherently detected. Each channel is demultiplexed with a root raised cosine matched filter of the same roll-off factor as at the transmitter. Then a chromatic dispersion equalization stage is used. It is described in detail in II-A and consists of downsampling to 2 SpS, chromatic dispersion compensation, down-conversion to single sample per symbol and recovering of CD broadening. Next, the nonlinear equalization (NLE) is applied by means of a deep convolutional neural network. We use  $2^{21}$  16-QAM symbols to train DCNN and  $2^{17}$  symbols for testing (the same number of symbols is used to calculate BER in the case of CDC and DBP equalizations). Mini-batch size is  $2^{17}$  symbols. Discrete distribution generator with random seed from MKL in C++ is used to generate transmission data. For convolutional layer weight initialization before joint optimization we used MXNet normal initializer with  $\sigma = 0.05$ . All nonlinear filters are initialized with a zero vector of appropriate length with 1 in the center. The learning rate is initially set at  $2 \cdot 10^{-4}$  and it is halved if the losses don't decrease for 100 epochs in a row.

For comparison purpose, we also consider the nonlinear equalizers based on the DBP method. In this case, after downsampling to 2 SpS DBP for central channel for SPM compensation [7] or DBP for 5 channels based on coupled nonlinear NLSEs [18] are applied. It should be noted that

for the DBP method we numerically optimized nonlinear parameter, because its value depends on the dispersion map, number of propagation steps and launched power [7]. At the next step, we compensate for the remaining nonlinear phase shift of all symbols (joint phase rotation in the complex plane) using the least mean square (LMS) algorithm. After nonlinear equalization step we apply the demodulation and calculate bit error rate (BER) for the central channel of interest (COI).

#### IV. NUMERICAL RESULTS

The first step of our study was to investigate the influence of the main characteristics of the proposed NLE scheme on the efficiency of nonlinearity compensation. We start out analysis by considering the 40-layers (1 layer per span) deep convolutional NN processing central channel with both polarization and signal transmission with launch power of 2 dBm.

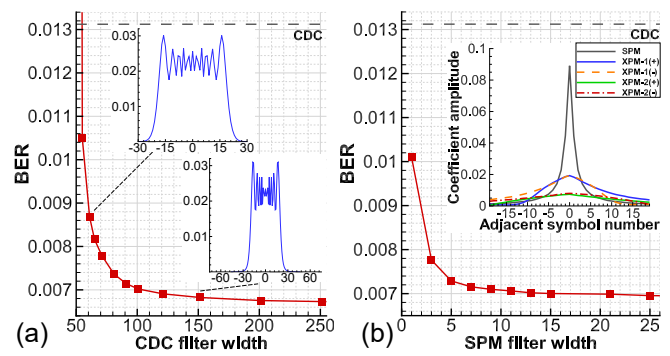


Fig. 3. BER as a function of convolutional CDC filter width (a) and “nonlinear” SPM filter width (b). Insets: (a) trained linear filters with 61 and 151 coefficients, (b) trained SPM, XPM-1 and XPM-2 filter coefficients.

We optimize the filter width from linear and nonlinear layers to find a trade-off between the DCNN performance and computational complexity. Fig. 3(a) shows BER as a function of CDC filter width for deep convolutional NN with 13 coefficients for SPM filter on each nonlinear step. As we can see, a neural network with linear filters width less than 50 coefficients cannot effectively compensate for the CD and the resulting BER level is higher than one-step CD compensation in frequency domain (dashed line “CDC” in the figure). On the other hand, application of filters with a width more than 100 coefficients just slightly increases the performance. The resulting lower bound close to the analytical estimation of the 1-span channel memory [30]:

$$M_{CD} = 2\pi|\beta_2|L_{Sp}B/T \approx 50, \quad (16)$$

where  $L_{Sp}$  is the span length and  $B$  is the signal bandwidth. However, it was shown [16] that the required CDC filter width is significantly larger than predicted by (16).

Fig. 3(b) shows BER as a function of “nonlinear” SPM filter width for DCNN with fixed CDC linear filter width on each layer (101 coefficients). The leftmost point on the figure corresponds to the case of the conventional DBP, when no information about neighboring symbols is used in nonlinear steps (1 coefficient width). It can be seen that using even one neighboring symbol on each side (3 coefficients width) allows

us to reduce BER by 23% compared to neural network without enhanced SSFM. Usage of SPM filters with more than 10 coefficients leads only to a slight performance improvement.

