# End-to-End Learning for 100G-PON Based on Noise Adaptation Network

Yongxin Xu, Luyao Huang, Wenqing Jiang, Xiaokai Guan, Weisheng Hu, and Lilin Yi

***Abstract*—End-to-end learning is a promising solution to realize the optimal performance of optical communication systems. By replacing the independent signal processing modules in the transmitter and receiver with deep neural networks, end-to-end system optimization can be achieved via training the neural networks together on a differentiable channel. In this paper, a noise adaptation network for channel modeling is proposed to simulate channel response and the impact of channel noise on transmitted signals. The structure of the noise adaptation network is multi-scale deep neural network (MscaleDNN), which can better characterize the channel at different frequencies. Based on the noise adaptation network, a novel end-to-end learning framework is further designed. Within the framework, memory buffer technology and constraint loss are introduced to significantly enhance the efficiency and performance of end-to-end learning. Experimental demonstration of the proposed end-to-end learning scheme is performed on a 100G passive optical network (PON) system based on intensity modulation and direct detection. The results indicate that, compared to the optimized Volterra non-linear equalization at the receiver and the joint equalization achieved by indirect approach, end-to-end optimization improves receiver sensitivity by 0.8 dB and 1.2 dB, respectively, and achieves a power budget of 31.4 dB. In particular, the advantages of the end-to-end learning are even more pronounced in the case of higher received optical power.**

***Index Terms*—Constraint loss, end-to-end learning, memory buffer, MscaleDNN, noise adaptation network, PON.**

## I. Introduction

STIMULATED by rapid traffic growth in the access and short-reach networks, single-wavelength data rate of passive optical network (PON) is moving from 25 Gbps to 50 Gbps and 100 Gbps. Nowadays, 50G-PON has been standardized by ITU-T and IEEE [1], [2], [3]. The 50G-PON physical layer keeps the intensity modulation and direct detection (IM/DD) scheme and employs digital signal processing (DSP) for the first time. The standardization of 100G-PON is still on the way, and research on the key technologies of 100G-PON has been widely investigated [4], [5], [6], [7], [8], [9], [10]. Compared to complex and expensive coherent architecture, the implementation of 100G-PON prefers low-cost IM/DD, especially in the O-band. The main challenges for high-speed IM/DD PON system are bandwidth limitation, channel impairments and low receiver sensitivity, which means that it is hard to meet the performance requirement such as an optical power budget of at least 29 dB [11], [12].

In recent years, as the in-depth research of machine learning technique in optical communication systems, end-to-end learning has earned a lot of attention and is regarded as a promising way to improve the overall system performance [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. In a conventional communication system, signal processing modules at both transmitter and receiver are designed to undertake individual tasks, such as encoding/decoding, modulation/demodulation, pulse shaping, as well as equalization. This block-wise approach allows each module to be analyzed and optimized separately. However, it is difficult to ensure global optimization of system performance when the channel model becomes more complex.

End-to-end learning provides a feasible solution to this problem. Generally, it uses two neural networks as transmitter and receiver to replace the independent signal processing modules. The neural networks are trained on a specific-designed differentiable channel model, which is a simulation of the real physical channel. Through joint training of transmitter and receiver by gradient-based algorithms, the system can achieve the best end-to-end performance. The effect of end-to-end learning is highly dependent on the accuracy of the differentiable channel model. The closer the differentiable channel is to the real physical channel, the better the end-to-end learning performance will be. Some studies directly use theoretical formulas or known numerical models of real physical channel to construct differentiable channel [14], [15], [16], [17], [18]. However, in a real optical communication system, the transfer function between the input signal and the output signal is highly complex, making it challenging to be accurately modeled with mathematical equations. Thus, generative adversarial network (GAN) [19], [20], [21], long short-term memory network (LSTM) [22] and other deep learning methods are used to simulate channel characteristics through a data-driven approach. These schemes circumvent the restriction on the need for prior knowledge of practical channel model in end-to-end learning. The neural network responsible for modeling channel becomes part of the end-to-end learning. It can be alternately trained with the neural networks in the

Manuscript received 8 May 2023; revised 17 August 2023, 13 October 2023, and 20 November 2023; accepted 7 December 2023. Date of publication 12 December 2023; date of current version 2 April 2024. This work was supported in part by the National Key R&D Program of China under Grant 2019YFB1803803 and in part by the National Natural Science Foundation of China under Grant 62025503. *(Corresponding author: Lilin Yi.)*

The authors are with the State Key Laboratory of Advanced Optical Communication Systems and Networks, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xuyongxin@sjtu.edu.cn; luyaohuang@sjtu.edu.cn; sjtujwq@sjtu.edu.cn; guanxiaokai@sjtu.edu.cn; wshu@sjtu.edu.cn; lilinyi@sjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JLT.2023.3341495.

Digital Object Identifier 10.1109/JLT.2023.3341495

transmitter and receiver until the loss functions of all networks converge, or it can be pre-trained independently and used as a fixed channel. There are also some end-to-end learning schemes choosing to optimize the transmitter and receiver without modeling the channel, for example, by reinforcement learning [23], cubature Kalman filtering [24] or gradient estimation [25], but they are more difficult to train due to the inaccurate and unstable feedback gradient flow.

However, the majority of current verifications of end-to-end learning are based on simulated communication systems, where many techniques may not be helpful and some aspects are easy to be overlooked. When carrying out end-to-end learning for a practical optical communication system in an online fashion, training the channel model becomes a great challenge. The channel state may be affected by the environment and undergo some changes, causing the pre-trained channel network to lose its match. Also, frequent data acquisition from the practical system during online training is time-consuming, as the data has to be collected from the oscilloscope, and synchronization, resampling and other operations are required. Although the validity of GAN and LSTM in channel modeling has been verified [26], [27], their network structures hinder the training efficiency. Both GAN and LSTM consume plenty of time to train. Besides, the design of neural networks in transmitter and receiver is also critical for end-to-end performance. A neural network with weak fitting ability cannot thoroughly compensate for channel impairments. Therefore, careful design of neural networks used for channel modeling, transmitter, and receiver is essential to achieve higher efficiency and performance in end-to-end learning.

Considering that the channel output is a random variable, we propose a noise adaptation network to model channel response and the impact of channel noise on transmitted signals based on the principle of maximum likelihood estimation. The structure of noise adaptation network adopts the feedforward multi-scale deep neural network (MscaleDNN) [28]. It is designed based on *Frequency Principle* [29] (the principle reveals the training dynamics of neural networks) and possesses the capability to effectively learn the response at different frequencies. Based on the noise adaptation network, a novel end-to-end learning framework is built. Through alternately training the noise adaptation network with the neural networks in transmitter and receiver, noise adaptation network gradually achieves an accurate approximation of the real physical channel and the neural networks in transmitter and receiver guide system performance to the optimum. In addition, to further enhance the training efficiency and performance, memory buffer technology [30] and constraint loss are introduced into the framework. Memory buffer can reduce the number of channel data requests from the practical system and accelerate training process. Constraint loss forces the transmitter to take on more signal processing tasks, thus ensuring a simple neural network in the receiver. It is more suitable for the DSP complexity requirements of PON.

The proposed end-to-end learning framework is applied to a 100Gbps O-band IM/DD PON experimental system with a 10-GHz Mach-Zehnder modulator (MZM). MscaleDNN is used for the neural network in the transmitter and a single layer of fully connected neural network (i.e., feed-forward equalizer, FFE) for the receiver. After online training, experimental results demonstrate that the end-to-end learning framework has a fast convergence rate. The noise adaptation network accurately models the experimental channel, enabling the transmitter and receiver neural networks to provide clean compensation for both linear and nonlinear impairments in the channel. Compared to the optimized traditional receiver-only Volterra non-linear equalization (VNLE) scheme and the joint equalization scheme achieved by indirect method [33], our approach improves receiver sensitivity by 0.8 dB and 1.2 dB, respectively, while providing even more significant gains at higher received optical powers. To the best of our knowledge, this is the first experimental system of 100G-PON that employs end-to-end learning.

