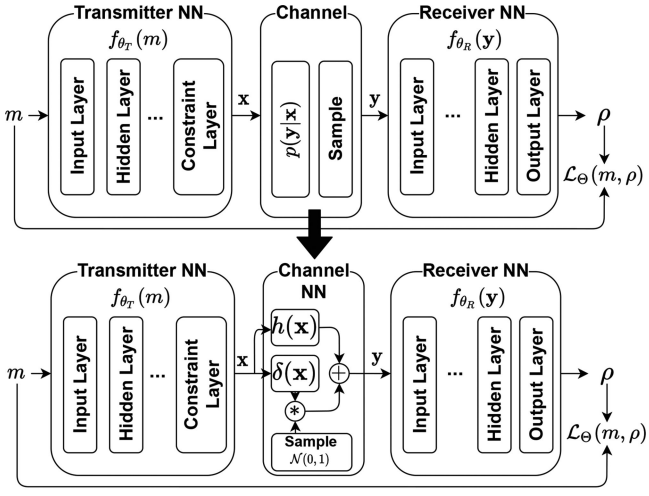
IEEE Photonics Journal
An IEEE Photonics Society Publication

Open Access

Model-Aware End-to-End Learning for SISO Optical Wireless Communication Over Poisson Channel

Volume 12, Number 6, December 2020

Ling-Han Si-Ma
Zhao-Rui Zhu
Hong-Yi Yu



Generic structure of the double nueral network and square-root autoencoder schemes over the Poisson channel

DOI: 10.1109/JPHOT.2020.3038534

Model-Aware End-to-End Learning for SISO Optical Wireless Communication Over Poisson Channel

Ling-Han Si-Ma, Zhao-Rui Zhu, and Hong-Yi Yu

Information Engineering University, Zhengzhou 450000, China

DOI:10.1109/JPHOT.2020.3038534

Manuscript received November 7, 2020; accepted November 11, 2020. Date of publication November 17, 2020; date of current version December 9, 2020. This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61901524 and 62071489, and in part by the National Natural Science Foundation for Post-doctoral Scientists China under Grant 2019M663477. Corresponding author: Hong-Yi Yu (e-mail: xxgcmaxyu@163.com).

Abstract: Faced with the challenge of transceiver design over the Poisson channel, we leverage the deep-learning technique and devise two novel end-to-end learning schemes to fulfill the design task in this paper. One of the schemes accords with the basic principle of the currently available autoencoder (AE) but is specially designed for the Poisson channel with the aid of the square root (SR) transform. The other scheme, following a different design philosophy from AE, is developed based on a double neural network (DNN) model, which regards the receiver and the transmitter as two separate networks. By these designs, the end-to-end learning task can be conducted over Poisson channel. Extensive computer simulations reveal that 1) the transceiver learned by the DNN scheme always performs better than or comparably to the currently available artificially designed transceivers, and 2) compared with the transceiver learned by DNN, the transceiver learned by SR-AE suffers performance loss in some cases, but the SR-AE scheme has a lower complexity to compute the loss function and fewer network parameters. This study takes the first step toward applying end-to-end learning techniques in the field of the Poisson channel and lays a foundation for further works on this topic.

Index Terms: Optical wireless communication, photon counter, Poisson channel, deep learning, transceiver optimization.

1. Introduction

The Poisson channel, as a type of standard channel model for optical communications [1], [2], has been widely studied for a long time [3], [4]. Recently, the discrete-time Poisson (DTP) channel has drawn much attention due to the prevalence of intensity modulation in optical wireless communication (OWC) and has been studied in various communication scenarios such as single-input single-output (SISO) [5], [6], multiple-input multiple-output [7], [8], multicarrier [9], [10] and multiuser scenarios [11], [12]. Although the mathematical model of the Poisson channel is clear and concise, the optimization problem formulated for a given communication system is always challenging and complex. This forces researchers to adopt some expedients, such as classified discussion, approximation and numerical search, to obtain suboptimal solutions or solutions with high implementation

complexity [12]–[15]. Such limitations of manpower motivate us to investigate the deep learning (DL) technique in this paper.

More recently, the success of DL in solving very complicated optimization problems has promoted the applications of various DL techniques in the communication field [16]. As one of the DL techniques, end-to-end learning follows the philosophy of joint optimization to train multiple neural network modules together in a single task. Such a training fashion caters to the need for joint transceiver optimization and leads to the popularization of end-to-end learning in current research related to transceiver design [17]–[23]. More specifically, based on whether the channel information is involved in the calculation of the training stage, end-to-end learning can be divided into two categories: model-aware and model-free. Model-aware (MA) means the channel model is available in the learning procedure of a neural network. Thus, in the MA case, the existing end-to-end learning method, called an autoencoder (AE) in DL terminology, merges the transceiver and the channel into the same neural network. Contrary to the concept of MA, model-free (MF) means the channel model is unknown in the learning procedure. Thus, various methods, such as the policy gradient method [18] and the generative adversarial network [19], have been developed to provide surrogates for the intermediate variables which are required by the learning procedure but related to the channel model [24].