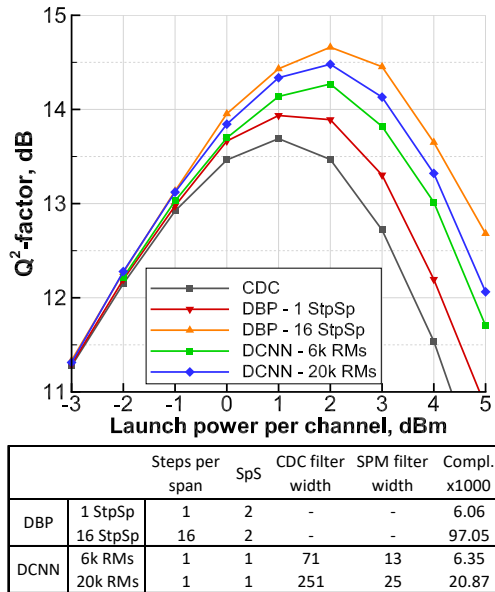


Fig. 4.  $Q^2$ -factor (recalculated from the directly counted BER) as a function of launch power per channel for NLEs processed single channel.

To evaluate the efficiency of the proposed scheme we compared it with a linear compensator and DBP with different number of steps per span. We determine BER using direct error counting and then recalculate  $Q^2$ -factor from BER using standard approach [31]:

$$Q^2 \text{ [dB]} = 20 \log_{10} \left[ \sqrt{10} \operatorname{erfc}^{-1} \left( \frac{8\text{BER}}{3} \right) \right]. \quad (17)$$

Fig. 4 shows  $Q^2$ -factor for COI as a function of launch power per channel for different configurations of NLE algorithms. As expected, the system with the linear compensator (grey line) shows the worst performing. Red line corresponds to the digital back-propagation method with 2 samples per symbol and 1 step per span (DBP - 1 StpSp) and it requires approximately 6000 RMs per transmitted symbols. For comparison we also consider DCNN with an architecture designed to have the same computational complexity (DCNN - 6k RMs). The main parameters and the complexity of the considered NLEs can be found in the table at the bottom of the figure. In this case the proposed scheme overtakes the DBP method by 0.31 dB. We are also interested in the best performance improvement achievable with these NLEs. Orange line corresponds to the DBP method with 2 SpS and 16 steps per span and a further increase in the number of steps does not lead to a significant performance improvement. The best performance obtained by DCNN is indicated by the blue line and it requires 20874 RMs per transmitted symbol. The best achieved  $Q^2$ -factor for deep convolutional NN processed single channel exceeds linear equalization performance by 0.82 dB and it is 0.15 dB lower than the best  $Q^2$ -factor for the single channel DBP with 2 SpS. It should be noted that in

this case DCNN has a computational complexity of almost 5 times less than the DBP method.

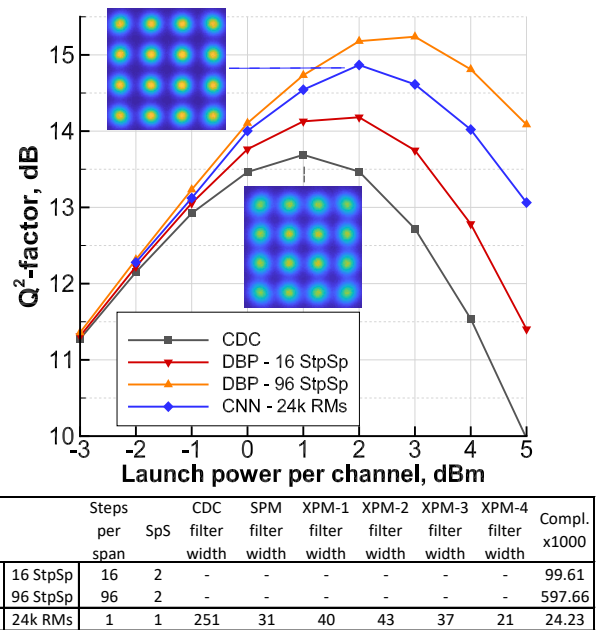


Fig. 5.  $Q^2$ -factor (recalculated from the directly counted BER) for the channel of interest as a function of launch power per channel for NLEs processed five spectral channels.