The rest of this paper is organized as follows. Firstly, the basic principles of noise adaptation network for channel modeling are introduced in Section II. Then, the principle and implementation process of proposed end-to-end learning framework are presented in Section III. Next, in Section IV, experimental studies and results are described. Finally, a conclusion is given in Section V.

## II. Noise Adaptation Network for Channel Modeling

In the end-to-end learning framework for optical communication systems, the construction of an accurate differentiable channel plays a key role in performing gradient–based optimization of transmitter and receiver. Considering that the real physical channel output signal is a random variable, a noise adaptation network is designed for modeling the channel from the perspective of parameter estimation through maximum likelihood estimation. Specifically, assuming that the channel input is $\boldsymbol{x} \in \mathbb{R}^{m \times sps}$, the output is $\boldsymbol{y} \in \mathbb{R}^{sps}$, and $\boldsymbol{y}$ follows a $sps$-dimensional Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^{sps}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{sps \times sps}$, the probability density function (PDF) of $\boldsymbol{y}$ is

$$f\ (\boldsymbol{y}) = \frac{1}{(2\pi)^{\frac{sps}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} \left[ (\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}) \right] \right\}, \tag{1}$$

where $m$ is the number of symbols contained in the input signal $\boldsymbol{x}$, $sps$ is the number of samples per symbol, and $\boldsymbol{y}$ represents the output signal corresponding to the center position of $\boldsymbol{x}$. Therefore, the noise adaptation network aims to estimate the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of the output signal $\boldsymbol{y}$. It characterizes the real physical channel, including imitating the channel frequency response and modeling the impact of channel noise on the transmitted signal.

According to the type of noise in the channel, two noise adaptation network architectures are designed. The first one is for the case where the noise is signal-dependent, which is often manifested as multiplicative noise. When such noise dominates, the network architecture is shown in Fig. 1. It consists of a mean network, denoted as $h$, and a variance network, denoted as $s$. They take the signal $\boldsymbol{x} \in \mathbb{R}^{m \times sps}$ as input and produce corresponding outputs, $h(\boldsymbol{x}) \in \mathbb{R}^{sps}$and $s(\boldsymbol{x}) \in \mathbb{R}^{sps}$, respectively. In the training mode, the noise adaptation network is trained based on the maximum log-likelihood function method. Let $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_B, \boldsymbol{y}_B)\}$ denote the $B$ groups

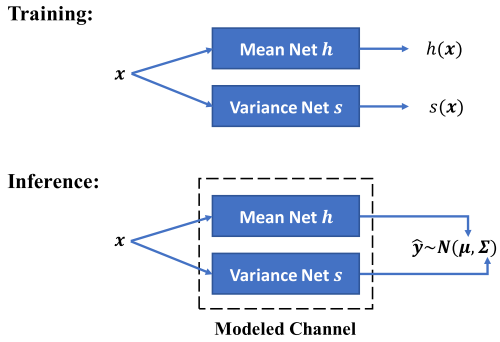


Fig. 1. Architecture of noise adaptation network when the noise is signal-dependent.

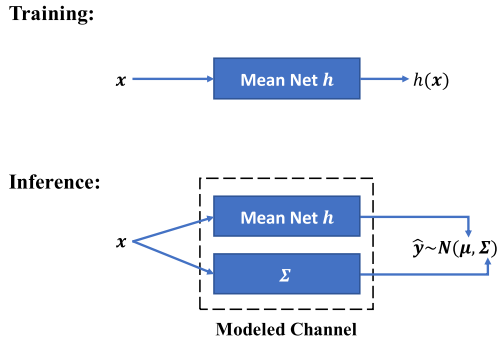


Fig. 2. Architecture of noise adaptation network when the noise is signal-independent.

of data and labels sampled from the training dataset. Then the maximized log-likelihood function can be expressed as follows:

$$\begin{aligned}&\arg\max_{h,s}\sum_{i=1}^{B}\ln\\&\left[\frac{1}{(2\pi)^{\frac{sps}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}_i-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_i)\right]\right\}\right]\\=&\arg\max_{h,s}\sum_{i=1}^{B}\left[-\frac{1}{2}\left[(\boldsymbol{y}_i-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_i)\right]\right.\\&\left.-\frac{sps}{2}\ln|\boldsymbol{\Sigma}_i|-\frac{sps}{2}\ln 2\pi\right],\end{aligned}\tag{2}$$

where $\boldsymbol{\mu}_i = h(\boldsymbol{x}_i)$, $\boldsymbol{\Sigma}_i = diag(s(\boldsymbol{x}_i))$, and $diag(\cdot)$ denotes the diagonalization of a vector. In the inference mode, mean network $h$ and variance network $s$ together constitute the modeled channel, simulating the channel response to the input signal $\boldsymbol{x}$ and the effect of channel noise on $\boldsymbol{x}$, respectively. Therefore, The output of the noise adaptation network is $\hat{\boldsymbol{y}} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, i.e., sampled from the Gaussian distribution with mean $\boldsymbol{\mu} = h(\boldsymbol{x})$ and covariance matrix $\boldsymbol{\Sigma} = diag(s(\boldsymbol{x}))$.

Another type of noise is signal-independent, such as white Gaussian noise, which is the most common situation in communication systems. When it dominates, the corresponding noise adaptation network architecture can be simplified as Fig. 2, it has only one mean network. In training mode, the maximized log-likelihood function can be expressed as follows:

$$\begin{aligned}&\arg\max_{h}\sum_{i=1}^{B}\ln\\&\left[\frac{1}{(2\pi)^{\frac{sps}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}\left[(\boldsymbol{y}_i-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}_i-\boldsymbol{\mu}_i)\right]\right\}\right]\\=&\arg\max_{h}\sum_{i=1}^{B}\left[-\frac{1}{2}\left[(\boldsymbol{y}_i-\boldsymbol{\mu}_i)^T\sigma^{-2}(\boldsymbol{y}_i-\boldsymbol{\mu}_i)\right]\right.\\&\left.-sps\ln\sigma-\frac{sps}{2}\ln 2\pi\right],\end{aligned}\tag{3}$$

where $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$, $\boldsymbol{I}$ denotes the unit matrix, $\sigma^2$ is a parameter to be estimated. When the training is completed, $\sigma^2$ is calculated on the full training dataset as follows:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}||h(\boldsymbol{x}_i)-\boldsymbol{y}_i||^2,\tag{4}$$

where $n$ is the size of the entire training dataset. Similarly, in the inference mode, the output $\hat{\boldsymbol{y}}$ of the noise adaptation network is also sampled from the Gaussian distribution with mean $\boldsymbol{\mu} = h(\boldsymbol{x})$ and covariance matrix $\boldsymbol{\Sigma}$.

Unlike previous schemes that fit channel with normal fully connected neural network, convolutional neural network or LSTM, the noise adaptation network uses MscaleDNN as its structure in order to balance the fitting ability and training cost. MscaleDNN is a kind of network architecture designed to solve high-frequency component learning problems according to the *Frequency Principle* [29]. The principle reveals the dynamics of the neural network training process and points out that deep neural networks often fit target functions from low to high frequencies, and they learn the low frequency content quickly with good generalization error, but they will be inadequate when learning high frequency content. Here the concept of frequency is introduced by the Fourier transform of the input-output mapping of the neural network. For communication channel and equalizer, they are also characterized by the response at different frequencies in frequency domain. Therefore, when the mean network of the noise adaptation network uses MscaleDNN to fit the channel response, the channel features from low to high frequencies can be well simulated. Similarly, neural network equalizers using MscaleDNN can also compensate for responses at different frequencies well.