For the Poisson channel, the currently available MA learning scheme (i.e., the AE) is only suitable for the Gaussian channel or its variant since the Gaussian distribution can be embedded into the network via the reparameterization trick [25]. Unfortunately, since the reparameterization trick stems from the exclusive property of Gaussian distribution, embedding the Poisson channel as a differentiable component into the AE is still an open problem. Additionally, although the MF learning schemes are not restricted by the channel types and are expected to be effective for learning over the Poisson channel, the value of developing an MA learning scheme for the Poisson channel cannot be ignored. *First*, the MA training scheme enjoys the advantages of easy-deployment and low cost compared with its MF counterpart. Even though the transceiver learned by MA may suffer performance loss due to the inaccuracy modeling of the actual channel [26], the MA learning scheme is still worth considering. *Second*, to obtain the surrogate intermediate variables, approximations are introduced in the MF schemes, which inevitably diminishes the learning effectiveness. Therefore, in the studies of MF schemes [18], [19], the MA scheme plays the role of a baseline to evaluate MF schemes. To the best of our knowledge, to date, no research has focused on the topic of end-to-end learning over the Poisson channel.

The above factors indeed motivate us to develop novel end-to-end learning schemes for the SISO DTP channel in the MA case. This research is a bedrock for further studies on more complex systems and advanced techniques in this field. Our main contributions are summarized as follows: 1) We first propose an square-root (SR) AE learning scheme to conduct the end-to-end learning over Poisson channel. This scheme accords with the design philosophy of currently available AE but introduce the SR transform trick to surmount the unavailability of an AE over the Poisson channel. 2) Instead of the design philosophy of AE, we propose a novel learning scheme called DNN which implement the transmitter and receiver by two separate networks respectively. Theoretical analysis reveals its intrinsic connection with the AE model. 3) A comparative study between the SR-AE and DNN indicates that the DNN scheme enjoys a performance advantage over the SR-AE in some cases but has a drawback in the computation complexity. 4) Different from the currently prevailing sigmoid activation function, an alternative function is proposed to embed the peak power constraint into the network; this is shown to allow the model to easily avoid the local minima and achieve high-quality results.

Notations: The following notations are used throughout this paper. Boldface upper-case letters denote matrices and the boldface lower-case letters denote vectors. $(\cdot)^{\mathrm{T}}$ denotes the transpose operation. For a given random variable, e.g. m, $\mathbb{E}_m\{\cdot\}$ denotes the expectation operation with respect to (w.r.t.) m. ∇ is the derivative operator, for example $\nabla_{\mathbf{x}} f(\mathbf{x})$ means computing the derivative of $f(\mathbf{x})$ w.r.t. $\mathbf{x}$. The letter accented by a tilde or a bar represents the approximation or the estimation of the original variable. Other main notations are listed in Table 1.

TABLE 1
List of Notations

$\mathcal{M}$	Message set
m	Message belonging to $\mathcal{M}$
$\mathcal{X}$	Transmitted constellation
$\mathbf{x}$	Transmitted symbol
$\mathbf{y}$	Received signal
N	Number of channel use
$\mathcal{N}(\mathbf{h}(\mathbf{x}), \mathbf{\Sigma}(\mathbf{x}))$	Multivariate Gaussian distribution whose mean and variance are the functions of $\mathbf{x}$
$\mathbf{v}$	Sample of a random variable following standard Gaussian distribution
$\boldsymbol{\theta}_T$,$\boldsymbol{\theta}_R$,$\boldsymbol{\Theta}$	Respective parameters of the transmitter, receiver and whole neural networks
$f_{\theta_T}(m)$ and $f_{\theta_R}(\mathbf{y})$	Transmitter neural network and receiver neural network
$f_{\theta_T}^{(SR)}(m)$ and $f_{\theta_R}^{(SR)}(\mathbf{y})$	Transceiver neural networks referring in particular to those of SR-AE scheme
$\mathcal{L}_{\boldsymbol{\Theta}}$	Expectation of the losses with the parameters $\boldsymbol{\Theta}$
ρ	likelihood vector for transmitted symbol
l	Sample of the loss
$l^{(D)}$	Sample of the loss defined for the transmitter network
W	Minimum batch size
A	Received peak photon number
η	Photon counts caused by background noise
λ	Mean of the Poisson distribution

2. System Model and Problem Statement

In this section, we introduce two kinds of neural network models used to describe a typical SISO communication system. Both of them can jointly optimize the transceiver by their corresponding training algorithm. A comparison between these two models highlights the challenge of applying the currently available AE over the DTP channel and provides insight into the design of the end-to-end learning scheme over the Poisson channel.