We also considered the deep convolutional NN that processed 5 WDM channels simultaneously and compared it with the DBP method for 5 channels based on coupled NLSEs (1) with different number of steps per span. Fig. 5 shows  $Q^2$ -factor for COI as a function of launch power per channel for different configurations of NLE algorithms. Red line corresponds to multi-channel DBP method with 2 SpS and 16 steps per span. It should be noted that multi-channel DBP with fewer steps per span shows the same performance or lower than the single channel DBP with 1 step per span, with significantly greater computational complexity. The best performance obtained by DCNN is indicated by the blue line and it requires 24234 RMs per transmitted symbol. Its parameters can be found in the table at the bottom of the figure. Orange line corresponds to the 5-channel DBP method based on coupled NLSEs with the best performance improvement. It has 96 steps per span and a further increase in the number of steps does not lead to a significant performance improvement. The proposed scheme overtakes the linear compensator by 1.2 dB and multi-channel DBP with 16 step per span by 0.7 dB. It shows the performance improvement lower by 0.36 dB compared to the best  $Q^2$ -factor achieved by 5-channel DBP method, but in the same time, DCNN has significantly less computational complexity. The received constellation diagrams, for the cases of linear compensator and DCNN based equalization taken at the point of optimum launched power, are shown in the inset of Fig. 5.

Subsequently we compared the computational complexity of the proposed NLE scheme based on deep convolutional NN with the DBP method based on coupled NLSEs in case of processing single and five spectral channels. Fig. 6 shows



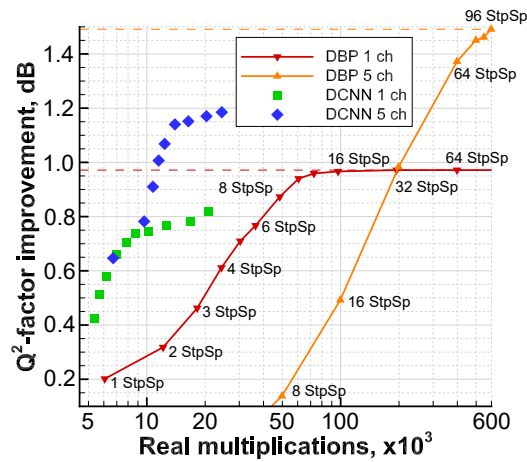


Fig. 6.  $Q^2$ -factor improvement (recalculated from the directly counted BER) for COI as a function of the number of required real multiplications per transmitted symbol.

the achieved  $Q^2$ -factor improvement in comparison with the linear compensator for DCNN and DBP in terms of required number of real multiplications per transmitted symbol. For DBP the number of steps per span varied and we considered DCNN with a different number of coefficients on linear and nonlinear layers. Dotted lines indicate maximum performance improvement achieved using the DBP method for single (red line) and five (orange line) spectral channels. As we can see, in all cases, DCNN-based equalizers show a larger performance improvement compared to the DBP method with the same complexity. It should be noted that when comparing with a single channel DBP equalizer, DCNN for 1 channel shows up to 0.45 dB higher  $Q^2$ -factor improvement with the same complexity, while 5-channel DCNN scheme achieve up to 0.75 dB higher performance improvement. Moreover, deep convolutional neural network processed five WDM channels shows better performance with lower computational complexity compared to the maximum achieved  $Q^2$ -factor for single channel DBP.

## V. CONCLUSION

We studied application of the convolutional neural networks for compensating nonlinear distortions in a long-haul ultra-high capacity fiber-optic transmission system. The introduced DCNN architectures mimics the traditional DBP algorithm by using each linear convolutional layer to compensate for the chromatic dispersion on a subsection of the link and the nonlinear activation layer to cancel the corresponding Kerr-effect induced nonlinearity. As a further development of the previously studied DNN-based compensation schemes [16], [20], we customize the nonlinear activation function to account for a different number of neighboring symbols from adjacent spectral channels, enabling to suppress a large portion of the XPM introduced signal distortions with low computational complexity. Furthermore, to achieve fast training and secure convergence in the optimum operating point a thoughtful 2-stage weight initialization scheme was applied by identifying the sub-optimal values for the single layer and then performing

joint optimization of the weights when cascading linearly all the convolutional layers of the DCNN architecture.