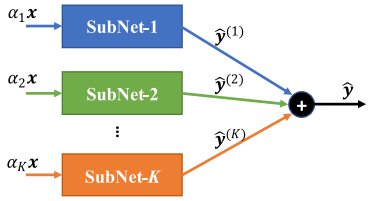


Fig. 3. Architecture of MscaleDNN.

The architecture of MscaleDNN is shown in Fig. 3, as a sum of $K$ subnetworks, in which each scale input goes through a

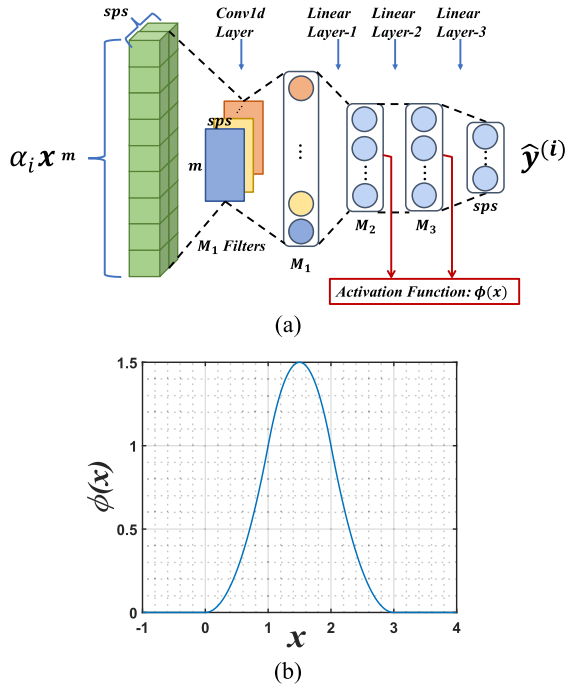


Fig. 4. (a) The concrete structure of the subnetwork in MscaleDNN. (b) The activation function $\phi(x)$.

subnetwork, $\alpha_1, \alpha_2, \ldots, \alpha_K$ are scale coefficients, usually set $\alpha_i = i$ or $\alpha_i = 2^{i-1}$ . The concrete subnetwork structure is a feedforward structure, containing a 1-D convolutional layer and three linear layers as shown in Fig. 4. Conv1d Layer has $M_1$ filters with the size of $m \times sps$, they convolve the input $\alpha_i \boldsymbol{x} \in \mathbb{R}^{m \times sps}$ and pass the result to the subsequent linear layers. The outputs of Linear Layer-1 and Linear Layer-2 need to be activated with the activation function $\phi\ (x) = (x-0)_+^2 - 3(x-1)_+^2 + 3(x-2)_+^2 - (x-3)_+^2$, where $x_+ = \max\{x, 0\}$ [28]. Linear Layer-1 and Linear Layer-2 have $M_2$, $M_3$ neurons respectively. Finally, the output $\hat{\boldsymbol{y}}^{(i)}$ is obtained.

It is worth emphasizing that in the experiments, the approximation error, generalization error and optimization error [31], [32] are always present when training noise adaptation network. As a result, the estimation of noise variance tends to be large and the features of channel may be masked. To overcome these errors, we truncate the output result. When the noise adaptation network is in the inference mode, the network output is sampled from the truncated Gaussian distribution (in the training mode, the Gaussian distribution is still used), i.e., the network output $\hat{\boldsymbol{y}} \sim \boldsymbol{TN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{a}, \boldsymbol{b})$ and the PDF of $\hat{\boldsymbol{y}}$ is as follows:

$$f\ (\hat{\boldsymbol{y}}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{a}, \boldsymbol{b}) = \frac{\exp\left\{-\frac{1}{2}\left[(\hat{\boldsymbol{y}}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{y}}-\boldsymbol{\mu})\right]\right\}}{\int_{\boldsymbol{a}}^{\boldsymbol{b}} \exp\left\{-\frac{1}{2}\left[(\hat{\boldsymbol{y}}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{y}}-\boldsymbol{\mu})\right]\right\} d\boldsymbol{y}} I_{[\boldsymbol{a},\boldsymbol{b}]}(\hat{\boldsymbol{y}}). \quad (5)$$

where $[\boldsymbol{a}, \boldsymbol{b}]$ is truncation interval, $I_{[\boldsymbol{a},\boldsymbol{b}]}(\hat{\boldsymbol{y}})$ is indicator function, $I_{[\boldsymbol{a},\boldsymbol{b}]}\ (\hat{\boldsymbol{y}}) = 1$ if $\boldsymbol{a} \leq \hat{\boldsymbol{y}} \leq \boldsymbol{b}$. In this paper, we set $\boldsymbol{a} = \boldsymbol{\mu} - 3\boldsymbol{\Sigma}_{dd}$ and $\boldsymbol{b} = \boldsymbol{\mu} + 3\boldsymbol{\Sigma}_{dd}$. The $\boldsymbol{\Sigma}_{dd}$ is a column vector composed of the main diagonal elements of $\boldsymbol{\Sigma}$.

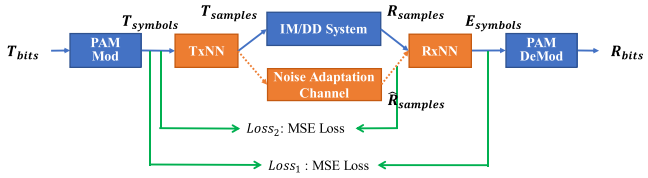


Fig. 5. End-to-end learning framework for IM/DD PON.

## III. Principles of End-to-End Learning

Generally, in an end-to-end communication system, two symmetric neural networks are used as transmitter and receiver. which are regarded as an autoencoder (AE) structure. And then they are trained on a differentiable channel, usually modeled with a neural network, by gradient-based algorithms. During the training process, some schemes alternately train the channel network with the networks in transmitter and receiver. Others pre-train the channel network and keep it fixed, then go on to train the networks in transmitter and receiver. The former enables better learning of channel characteristics through continuous dynamic learning.

In this work, for an IM/DD PON system, we employ the noise adaptation network to model the IM/DD system and provide a differentiable channel, i.e., noise adaptation channel, for end-to-end training. MscaleDNN is used at the transmitter and a single-layer fully connected neural network (i.e., FFE) is used at the receiver. Noise adaptation channel is trained alternately with the neural networks in transmitter and receiver. Moreover, constraint loss and memory buffer are designed to further improve training performance and efficiency.