2.1 DNN Model

As shown in Fig. 1 (top), let us consider a DNN model for an SISO communication system over a certain channel in which the three main components, the transmitter neural network, Gaussian channel and receiver neural network, are symbolized by $f_{\theta_T}(m)$, $p(\mathbf{y}|\mathbf{x})$ and $f_{\theta_R}(\mathbf{y})$ respectively and successively connected. Specifically, a message m, $m \in \mathcal{M} = \{1, \ldots 2^K\}$ is mapped to a transmitted symbol $\mathbf{x}$ by the transmitter neural network, where K denotes the number of bits transmitted per symbol. Assume that N time-intervals (channels) are occupied to transmit the symbol $\mathbf{x}$ in a nonnegative real-number space with a peak-power constraint. Then, for each symbol $\mathbf{x}$, the channel component outputs a received signal $\mathbf{y}$ sampled from a random variable whose conditional probability mass function (PMF) or probability density function (PDF) is denoted by $p(\mathbf{y}|\mathbf{x})$. The task of the receiver neural network is to output the likelihood $\boldsymbol{\rho} = [\rho_1, \ldots, \rho_M]^{\mathrm{T}}$ for each symbol in $\mathcal{M}$ based on $\mathbf{y}$. Let $\boldsymbol{\theta}_T$ and $\boldsymbol{\theta}_R$ separately denote the parameter vectors of the transceiver neural networks. Assume that $f_{\theta_T}(m)$ and $f_{\theta_R}(\mathbf{y})$ are differentiable w.r.t. $\boldsymbol{\theta}_T$ and $\boldsymbol{\theta}_R$ respectively, and that the categorical cross-entropy $l(\boldsymbol{\rho}, m) = -\log(\rho_m)$ is adopted as the loss function. Finally, the desired transceiver optimization problem can be formulated as follows [19]:

$$\min_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}}(m, \boldsymbol{\rho}) \triangleq \min_{\boldsymbol{\Theta}} \mathbb{E}_m \left\{ \int l(f_{\theta_R}(\mathbf{y}), m)\, p(\mathbf{y}|f_{\theta_T}(m)) \mathrm{d}\mathbf{y} \right\} \tag{1}$$

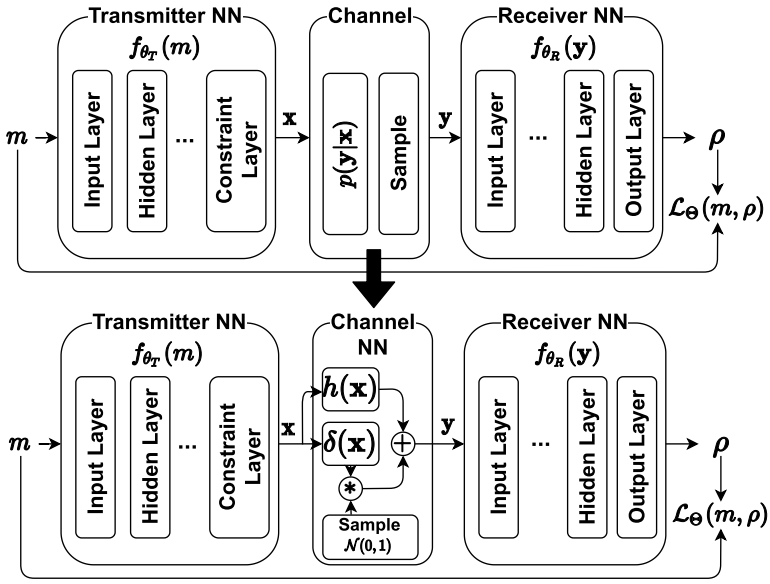


Fig. 1. Structural transformation from the DNN model to the AE model over the Gaussian channel.

where $\rho = f_{\theta_R}(\mathbf{y})$, and $\Theta = [\theta_T^\mathsf{T}, \theta_R^\mathsf{T}]^\mathsf{T}$ constitutes the parameter space of this optimization problem. In the field of DL, this optimization is conducted by gradient descent or a variant w.r.t Θ, i.e., by updating Θ according to the direction of $\nabla_\Theta \mathcal{L}_\Theta$. It can be noticed that, according to (1), the DNN model can be applied over an arbitrary type of channel as long as the derivative is available. This flexible adaptability is an advantage of the DNN model. However, instead of the DNN model, the AE is currently a popular method for MA end-to-end learning and has a strong association with DNN.