Through a detailed complexity analysis the number of real multiplications has been identified as a function of the dimensions of the architecture. In addition we examined the performance of the proposed scheme in a 3200 km 11x400-Gb/s RRC WDM PDM-16QAM transmission system when equalization was applied separately on a per channel basis, or the nonlinear equalizer was compensating simultaneously 5 neighboring channels. The results showed that our scheme exceeded the performance of the linear equalizer by 0.8 dB in the single channel scenario and by 1.2 dB in the multi-channel case. When comparing with a DBP equalizer of the same complexity, in the single channel compensation case, the DCNN scheme achieves up to 0.45 dB higher  $Q^2$ -factor improvement performance. More impressive are the results in the multi-channel equalization case, in which the complexity of the DCNN remains almost unchanged, whereas the DBP schemes require significantly a higher number of real multiplications per transmitted symbol. Our simulations show that the suggested DCNN based equalizer can have 4 times less complexity than a multi-channel DBP based scheme and still achieve 0.7dB more improved  $Q^2$ -factor performance. Our results show clearly the great potential of our proposed equalization method in extending the capacity of future transmission systems.

## ACKNOWLEDGMENT

The work was supported by the Russian Science Foundation (Grant No. 17-72-30006). SS and SKT acknowledges support by the EPSRC Programme grant TRANSNET (EP/R035342/1).

## REFERENCES

- [1] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662–701, Feb. 2010.
- [2] A. Ellis and M. Sorokina, *Optical Communication Systems: Limits and Possibilities*. Jenny Stanford Publishing, 2019.
- [3] O. Vassilieva, I. Kim, and T. Ikeuchi, "Enabling technologies for fiber nonlinearity mitigation in high capacity transmission systems," *J. Lightwave Technol.*, vol. 37, no. 1, pp. 50–60, Jan. 2019.
- [4] E. Agrell *et al.*, "Roadmap of optical communications," *J. Opt.*, vol. 18, no. 6, p. 063002, Jun. 2016.
- [5] S. K. Turitsyn, I. S. Chekhovskoy, and M. P. Fedoruk, "Nonlinear Fourier transform for characterization of the coherent structures in optical microresonators," *Opt. Lett.*, vol. 45, no. 11, pp. 3059–3062, Jun. 2020.
- [6] S. K. Turitsyn *et al.*, "Nonlinear fourier transform for optical data processing and transmission: Advances and perspectives," *Optica*, vol. 4, no. 3, pp. 307–322, Mar. 2017.
- [7] E. Ip and J. M. Kahn, "Compensation of dispersion and nonlinear impairments using digital backpropagation," *J. Lightw. Technol.*, vol. 26, no. 20, pp. 3416–3425, Oct. 2008.
- [8] E. Ip, "Nonlinear compensation using backpropagation for polarization-multiplexed transmission," *J. Lightw. Technol.*, vol. 28, no. 6, pp. 939–951, Mar. 2010.
- [9] H. J. Kelley, "Gradient theory of optimal flight paths," *ARS Journal*, vol. 30, no. 10, pp. 947–954, Oct. 1960.
- [10] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [11] F. Musumeci *et al.*, "An overview on application of machine learning techniques in optical networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1383–1408, Jun. 2019.