### A. End-to-End Learning Framework

The end-to-end learning framework is shown as Fig. 5. Starting from the transmitter, the transmitted bit sequence $T_{bits}$ is first modulated into PAM symbols $T_{symbols}$ with amplitude [-3,-11,3]. After power normalization, a sliding window is applied to $T_{symbols}$ to construct the t-th input $\boldsymbol{s}_t = [T_{symbols}(t - \frac{m-1}{2}), \ldots, T_{symbols}(t + \frac{m-1}{2})]^T \in \mathbb{R}^m$ for TxNN with a window size of $m$. TxNN is responsible for pulse shaping and pre-equalization. Its structure is the same as MscaleDNN presented in Figs. 3 and 4, except that TxNN's input data has a dimension of $m \times 1$ rather than $m \times sps$. The t-th output of TxNN is denoted by $\boldsymbol{e}_t \in \mathbb{R}^{sps}$, and all outputs constitute the transmitter signal $T_{samples}$ and the power of $T_{samples}$ is normalized. Noise adaptation channel simulates the transmission process of $T_{samples}$ in the IMDD system. The t-th input of Noise adaptation channel is $\boldsymbol{x}_t = [T_{samples}(t - \frac{m-1}{2}), \ldots, T_{samples}(t + \frac{m-1}{2})]^T \in \mathbb{R}^{m \times sps}$. The corresponding output $\hat{\boldsymbol{y}}_t \in \mathbb{R}^{sps}$ is sampled from the truncated normal distribution, as shown in (5) with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ estimated according to (2) or (3) and (4). All outputs constitute the receiver signal $\hat{R}_{samples}$. At the receiver, $\hat{R}_{samples}$ is recovered to modulated symbols by RxNN, noted as $E_{symbols}$. RxNN is a single-layer fully connected neural network. The $t$-th input of RxNN is $\boldsymbol{y'}_t =$

$[\hat{R}_{samples}(t-\frac{m'-1}{2}),\ldots,\hat{R}_{samples}(t+\frac{m'-1}{2})]^{\mathrm{T}} \in \mathbb{R}^{m'\times sps}$. The output is $s'_t \in \mathbb{R}$. Through PAM demodulation, $E_{symbols}$ is finally mapped to bit sequence $R_{bits}$.

In this framework, noise adaptation channel is trained alternately with TxNN and RxNN through stochastic gradient descent. The task of noise adaptation channel is to learn the mean value $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ of the output signal in the IM/DD system according to (2) or (3) and (4) with the data $T_{samples}$ and $R_{samples}$. $R_{samples}$ is obtained by performing synchronization and resampling on the signal acquired from the oscilloscope. For TxNN and RxNN, the task is to minimize the joint mean square error (MSE) loss $Loss_{TR}$, which is defined as follows:

$$Loss_{TR} = Loss_1 + Loss_2 \tag{6}$$

where $Loss_1$ is the MSE loss between $T_{symbols}$ and $E_{symbols}$.It can be expressed as:

$$Loss_1 = \frac{1}{n}\sum_{i=1}^{n}||T_{symbols}(i) - E_{symbols}(i)||^2 \tag{7}$$

The $Loss_2$ is the MSE loss between $T_{symbols}$ and $\hat{R}_{samples}$. It can be expressed as:

$$Loss_2 = \frac{1}{n}\sum_{i=1}^{n}||T_{symbols}(i) - \hat{R}_{symbols}(i)||^2 \tag{8}$$

$$\hat{R}_{symbols}(i) = \frac{1}{sps}\sum_{k=1}^{sps}\hat{R}_{samples}(k,i) \tag{9}$$

where $n$ is the size of $T_{symbols}$.

The $Loss_2$ is a constraint loss, similar to the constraints imposed on the target solution in the optimization area. The $Loss_2$ forces the TxNN to take on more signal processing tasks in end-to-end learning, thus ensuring that the RxNN can use a simple FFE structure, which is more in line with the DSP complexity requirements of PON. $Loss_1$ and $Loss_2$ have the same physical meaning, so for simplicity, their weights in the overall loss $Loss_{TR}$ are considered to be equal.

### B. Memory Buffer-Assisted Noise Adaptation Channel Training

Modeling channel with neural network requires a large amount of input and output data from the real physical channel. In the previous alternate training approach, each time the training of the channel network is started, the neural network in the transmitter is fixed. Then the different sequences of random bits are continuously sent to generate a sufficient amount of channel training data. Since the neural network in the transmitter is fixed, this approach actually leads to a single distribution of channel input data. It does not allow the channel network to be more generalized. What's more, acquiring channel output data from the experimental system frequently is time-consuming, because the data collected from the oscilloscope still needs to be synchronized and resampled.

Therefore, memory buffer-assisted training mechanism is introduced to improve the training performance and accelerate the training process of the channel network. As shown in Fig. 6, we first save the input signal $T_{samples}$ and the output signal $R_{samples}$ generated by the IM/DD system into memory buffer. When starting to train the channel network, $N$ sets of signals are randomly sampled from memory buffer to construct the training dataset. With a sliding window of size $m$, we can obtain $n$ pairs of training data $\boldsymbol{x} \in \mathbb{R}^{m\times sps}$ and label $\boldsymbol{y} \in \mathbb{R}^{sps}$ from each set of signals. Then $N*n$ pairs of data and labels are used to train the noise adaptation channel through supervised learning. The simple mechanism not only enables a richer distribution of training data for the channel network, allowing better generalization of the noise adaptation channel, but also saves end-to-end training time by avoiding frequent data acquisition from real physical channel.

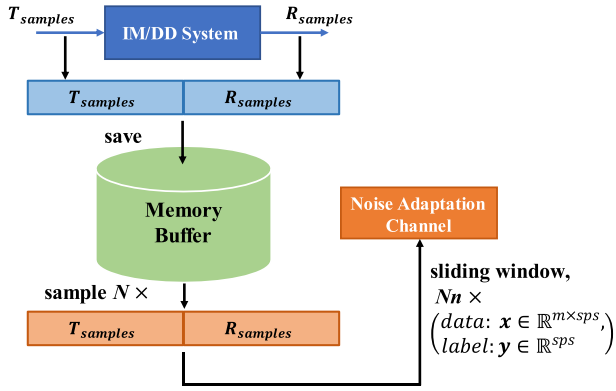


Fig. 6. Memory buffer-assisted noise adaptation channel training end-to-end learning.

### C. End-to-End Training Process

The detailed training process of end-to-end learning based on the noise adaptation network is summarized in Algorithm 1, including initialization, main loop and finetune RxNN in the IM/DD system. In the main loop, noise adaptation channel is first trained in inner loop-1, then TxNN and RxNN are trained in inner loop-2. The two inner loops are repeated until the loss functions converge. Finally, to compensate for the difference between the modeled channel and the real physical channel for better end-to-end learning performance, RxNN is finetuned with the transmitted and received signals from IM/DD system. Adam optimizer is used in the training process.

At the inner loop-1, we need acquire data from the experimental system to train the noise adaptation channel. Under each main loop, inner loop-1 is executed ChEpoch times. Each time, random bits are generated and mapped to PAM-4 symbols, and after processing by TxNN and power normalization, we obtain $T_{samples}$. Then we load $T_{samples}$ into AWG. After 20-km transmission of IM/DD system, we collect signals from the oscilloscope, and perform synchronization and resampling to obtain $R_{samples}$. Finally, we save a set of training samples $[T_{samples}, R_{samples}]$ into the memory buffer.