2.2 AE Model

By regarding the entire communication system as a single neural network, the AE model gains the advantage of executing an end-to-end back propagation over the whole system [27]. Although, in the way of constructing the neural network, the AE model is quite different with the above introduced DNN model, there is a strong connection between these two models. As depicted in Fig. 1 (bottom), the key to transforming the DNN model to an AE is applying the reparameterization trick over the Gaussian distribution [25], namely, if the conditional PDF of the channel follows a general Gaussian distribution $\mathcal{N}(\mathbf{h}(\mathbf{x}), \mathbf{\Sigma}(\mathbf{x}))$ whose mean $\mathbf{h}(\mathbf{x})$ and variance $\mathbf{\Sigma}(\mathbf{x})$ are the element-wise functions of $\mathbf{x}$, the DNN model can be transformed to an AE by the trick as follows:

$$\mathbf{y} \triangleq f_{\mathbf{v}}(\mathbf{x}) = \mathbf{h}(\mathbf{x}) + \sqrt{\mathbf{\Sigma}(\mathbf{x})}\mathbf{v}, \tag{2}$$

where $\mathbf{v}$ is a sample of an N-ary random vector obeying the standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This equation actually reflects a well-known property that an arbitrary normal distribution can be obtained by scaling and translating a standard normal distribution. Then, with the assumption that both $\mathbf{h}(\mathbf{x})$ and $\mathbf{\Sigma}(\mathbf{x})$ are differentiable w.r.t $\mathbf{x}$, (2) and (1) are further combined to obtain:

$$\mathcal{L}_\Theta(m, \rho) \triangleq \mathbb{E}_m \left\{ \int l(f_{\theta_R}(f_{\mathbf{v}}(f_{\theta_T}(m))), m) p(\mathbf{v}) \mathrm{d}\mathbf{v} \right\} = \mathbb{E}_{m,\mathbf{v}} \left\{ l(f_\Theta(m), m) \right\}, \tag{3}$$

where $f_\Theta(m) \triangleq f_{\theta_R}(f_{\mathbf{v}}(f_{\theta_T}(m)))$ for notational convenience. Then, the optimization of (3), i.e., $\min_\Theta \mathcal{L}_\Theta$, is conducted by gradient descent or its variant w.r.t Θ.

2.3 Problem Statement

Although, (1) or (3) can lead to an analytical expression for the gradient, $\nabla_{\boldsymbol{\Theta}}\mathcal{L}_{\boldsymbol{\Theta}}$, numerically evaluating such an expression is computationally expensive. Thanks to the chain structure of the network, a complex gradient computation can be done by recursively applying the chain rule. This computation method is commonly known as back propagation [27, Ch. 6]. In this pattern, each layer of the network needs to compute their own derivative. For this reason, to realize an AE scheme over the Poisson channel, we need to not only embed the Poisson channel into the network but also to make sure the output of each layer in the network differentiable w.r.t its own input [19]. Unfortunately, these two targets are impeded by the following obstacles.

1) *Unavailable reparameterization trick:* Supported by the property described by (2), this trick is only effective for the Gaussian channel. As far as we know, the Poisson distribution does not hold this property and thus can not be embedded into the network.
2) *Discrete output:* Even if the Poisson channel could be embedded as a layer into the network, the discrete output of the Poisson layer will make the subsequent layer non-differentiable.

These challenges indeed motivate us to devise novel end-to-end learning schemes over the Poisson channel. Details of our proposed schemes are provided in the following section.

3. Theoretical Analysis

In this section, we consider the case where the channel model is available in the training process and develop two kinds of end-to-end learning schemes for communication systems under the Poisson channel.

3.1 Square-Root (SR) AE Scheme

Let us consider a specific SISO DTP channel

$$p(\mathbf{y}|\mathbf{x}) = \prod_{n=1}^{N} \frac{(Ax_n + \eta)^{y_n}}{y_n!} e^{-Ax_n - \eta} \tag{4}$$

in which A, η and $\mathbf{1}_N$ respectively denote the received peak photon number, the photon counts caused by background noise and an N-ary vector whose entries are all one. Then, we attempt to establish an AE model for this channel with the aid of the square-root (SR) variance-stabilization transform [28] whose property is given below.

Property 1: For a random variable $\mathscr{Y}$ whose PMF satisfies $y = e^{-\lambda}\frac{\lambda^y}{y!}$, its square root $\sqrt{\mathscr{Y}}$ asymptotically obeys the Gaussian distribution as $\lambda \to \infty$, say, $\sqrt{\mathscr{Y}} \sim \sqrt{\lambda} + \mathscr{V}$, where $\mathscr{V} \sim \mathcal{N}(0, \frac{1}{4})$.