- [12] F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An optical communication's perspective on machine learning and its applications," *J. Lightw. Technol.*, vol. 37, no. 2, pp. 493–516, Jan. 2019.
- [13] J. Mata *et al.*, "Artificial intelligence (AI) methods in optical networks: A comprehensive survey," *Opt. Switching Netw.*, vol. 28, pp. 43–57, Apr. 2018.
- [14] S. Zhang *et al.*, "Field and lab experimental demonstration of nonlinear impairment compensation using neural networks," *Nat. Commun.*, vol. 10, no. 1, p. 3033, Dec. 2019.
- [15] A. Napoli *et al.*, "Reduced complexity digital back-propagation methods for optical communication systems," *J. Lightw. Technol.*, vol. 32, no. 7, pp. 1351–1362, Apr. 2014.
- [16] C. Hager and H. D. Pfister, "Deep Learning of the Nonlinear Schrödinger Equation in Fiber-Optic Communications," in *Proc. IEEE Int. Symp. Information Theory (ISIT)*. Vail, CO, Jun. 2018, pp. 1590–1594.
- [17] M. Secondini, D. Marsella, and E. Forestieri, "Enhanced split-step Fourier method for digital backpropagation," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Sep. 2014, p. We.3.3.5.
- [18] G. Agrawal, *Nonlinear Fiber Optics*. Elsevier, 2013.
- [19] E. F. Mateo, F. Yaman, and G. Li, "Efficient compensation of inter-channel nonlinear effects via digital backward propagation in WDM optical transmission," *Opt. Express*, vol. 18, no. 14, pp. 15 144–15 154, Jul. 2010.
- [20] C. Häger and H. D. Pfister, "Wideband Time-Domain Digital Back-propagation via Subband Processing and Deep Learning," in *European Conference on Optical Communication, ECOC*, Sep. 2018, p. Tu4F.4.
- [21] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, Dec. 2014, pp. 1–15.
- [22] C. Fougstedt, A. Sheikh, P. Johannisson, A. G. i. Amat, and P. Larsson-Edefors, "Power-Efficient Time-Domain Dispersion Compensation Using Optimized FIR Filter Implementation," in *Proc. Signal Processing in Photonic Communications*, 2015, p. SpT3D.3.
- [23] C. Fougstedt, M. Mazur, L. Svensson, H. Eliasson, M. Karlsson, and P. Larsson-Edefors, "Time-Domain Digital Back Propagation: Algorithm and Finite-Precision Implementation Aspects," in *Proc. Opt. Fiber Commun. Conf.*, 2017, p. W1G.4.
- [24] A. Eghbali, H. Johannsson, O. Gustafsson, and S. J. Savory, "Optimal Least-Squares FIR Digital Filters for Compensation of Chromatic Dispersion in Digital Coherent Optical Receivers," *J. Lightw. Technol.*, vol. 32, no. 8, pp. 1449–1456, Apr. 2014.
- [25] C. Häger, H. D. Pfister, R. M. Büttler, G. Liga, and A. Alvarado, "Model-based machine learning for joint digital backpropagation and PMD compensation," in *Proc. Opt. Fiber Commun. Conf.*, 2020, p. W3D.3.
- [26] V. Oliari, S. Goossens, C. Hager, G. Liga, R. M. Butler, M. van den Hout, S. van der Heides, H. D. Pfister, C. Okonkwo, and A. Alvarado, "Revisiting Efficient Multi-Step Nonlinearity Compensation With Machine Learning: An Experimental Demonstration," *Journal of Lightwave Technology*, vol. 38, no. 12, pp. 3114–3124, Jun. 2020.
- [27] M. Secondini, S. Rommel, F. Fresi, E. Forestieri, G. Meloni, and L. Poti, "Coherent 100G nonlinear compensation with single-step digital backpropagation," in *Proc. Int. Conf. Opt. Netw. Design and Modeling (ONDM)*, May 2015, pp. 63–67.
- [28] L. Zhu and G. Li, "Nonlinearity compensation using dispersion-folded digital backward propagation," *Opt. Express*, vol. 20, no. 13, pp. 14 362–14 370, Jun. 2012.
- [29] D. Marcuse, C. Manyuk, and P. Wai, "Application of the manakov-pmd equation to studies of signal propagation in optical fibers with randomly varying birefringence," *J. Lightw. Technol.*, vol. 15, no. 9, pp. 1735–1746, Sep. 1997.
- [30] S. J. Savory, "Digital filters for coherent optical receivers," *Opt. Express*, vol. 16, no. 2, pp. 804–817, Nov. 2008.
- [31] W. Freude *et al.*, "Quality metrics for optical signals: Eye diagram, Q-factor, OSNR, EVM and BER," in *Proc. Int. Conf. Transparent Opt. Netw. (ICTON)*, 2012, p. Mo.B1.5.