In practical implementation, we can pre-train the TxNN, noise adaptation channel and RxNN according to the specified channel conditions (including signal rate, fiber distance, transmission

**Algorithm 1:** End-to-End Learning Based on Noise Adaptation Network.

1: Select the type of noise adaptation network.
2: Randomly initialize noise adaptation channel $\boldsymbol{C_{h,s}}$, TxNN $\boldsymbol{T_\theta}$, RxNN $\boldsymbol{R_w}$ and memory buffer.
3: **Main loop**:
4: **for** $i = 1$ : MainEpoch
5: **Inner loop-1**:
6: **for** $j = 1$ : ChEpoch
7: Keep $\boldsymbol{T_\theta}$ and $\boldsymbol{R_w}$ fixed, send random bit sequence.
8: Save the transmitted signal $\boldsymbol{T_{samples}}$ and received signal $\boldsymbol{R_{samples}}$ to memory buffer.
9: Sample $\boldsymbol{N}$ sets of $\boldsymbol{T_{samples}}$ and $\boldsymbol{R_{samples}}$, construct training dataset. Update $\boldsymbol{C_{h,s}}$ according to (2) or (3) and (4).
10: **endfor**
11: **Inner loop-2**:
12: **for** $k = 1$ : TREpoch
13: Keep $\boldsymbol{C_{h,s}}$ fixed, set to inference mode and its output $\hat{\boldsymbol{y}} \sim \boldsymbol{TN}(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{a}, \boldsymbol{b})$.
14: Send random bit sequence, construct training dataset with $\boldsymbol{T_{symbols}}$, $\hat{\boldsymbol{R}}_{\boldsymbol{samples}}$ and $\boldsymbol{E_{symbols}}$.
15: Update $\boldsymbol{T_\theta}$ and $\boldsymbol{R_w}$ by minimizing the loss function defined as (6).
16: **endfor**
17: **endfor**
18: **Finetune RxNN in IM/DD system**:
19: **for** $l = 1$ : FEpoch
20: Keep $\boldsymbol{T_\theta}$ fixed, send random bit sequences.
21: Construct training dataset with IM/DD system output signal $\boldsymbol{R_{samples}}$ and $\boldsymbol{T_{symbols}}$.
22: Update $\boldsymbol{R_w}$ by minimizing the loss function defined as (7).
23: **endfor**

power, etc.). Then update the TxNN and RxNN by periodically tracking the channel state with noise adaptation channel. Due to the memory buffer mechanism, only a small number of training samples are needed. Another application is to perform end-to-end learning at the optical back-to-back scenario so that the TxNN can compensate impairments of OLT transmitter and ONU receiver through constraint loss. Then each ONU trains a low-complexity RxNN (FFE) independently to compensate for residual impairments of devices and the impairments introduced by O-band channel transmission.

## IV. Experimental Verification and Results

### A. Experimental Setup

The end-to-end learning algorithm is evaluated in the 100 Gb/s PAM-4 IM/DD transmission system operating at the O-band. Two typical experimental system architectures with equalizers are shown in Fig. 7(a) and (b). The former is a receiver-only equalization scheme. The latter is a joint equalization scheme

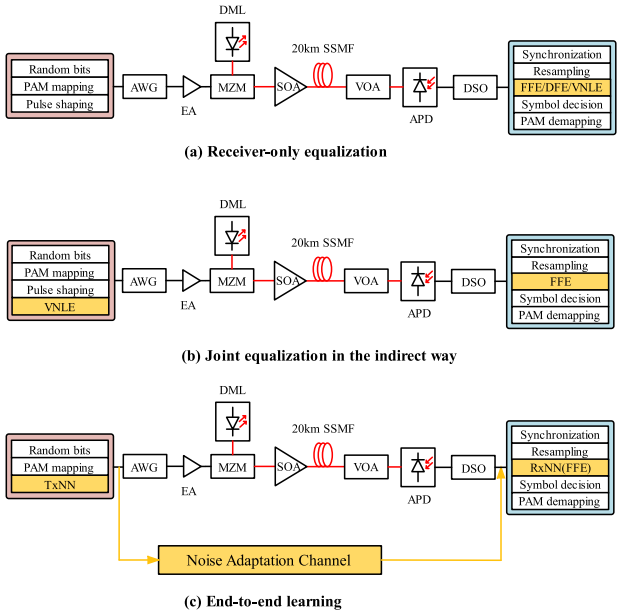


Fig. 7. 100 Gb/s PAM-4 IM/DD transmission system setup of (a) receiver-only equalization, (b) joint equalization in the indirect way and (c) end-to-end learning.

implemented in an indirect way [31], which means that the equalizer trained at the receiver is first put on the transmitter and then a new equalizer is trained at the receiver. In Fig. 7(a), the random bits are generated and mapped to 50 GBaud PAM-4 symbols firstly. After pulse shaping is performed by a root raised cosine (RRC) filter with roll-off factor of 0.4 (traversal optimization from 0.1 to 1 with interval of 0.1), the digital signal is converted to the analog signal by a Keysight M8194A arbitrary waveform generator (AWG) with the sampling rate of 120 GSa/s and output voltage amplitude of 200 mV. The signal from AWG is amplified by a 23-dB electrical amplifier (EA), then modulated by a 10-GHz Mach–Zehnder modulator (MZM). Meanwhile, the laser from a 1310 nm directly modulated laser (DML) is injected to the MZM biased at its quadrature point. A semiconductor optical amplifier (SOA) with a noise figure of 7.5 dB is followed to control the optical launch power. The launch power is set to 12dBm for better power budget. After 20-km standard single mode fiber (SSMF) transmission, the power of the received signal is adjusted by a variable optical attenuator (VOA). Then the signal is detected by a 30-GHz avalanche photodiode (APD). Finally, the signal is extracted by a Tektronix digital storage oscilloscope (DSO) with a 33 GHz bandwidth and 100 GSa/s sampling rate for subsequent offline DSP, including synchronization, resampling to 2 sps, equalization, symbol decision and PAM demapping. Half-symbol-spaced FFE, decision feedback equalizer (DFE) and VNLE are used for equalization. Because half-symbol-spaced equalizers are no sensitivity to timing phase and have superior performance in most cases over their symbol-spaced counterparts. Fig. 8 shows the optical back-to-back frequency response of experimental system measured by a Vector Network Analyzer (VNA). The end-to-end 3-dB bandwidth is about 9.578 GHz and the 10-dB bandwidth is about 22.405 GHz.

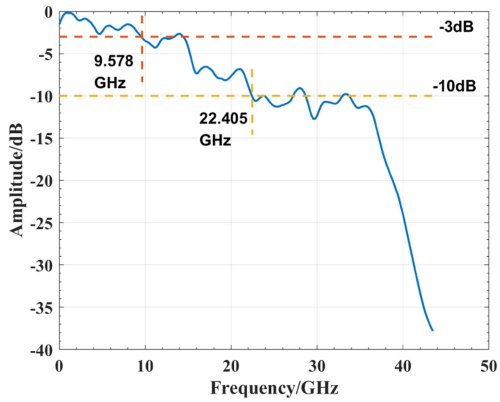


Fig. 8. Optical back-to-back frequency response of the 100 Gb/s PAM-4 IM/DD transmission system.

TABLE I
NETWORK STRUCTURE PARAMETERS OF MSCALEDNN

| Parameter | Value |
|---|---|
| Number of subnetworks $K$ | 5 |
| Scale coefficients | $\alpha_i = 2^{i-1}, i = 1,2,\ldots,5$ |
| $m$ | 129 |
| $sps$ | 2 |
| $M_1$ | 32 |
| $M_2$ | 16 |
| $M_3$ | 16 |

In Fig. 7(b), we first train a half-symbol-spaced VNLE at the receiver with the same structure as shown in Fig. 7(a). But its output is 2 sps and it uses transmitted signal after pulse shaping as the training label. Then VNLE is moved to the transmitter and acts as a pre-equalizer. Finally, to compensate for residual impairments and obtain better performance, a half-symbol-spaced FFE is trained as the post-equalizer.

Fig. 7(c) is the experimental system architecture of end-to-end learning based on noise adaptation network. It complements the detailed experimental setup for the framework shown in Fig. 5. In the experiment, the transmitted signal is affected by amplifier spontaneous emission (ASE) noise in SOA. After being detected by APD, the shot noise, dark current noise and thermal noise are involved. These noises are independent of each other and have an additive effect on the output signal, which can be simply modeled as signal-independent Gaussian noise. Therefore, the noise adaptation network presented in Fig. 2 is adopted to model experimental channel. The network structure parameters of MscaleDNN used by TxNN and noise adaptation channel are shown in Table I, and RxNN has 130 (65 symbols $\times$ 2 sps) neurons.