Let $\mathbf{y}^{(\mathrm{SR})}$ stand for the element-wise square root of $\mathbf{y}$, where $\mathbf{y}$ is the received signal from (4). As revealed by Property 1, $\mathbf{y}^{(\mathrm{SR})}$ can be regarded as a sample from the following SR Gaussian channel:

$$p(\mathbf{y}^{(\mathrm{SR})}|\mathbf{x}) = (2/\pi)^{N/2}\, e^{-2\left\|\mathbf{y}^{(\mathrm{SR})} - \sqrt{A\mathbf{x}+\eta\mathbf{1}_N}\right\|_2^2}, \tag{5}$$

where $\sqrt{\cdot}$ denotes an element-wise square root operation when this operator acts on a vector or metric. Obviously, by the reparameterization trick (2), the SR Gaussian channel (5) can be further expressed as

$$\mathbf{y}^{(\mathrm{SR})} \triangleq f_{\mathbf{v}}(\mathbf{x}) = \sqrt{A\mathbf{x} + \eta\mathbf{1}_N} + \sqrt{\boldsymbol{\Sigma}}\mathbf{v}, \tag{6}$$

which is a special case of (2) with $\mathbf{h}(\mathbf{x}) = A\mathbf{x} + \eta\mathbf{1}_N$ and $\boldsymbol{\Sigma}(\mathbf{x}) = \frac{1}{4}\mathbf{I}$. Then, let $f_{\boldsymbol{\theta}_T}^{(\mathrm{SR})}(m)$ and $f_{\boldsymbol{\theta}_R}^{(\mathrm{SR})}(\mathbf{y}^{(\mathrm{SR})})$ represent the transmitter and receiver components in the SR-AE network respectively. (6) leads us to the loss function defined by (3), where $f_{\boldsymbol{\Theta}}(m) = f_{\boldsymbol{\theta}_R}^{(\mathrm{SR})}(\sqrt{Af_{\boldsymbol{\theta}_T}^{(\mathrm{SR})}(m) + \eta\mathbf{1}_N} + \sqrt{\boldsymbol{\Sigma}}\mathbf{v})$. Then, by the law of large numbers, the mean in (3) can be estimated by a batch of samples as follows:

$$\widetilde{\mathcal{L}_{\boldsymbol{\Theta}}} = \frac{1}{W}\sum_{w=1}^{W} l(f_{\boldsymbol{\Theta}}(m^{(w)}), m^{(w)}), \tag{7}$$

Algorithm 1: Communication Over the Poisson Channel (SR-AE).

1: ▷ Transmitter:
2: $\mathbf{x} = f_{\theta_T}^{(\mathrm{SR})}(m) \leftarrow$ message: $m \in \mathbb{M} = \{1, \ldots, M\}$
3: ▷ Poisson channel:
4: Generate $\mathbf{y}$ from $p(\mathbf{y}|\mathbf{x})$ (4) $\leftarrow$ transmitted signal: $\mathbf{x}$
5: ▷ Receiver:
6: $\mathbf{y}^{(\mathrm{SR})} = \sqrt{\mathbf{y}} \leftarrow$ received signal: $\mathbf{y}$
7: $\boldsymbol{\rho} = f_{\theta_R}^{(\mathrm{SR})}(\mathbf{y}^{(\mathrm{SR})}) \leftarrow$ transformed signal: $\mathbf{y}^{(\mathrm{SR})}$
8: Gain estimate of m:
$\hat{m} = \arg\max_{m \in \mathbb{M}} \rho_m \leftarrow$ likelihood vector: $\boldsymbol{\rho}$

where W is the number of samples in a batch and $m^{(w)}$ stands for the wth training sample. Finally, we can easily compute the gradient by $\nabla_{\boldsymbol{\Theta}} \widetilde{\mathcal{L}_{\boldsymbol{\Theta}}}$ and train the AE over SR Gaussian channel by a usual optimizer such as stochastic gradient descent (SGD) or its variant [27, Ch. 8].

Owing to Property 1, after the learning procedure is completed, we can use the obtained transceivers, $f_{\theta_T}^{(\mathrm{SR})}(m)$ and $f_{\theta_R}^{(\mathrm{SR})}(\mathbf{y}^{(\mathrm{SR})})$, over the Poisson channel as long as the received signal $\mathbf{y}$ is processed by the SR transform before being input into $f_{\theta_R}^{(\mathrm{SR})}(\mathbf{y}^{(\mathrm{SR})})$. The specific communication process of the SR-AE is detailed below.

3.2 DNN Scheme

In this subsection, our main purpose is to develop a training method based on the DNN model over Poisson channels. The DNN model has the same goal as that of AE (minimizing $\mathcal{L}_{\boldsymbol{\Theta}}$ w.r.t $\boldsymbol{\Theta}$) but requires a different algorithm because its loss function $\mathcal{L}_{\boldsymbol{\Theta}}$ is defined in a distinct way. This can be confirmed by comparing (1) and (3). For the DNN scheme, a simple and effective method for optimizing the parameters of the networks is applying gradient descent, which presupposes the gradient of the loss function, i.e., $\nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}} = [(\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}})^{\mathsf{T}}, (\nabla_{\theta_R} \mathcal{L}_{\boldsymbol{\Theta}})^{\mathsf{T}}]^{\mathsf{T}}$, where $\mathcal{L}_{\boldsymbol{\Theta}}$ is defined by (1). The gradient, $\nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}}$, is divided into two parts, $\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}}$ and $\nabla_{\theta_R} \mathcal{L}_{\boldsymbol{\Theta}}$, deliberately for convenient analysis. Assume that $p(\mathbf{y}|\mathbf{x})$ is differentiable w.r.t. $\mathbf{x}$. According to (1), the first part of the gradient $\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}}$ can be expressed as