The training process of Fig. 7(c) is described in Section III-C, and parameters corresponding to each procedure are set in Table II. At initialization, memory buffer is set as a queue and can save up to 500 sets of $\boldsymbol{T}_{samples}$ and $\boldsymbol{R}_{samples}$. In the early stage of end-to-end training, the channel model is not accurate and is not advisable to train TxNN and RxNN too many times. So we make TREpoch grow linearly from 5 to 50 before the 6-th main loop. For each completed main loop, 32768 random bits are sent to calculate the BER and evaluate the current performance of TxNN and RxNN. When the entire end-to-end training process is finished, 3 sets of 32768 random bits are used for test and the average BER is calculated to compare with other schemes.

TABLE II
TRAINING PARAMETERS OF END-TO-END LEARNING

| | Parameter | Value |
|---|---|---|
| Initialization | Memory buffer size | 500 sets |
| Main loop | MainEpoch | 150 |
| Inner loop-1 | ChEpoch | 4 |
| | Number of bits sent | 8192 |
| | Learning rate | 1e-4 |
| | Batchsize | 512 |
| | Number of samples from memory buffer per ChEpoch | 25 sets |
| Inner loop-2 | TREpoch | 50 |
| | Number of bits sent | 2048 |
| | Learning rate | 5e-4 |
| | Batchsize | 1024 |
| Finetune RxNN | FEpoch | 50 |
| | Number of bits sent | 32768 |
| | Learning rate | 5e-4 |
| | Batchsize | 32 |

TABLE III
SETTINGS FOR DIFFERENT CASES

| | TxNN | Channel Network structure | Training Strategies |
|---|---|---|---|
| Case-1 | MscaleDNN | MscaleDNN | Noise adaptation channel, Memory buffer, Constraint loss |
| Case-2 | ResNet | ResNet | / |
| Case-3 | MscaleDNN | ResNet | / |
| Case-4 | MscaleDNN | ResNet | Noise adaptation channel, Memory buffer, Constraint loss |

### B. Comparison of Different End-to-End Learning Cases

Firstly, we compare four different end-to-end learning cases to demonstrate the effect of noise adaptation channel, constraint loss and memory buffer.

The settings of four cases are shown in Table III. The first case (case-1) is the proposed end-to-end learning framework based on noise adaptation network. Compared with case-1, the second case (case-2) takes a residual network (ResNet) shown in Fig. 9 as TxNN and channel network. It uses exponential linear unit (ELU) [34] as the activation function for better performance. When the channel network models the channel, it just considers the MSE loss between its output and the real physical channel output, and does not model the impact of noise on the signal. In the case-2, constraint loss and memory buffer are not used, so the ChEpoch is set as 20 and the number of bits sent is set as 16384 to allow the channel network to be fully trained. However, ChEpoch cannot be set too large in these cases. Because it will cause the channel network to overfit on a single distribution of training dataset, leading to poor end-to-end learning performance. In the third case (case-3), TxNN uses

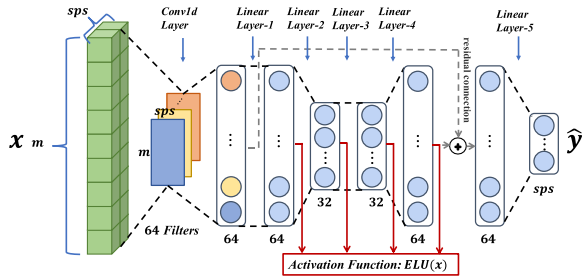


Fig. 9. Structure of ResNet used in case-2,3,4.

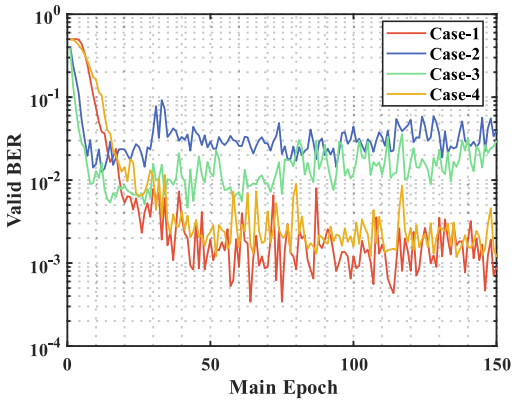


Fig. 10. Valid BER versus training epoch of four end-to-end learning cases at ROP = −15 dBm.

MscaleDNN, and other settings are the same to case-2. In the fourth case (case-4), ResNet is used as the network structure for noise adaptation channel, and other settings are same to case-1.

At the received optical power (ROP) of −15 dBm, the variation of valid BER with training epoch (i.e., main loop) for the four cases is shown in Fig. 10. From the comparison between case-2 and case-3, as well as case-1 and case-4, it can be observed that case-3 and case-1 can achieve lower BER levels than case-2 and case-4, respectively. It indicates that MscaleDNN has stronger fitting ability and can better compensate for high-frequency components in channel response when it acts as the structure of channel network and TxNN.

Furthermore, the valid BER of case-1 and case-4 drops to a lower level than that of case-2 and case-3. Case-1 and case-4 take noise adaptation network to simulate the mean and variance of the output signal, which is closer to actual system conditions. Meanwhile, the memory buffer provides random training data with different distributions for channel training. It can reduce the fitting bias and improve the generalization of the channel network. In the early epochs of end-to-end training, the reduction speed of valid BER in case-1 and case-4 is not as fast as that in case-2 and case-3. This is because in case-1 and case-4 ChEpoch is set to 4 while in case-2 and case-3 it is set to 20. So in the early epochs, there is not enough data in the memory buffer for training, resulting that the channel networks in case-1 and case-4 are not trained too many times. The most time-consuming part of the end-to-end learning, besides network training, is data acquisition from experimental system. The larger the ChEpoch,

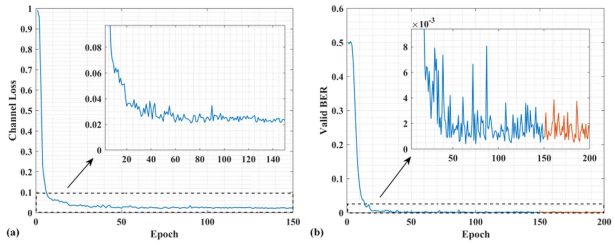


Fig. 11. (a) Training loss of noise adaptation channel versus training epoch and (b) valid BER of end-to-end experimental system versus training epoch and finetuning epoch at ROP = −15 dBm.

the longer the required training time. For case-2 and case-3, the total size of samples acquired from the communication system is MainEpoch × 20 × 16384 (8192 symbols × 2sps), while for case-1 and case-4, it reduces to MainEpoch × 4 × 8192 (4096 symbols × 2sps). By introducing memory buffer, we can reuse the channel data and reduce the training time.

Therefore, by comparing different cases, we adopt MscaleDNN as the TxNN and channel network structure with training strategies such as noise adaptation channel, memory buffer and constraint loss.

### C. Comparison Between End-to-End Learning and Other Equalization Schemes

Next, the end-to-end learning based on noise adaptation network is compared with two equalization schemes shown in Fig. 7(a) and (b).