$$\begin{aligned}
\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}} &= \mathbb{E}_m \left\{ \int l(f_{\theta_R}(\mathbf{y}), m) \nabla_{\theta_T} p(\mathbf{y}|f_{\theta_T}(m)) \mathrm{d}\mathbf{y} \right\} \\
&= \mathbb{E}_m \left\{ \int l(f_{\theta_R}(\mathbf{y}), m) \nabla_{\theta_T} \log p(\mathbf{y}|f_{\theta_T}(m)) p(\mathbf{y}|f_{\theta_T}(m)) \mathrm{d}\mathbf{y} \right\} \\
&= \mathbb{E}_{m,\mathbf{y}} \left\{ l(f_{\theta_R}(\mathbf{y}), m) \left[\nabla_{\theta_T} f_{\theta_T}(m) \right]^{\mathsf{T}} \nabla_{\mathbf{x}=f_{\theta_T}(m)} \log p(\mathbf{y}|\mathbf{x}) \right\}, \qquad (8)
\end{aligned}$$

in which the first equality comes from the definition of $\mathcal{L}_{\boldsymbol{\Theta}}$ (1), the second equality holds due to the log-trick $\nabla_{\mathbf{z}} u(\mathbf{z}) = u(\mathbf{z}) \nabla_{\mathbf{z}} \log(u(\mathbf{z}))$, and the third equality leverages the expectation formula and the chain rule. Again, $\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}}$ can be estimated by

$$\begin{aligned}
\widetilde{\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}}} &= \frac{1}{W} \sum_{w=1}^{W} l(f_{\theta_R}(\mathbf{y}^{(w)}), m^{(w)}) \left[\nabla_{\theta_T} f_{\theta_T}(m^{(w)}) \right]^{\mathsf{T}} \nabla_{\mathbf{x}^{(w)}} \log p(\mathbf{y}^{(w)}|\mathbf{x}^{(w)}) \\
&= \nabla_{\theta_T} \frac{1}{W} \sum_{w=1}^{W} l(f_{\theta_R}(\mathbf{y}^{(w)}), m^{(w)}) \left[f_{\theta_T}(m^{(w)}) \right]^{\mathsf{T}} \nabla_{\mathbf{x}^{(w)}} \log p(\mathbf{y}^{(w)}|\mathbf{x}^{(w)}) \\
&\triangleq \nabla_{\theta_T} \frac{1}{W} \sum_{w=1}^{W} l^{(D)}(\mathbf{y}^{(w)}, \mathbf{x}^{(w)}, m^{(w)}) \qquad (9)
\end{aligned}$$

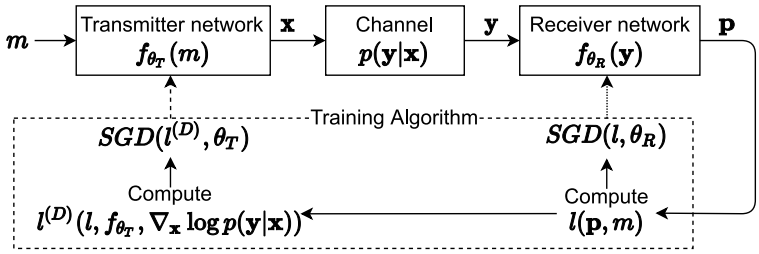


Fig. 2. Flowchart of the training algorithm for the DNN under the MA case.

Algorithm 2: DNN Training Algorithm.

1: **While:**
2: ▷ Transmitter:
3: $\mathbf{X}_t = f_{\theta_T}(\mathbf{m}_t) \leftarrow$ training samples: $\mathbf{m}_t \in \mathbb{M}^W$
4: Transmit($\mathbf{X}_t$)
5: ▷ Poisson channel:
6: Sample $\mathbf{Y}_t$ from $p(\mathbf{Y}_t|\mathbf{X}_t) \leftarrow \mathbf{X}_t$
7: ▷ Receiver:
8: $\mathbf{P}_t = f_{\theta_R}(\mathbf{Y}_t) \leftarrow$ received signals: $\mathbf{Y}_t$
9: $\mathbf{l}(\mathbf{P}_t, \mathbf{m}_t), \mathbf{l}^{(D)}(\mathbf{P}_t, \mathbf{Y}_t, \mathbf{X}_t, \mathbf{m}_t) \leftarrow \mathbf{P}_t, \mathbf{Y}_t, \mathbf{X}_t, \mathbf{m}_t$
10: Training f_{θ_R} by SGD($\theta R, \mathbf{l}$) $\leftarrow \theta R, \mathbf{l}$
Training f_{θ_T} by SGD($\theta T, \mathbf{l}^{(T)}$) $\leftarrow \theta T, \mathbf{l}^{(D)}$
11: **Break:** If stop condition is true