The noise adaptation channel, TxNN and RxNN are trained according to the procedure in Section III-C. At ROP = −15 dBm, the training loss of noise adaptation channel is shown in Fig. 11(a). The valid BER of experimental system with TxNN and RxNN is shown in Fig. 11(b). In Fig. 11(a), it can be seen that the channel loss decreases rapidly and the convergence process is smooth with no great fluctuations in the curve, which is the benefit of the memory buffer-assisted noise adaptation channel training. In Fig. 11(b), the first 150 epochs are used to execute the main loop, and the last 50 epochs are for finetuning RxNN. According to the results, the valid BER in the latter stage is not much lower than the previous stage. It implies that the noise adaptation channel is an accurate simulation of the real physical signal, so there are not many differences to be compensated by finetuning RxNN.

Fig. 12 shows the power spectrum of the output signal on the noise adaptation channel and the real physical channel after end-to-end training is completed at ROP = -15 dBm. The power spectrum is flat because the signal is first preprocessed by TxNN. It compensates for the high frequency fading in the channel spectrum. The noise adaptation channel fits the real physical channel very well when the frequency range is 0∼30 GHz, while the frequencies above 30 GHz lead to large differences in the power spectrum. This is because the corresponding amplitudes are extremely small at high frequencies. It is more than 40 dB smaller than that at low frequencies, which means that there are few useful signal components and mostly noise components.

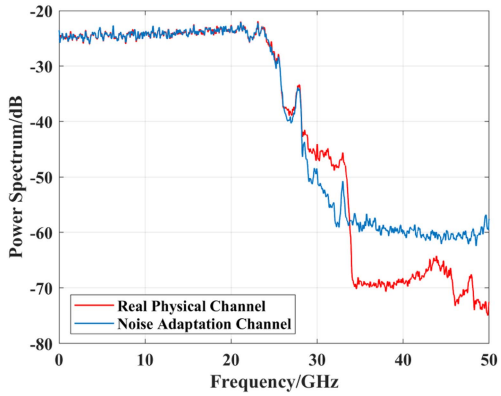


Fig. 12. Power spectrum of the output signal on the noise adaptation channel and the real physical channel at ROP = −15 dBm.

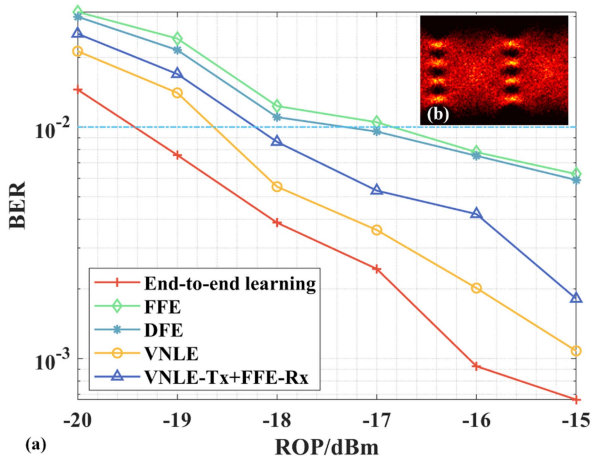


Fig. 13. (a) BER comparison of the three schemes under different ROPs, (b) eye diagram (sps = 2) of the real physical channel output signal in the end-to-end learning scheme at ROP = −15 dBm.

Fig. 13(a) shows the BER performance comparison of the three schemes at different ROPs. In the receiver-only equalization scheme, the tap coefficients of FFE, DFE and VNLE are updated through least mean square (LMS) algorithm. The 16384 transmitted symbols are used for training. FFE with a memory length of 130 is trained for 25 epochs at the learning rate of 0.005. DFE with a feedforward memory length of 130 and a feedback memory length of 25 is trained for 50 epochs at the learning rate of 0.001. The 1st, 2nd and 3rd order memory lengths of VNLE are (258,26,10), which are obtained by greedy optimization algorithm from the (11,1) to (262,30,14). VNLE is trained for 50 epochs at the learning rate of 0.005. "VNLE-Tx + FFE-Rx" denotes the joint equalization scheme implemented in the indirect way. The BER of all schemes is averaged over 3 sets of transmitted symbols with a length of 16384 to avoid accidental biases.

From Fig. 13(a), it can be observed that the BER performance of FFE and DFE is similar. This is because DFE introduces feedback filter to compensate for the zeros of transfer function. While in O-band transmission, the signal is not affected by dispersion, the power spectrum does not have zeros after direct detection, so DFE is not better than FFE. Considering that there are not only linear impairments such as band-limit in the system, but also non-linear impairments such as MZM and SOA nonlinearities, nonlinear equalizers are needed to compensate both of them. Therefore, VNLE and VNLE-Tx + FFE-Rx cause the BER to be further reduced. For VNLE-Tx + FFE-Rx, since VNLE cannot compensate for system impairments clearly and channel conditions vary slightly over time, the equalization capability decreases when a pre-trained VNLE is placed at the transmitter. Although another FFE is trained at the receiver to compensate for residual impairments, it is not as good as using VNLE only at the receiver. Compared to VNLE and VNLE-Tx + FFE-Rx, the end-to-end learning improves the receiver sensitivity by 0.8 dB and 1.2 dB, respectively, at the threshold of BER = 1e-2. The advantage of end-to-end learning is more obvious in the case of higher ROP, where the factors limiting the system are linear impairments and non-linear impairments. A near-global optimal signal processing scheme can be found to restore the signal through training on an accurate channel network model. However, at low ROP, the limiting factor for the system is noise, which cannot be compensated by equalization, so all the schemes become less different at low ROP.

Fig. 13(b) shows the eye diagram of the real physical channel output signal in the end-to-end learning scheme at ROP = −15 dBm. The eye diagram is wide open and clear. This indicates that TxNN compensates well for the vast majority of signal impairments under the control of the constrained loss, so that the compensation of residual system impairments can be accomplished by using a simple FFE as the RxNN. It is more suitable for PON system's demand for low-complexity DSP in ONUs.

Generalization is necessary for end-to-end learning. Since neural network-based modeling is data-driven, generalization can be achieved by modifying the training method and adding training data from different scenarios. In the proposed noise adaptation network architecture, we can collect signal datasets of different configurations and add a vector to denote channel configuration. Then signal vector and corresponding configuration vector are embedded together into the network for training. Finally, the output signal under expected configuration will be obtained. It will avoid repeatedly training different channel networks in the PON system.

## V. Conclusion

In this paper, a noise adaptation network to model channel response and the impact of channel noise on transmitted signals is proposed according to the principle of maximum likelihood estimation. It takes MscaleDNN as the network structure, which can better model the channel characteristics at different frequencies. Based on the noise adaptation network, we design a novel end-to-end learning framework and introduce memory buffer technology and constraint loss to enhance training efficiency and performance. The end-to-end learning framework is validated by a 100Gbps O-band IM/DD PON experimental system. After online training, the experimental results show that, noise adaptive network can achieve accurate modeling of real

physical channel and the end-to-end learning framework has a fast convergence rate. Compared with the optimized traditional receiver-only VNLE scheme and the joint equalization scheme implemented by the indirect method, the receiver sensitivity is improved by 0.8 dB and 1.2 dB, respectively, corresponding to a 31.4-dB loss budget. For future work, the content of end-to-end optimization will be further extended, such as geometric and probabilistic shaping at higher order modulation. The channel configuration parameters are considered to be embedded into the input of the neural network to enhance generalization ability. Meanwhile, the complexity of neural network will be further optimized by pruning to meet practical applications. Except for PON, the end-to-end learning framework can also be applied to short-reach optical interconnects and metro network.

## ACKNOWLEDGMENT

The authors would like to thank Mr. Yanming Zhou of Keysight Technologies (China) Co., Ltd. for providing AWG M8194A and relevant support.

## REFERENCES

[1] R. Bonk et al., "50G-PON: The first ITU-T higher-speed PON system," *IEEE Commun. Mag.*, vol. 60, no. 3, pp. 48–54, Mar. 2022.
[2] D. Zhang, D. Liu, X. Wu, and D. Nesset, "Progress of ITU-T higher speed passive optical network (50G-PON) standardization," *J. Opt. Commun. Netw.*, vol. 12, no. 10, pp. D99–D108, Oct. 2020.
[3] C. Knittle, "IEEE 50 Gb/s EPON (50G-EPON)," in *Proc. Opt. Fiber Commun. Conf.*, 2020, pp. Th1B–Th12.
[4] L. Yi, T. Liao, L. Huang, L. Xue, P. Li, and W. Hu, "Machine learning for 100 Gb/s/λ passive optical network," *J. Lightw. Technol.*, vol. 37, no. 6, pp. 1621–1630, Mar. 2019.
[5] M. G. Saber et al., "100 Gb/s/λ duo-binary PAM-4 transmission using 25G components achieving 50 km reach," *IEEE Photon. Technol. Lett.*, vol. 32, no. 3, pp. 138–141, Feb. 2020.
[6] J. Zhang et al., "SOA pre-amplified 100 Gb/s/λ PAM-4 TDM-PON downstream transmission using 10 Gbps O-band transmitters," *J. Lightw. Technol.*, vol. 38, no. 2, pp. 185–193, Jan. 2020.
[7] L. Huang et al., "Low-complexity Volterra-inspired neural network equalizer in 100-G band-limited IMDD PON system," *Opt. Lett.*, vol. 47, no. 21, pp. 5692–5695, Oct. 2022.
[8] P. Torres-Ferrera, G. Rizzelli, H. Wang, V. Ferrero, and R. Gaudino, "Experimental demonstration of 100 Gbps/λ C-band direct-detection downstream PON using non-linear and CD compensation with 29 dB+ OPL Over 0 Km–100 Km," *J. Lightw. Technol.*, vol. 40, no. 2, pp. 547–556, Jan. 2022.
[9] P. Torres-Ferrera, H. Wang, V. Ferrero, and R. Gaudino, "100 Gbps/λ PON downstream O- and C-band alternatives using direct-detection and linear-impairment equalization," *J. Opt. Commun. Netw.*, vol. 13, no. 2, pp. A111–A123, Feb. 2021.
[10] J. Zhang and Z. Jia, "Coherent passive optical networks for 100G/λ-and-beyond fiber access: Recent progress and outlook," *IEEE Netw.*, vol. 36, no. 2, pp. 116–123, Mar./Apr. 2022.
[11] V. Houtsma and D. Van Veen, "Optical strategies for economical next generation 50 and 100G PON," in *Proc. Opt. Fiber Commun. Conf.*, 2019, Paper M2B.1.
[12] V. Houtsma, A. Mahadevan, N. Kaneda, and D. van Veen, "Transceiver technologies for passive optical networks: Past, present, and future," *J. Opt. Commun. Netw.*, vol. 13, no. 1, pp. A44–A55, Jan. 2021.
[13] V. Bajaj, F. Buchali, M. Chagnon, S. Wahls, and V. Aref, "Deep neural network-based digital pre-distortion for high baudrate optical coherent transmission," *J. Lightw. Technol.*, vol. 40, no. 3, pp. 597–606, Feb. 2022.
[14] B. Karanov et al., "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, Oct. 2018.
[15] V. Neskorniuk et al., "End-to-end deep learning of long-haul coherent optical fiber communications via regular perturbation model," in *Proc. Eur. Conf. Opt. Commun.*, 2021, pp. 1–4.
[16] V. Aref and M. Chagnon, "End-to-end learning of joint geometric and probabilistic constellation shaping," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2022, pp. 1–3.
[17] J. Song, C. Häger, J. Schröder, A. G. I. Amat, and H. Wymeersch, "Model-based end-to-end learning for WDM systems with transceiver hardware impairments," *IEEE J. Sel. Topics Quantum Electron.*, vol. 28, no. 4, Jul./Aug. 2022, Art. no. 7700114.
[18] V. Neskorniuk et al., "Memory-aware end-to-end learning of channel distortions in optical coherent communications," *Opt. Exp.*, vol. 31, no. 1 pp. 1–20, Dec. 2022.
[19] B. Karanov et al., "End-to-end learning in optical fiber communications: Experimental demonstration and future trends," in *Proc. Eur. Conf. Opt. Commun.*, 2020, pp. 1–4.
[20] B. Karanov, M. Chagnon, V. Aref, D. Lavery, P. Bayvel, and L. Schmalen, "Concept and experimental demonstration of optical IM/DD end-to-end system optimization using a generative model," in *Proc. Opt. Fiber Commun. Conf.*, 2020, Paper Th2A.48.
[21] M. Li and S. Wang, "End-to-end learning for chromatic dispersion compensation in optical fiber communication," *IEEE Commun. Lett.*, vol. 26, no. 8, pp. 1829–1832, Aug. 2022.
[22] M. Li, D. Wang, Q. Cui, Z. Zhang, L. Deng, and M. Zhang, "End-to-end learning for optical fiber communication with data-driven channel model," in *Proc. IEEE Opto-Electron. Commun. Conf.*, 2020, pp. 1–3.
[23] Q. Zhang et al., "An improved end-to-end autoencoder based on reinforcement learning by using decision tree for optical transceivers," *Micromachines*, vol. 13, no. 1, 2022, Art. no. 31.
[24] O. Jovanovic, M. P. Yankov, F. Da Ros, and D. Zibar, "Gradient-free training of autoencoders for non-differentiable communication channels," *J. Lightw. Technol.*, vol. 39, no. 20, pp. 6381–6391, Oct. 2021.
[25] Z. Niu, H. Yang, H. Zhao, C. Dai, W. Hu, and L. Yi, "End-to-end deep learning for long-haul fiber transmission using differentiable surrogate channel," *J. Lightw. Technol.*, vol. 40, no. 9, pp. 2807–2822, May 2022.
[26] H. Yang et al., "Fast and accurate optical fiber channel modeling using generative adversarial network," *J. Lightw. Technol.*, vol. 39, no. 5, pp. 1322–1333, Mar. 2021.
[27] D. Wang et al., "Data-driven optical fiber channel modeling: A deep learning approach," *J. Lightw. Technol.*, vol. 38, no. 17, pp. 4730–4743, Sep. 2020.
[28] Z. Liu, W. Cai, and Z.-Q. J. Xu, "Multi-scale deep neural network (MscaleDNN) for solving Poisson-Boltzmann equation in complex domains," *Commun. Comput. Phys.*, vol. 28, no. 5, pp. 1970–2001, Nov. 2020.
[29] Z.-Q. J. Xu, Y. Zhang, T. Luo, Y. Xiao, and Z. Ma, "Frequency principle: Fourier analysis sheds light on deep neural networks," *Commun. Comput. Phys.*, vol. 28, no. 5, pp. 1746–1767, Nov. 2020.
[30] S. Zhang and R. S. Sutton, "A deeper look at experience replay," in *Proc. NIPS 2017 Deep Reinforcement Learn. Symp.*, 2017.
[31] P. Jin et al., "Quantifying the generalization error in deep learning in terms of data distribution and neural network smoothness," *Neural Netw.*, vol. 130, pp. 85–99, Oct. 2020.
[32] A. Jentzen and T. Welti, "Overall error analysis for the training of deep neural networks via stochastic gradient descent with random initialisation," *Appl. Math. Comput.*, vol. 455, Oct. 2023, Art. no. 127907.
[33] 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.
[34] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–14.