Notations: $\mathbf{m}_t$, $\mathbf{X}_t$, $\mathbf{Y}_t$ and $\mathbf{P}_t$ respectively denote a batch of samples from the variables, m, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{p}$, e. g., $\mathbf{m}_t$ consists of W samples from the message m, i.e., $\mathbf{m}_t = [m_1, \ldots, m_W]^{\mathrm{T}}$

where $\nabla_{\theta_T} f_{\theta_T}(m^{(w)})$ denotes the Jacobian of the function $f_{\theta_T}(m^{(w)})$ w.r.t θ_T and

$$l^{(D)}(\mathbf{y}^{(w)}, \mathbf{x}^{(w)}, m^{(w)}) \triangleq l(f_{\theta_R}(\mathbf{y}^{(w)}), m^{(w)}) \left[f_{\theta_T}(m^{(w)})\right]^{\mathrm{T}} \nabla_{\mathbf{x}^{(w)}} \log p(\mathbf{y}^{(w)}|\mathbf{x}^{(w)}). \tag{10}$$

$l^{(D)}(\mathbf{y}^{(w)}, \mathbf{x}^{(w)}, m^{(w)})$ can be viewed as a special loss defined for the transmitter network and $\nabla_{\mathbf{x}^{(w)}} \log p(\mathbf{y}^{(w)}|\mathbf{x}^{(w)})$ is determined by the distribution of the specific channels.

For the second part of the gradient, $\nabla_{\theta_R} \mathcal{L}_{\boldsymbol{\Theta}}$, we have

$$\nabla_{\theta_R} \mathcal{L}_{\boldsymbol{\Theta}} = \mathbb{E}_m \left\{ \int \nabla_{\theta_R} l(f_{\theta_R}(\mathbf{y}), m) p(\mathbf{y}|f_{\theta_T}(m)) \mathrm{d}\mathbf{y} \right\} = \mathbb{E}_{m,\mathbf{y}} \left\{ \nabla_{\theta_R} l(f_{\theta_R}(\mathbf{y}), m) \right\}, \tag{11}$$

Then, estimate (11) by sampling

$$\widetilde{\nabla_{\theta_R} \mathcal{L}_{\boldsymbol{\Theta}}} = \frac{1}{W} \sum_{w=1}^{W} \nabla_{\theta_R} l\left(f_{\theta_R}(\mathbf{y}^{(w)}), m^{(w)}\right) \tag{12}$$

Thus, combining (9) and (12), we can finally obtain the estimate of $\nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}}$, i.e., $[(\widetilde{\nabla_{\theta_T} \mathcal{L}_{\boldsymbol{\Theta}}})^{\mathrm{T}}, (\widetilde{\nabla_{\theta_R} \mathcal{L}_{\boldsymbol{\Theta}}})^{\mathrm{T}}]^{\mathrm{T}}$. The proposed training method is illustrated in Fig. 2 and detailed in Algorithm 2.

Remark 1: By the comparison between the training algorithms of the DNN and SR-AE schemes, we notice that the former needs to compute two loss function as defined by (9) and (12) but the later only computes one loss function (7); this highlights the low-complexity advantage of the SR-AE scheme in computing the loss function.

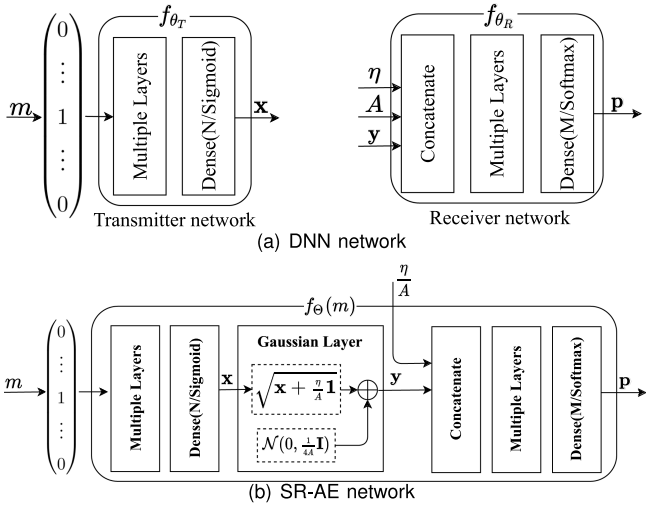


Fig. 3. Generic structure of the DNN and SR-AE schemes over the Poisson channel.

3.3 Neural Network Configuration

In this section, we detail the generic structure of the above introduced models. Assume that A and d are available at the receiver side, which is a common assumption in studies related to maximum likelihood detection (MLD) over Poisson channels and can be realized by pilot techniques since they are almost invariant for thousands of successive symbols [11]–[14].

1) *DNN:* We only adopt dense layers to comprise the transmitter and receiver neural networks. As shown in Fig. 3, we feed the one-hot version of the message m into the transmitter network. Determined by the system constraints of the normalized peak-power and the number of channel use, the last layer is an N-unit dense layer with sigmoid activation, which is the most common activation function for embedding the peak-limited constraint into the network [22], [29]. The last layer of the receiver network is also fixed as an M-unit dense layer with softmax activation, enabling the receiver network to deal with M-category classification (i.e., signal detection). Additionally, a concatenation layer is used before the dense layers in the receiver network to concatenate the three inputs together. The other layers, briefly represented by 'multiple layers' in Fig. 3, can be arbitrarily adjusted according to the specific situation.
2) *SR-AE:* The architecture of the SR-AE is illustrated in Fig. 3. It can also be noticed that the structure of the SR-AE is basically consistent with that of the DNN except for the embedded channel layer and the concatenate layer. Specifically, the channel layer can be implemented by the built-in Gaussian layer of Keras.

Remark 2: Owing to the decoupling effect caused by the SR transform between the signal and noise, only two variables, $\mathbf{y}$ and $\frac{\eta}{A}$, are input into the concatenate layer. Assume U units in the layer subsequent to the concatenate layer for both the DNN and SR-AE networks, which means that, in the U-unit layer, the DNN network has a $3 \times U$ parameter matrix while the SR-AE network has a $2 \times U$ parameter matrix. Thus, the AE model has an advantage of using fewer parameters.

4. Simulation Results

In this section, our main purpose is to evaluate the performance of the above algorithm by simulations.

TABLE 2
Specific Network Structures of the AE and DNN Schemes for $K = 4, N = 2$

AE	
Layer	Number of units
Dense+Relu	16
Dense+Sigmoid	2
AWGN	2
Concatenation	3
Dense+Relu	16
Dense+Softmax	16

DNN		
Layer		Number of units
Transmitter network	Dense+Relu	16
	Dense+Sigmoid	2
Receiver network	Concatenation	4
	Dense+Relu	16
	Dense+Softmax	16

4.1 Evaluation Over Poisson Channels

In our simulations, intensity modulation and the peak-power constraint are considered, which leads to the normalized transmitted symbols restricted into an N-dimensional nonnegative unit hypercube. Performance comparisons are undertaken between the transceivers respectively learned by the SR-AE and DNN schemes. In the rest of this section, these learned transceivers are called an SR-AE transceiver and a DNN transceiver respectively for convenience. All of the simulations are conducted with the following settings.

- *Simulation settings:* We evaluate both of the SR-AE and DNN transceivers over the Poisson channel. The squared pulse amplitude modulation (SPAM) [13] and (7,4) Hamming code are set as the respective baselines for the cases of and $K = 4, N = 7$. For $N = 2$, the normalized SPAM constellation is given by $\mathbf{x} \in \mathcal{X} = \{\frac{1}{9}(i^2, j^2) | 0 \le i \le 3, 0 \le j \le 3\}$. The corresponding MLD is defined as $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{X}} p(\mathbf{y}|\mathbf{x})$ where $p(\mathbf{y}|\mathbf{x})$ is defined by (4) and $\hat{\mathbf{x}}$ denotes the estimate of $\mathbf{x}$.

In what follows, we will provide and analyze the specific simulation results.

4.1.1 Case 1: $K = 4, N = 2$: In this case, we train the networks with two phases and Adam optimizer [27, Ch. 8] whose learning rate is set to be 0.01. In the two phases, we set the collocation, batchsize\epoch, as 64\15 and 1000\10 respectively. The structures of the SR-AE and DNN schemes are provided in Table 2, where the depth and width of the networks are empirically determined, such that increasing the depth and width cannot further promote the learning performance. Besides, for the SR-AE scheme, the parameters of the Gaussian layer are set according to Fig. 3. These two learning schemes are trained with various transmitted peak optical power, $P = -85, -84, -83$ dBm, since Poisson noise is signal-dependent. Other fixed parameters, including background noise count rate, photon detection efficiency, symbol duration per channel use and energy of single photon, are denoted by N_{bcr}, C, T, and E_p; and set to be 7.27×10^3, 20%, 10^{-4} and 4.42×10^{-19} respectively. These parameters determine the values of A and η by the equalities $A = CPT/E_p$ and $\eta = N_{dcr}T$ [13]. In particular, for Poisson channel, the loss $l^{(D)}$ of DNN (10) can be written as

$$l^{(D)} = l(f_{\theta_R}(\mathbf{y}^{(w)}), m^{(w)}) \left[f_{\theta_T}(m^{(w)})\right]^{\mathsf{T}} \left(A\mathbf{y}^{(w)} \oslash (A\mathbf{x}^{(w)} + \eta \mathbf{1}_N) - A\mathbf{1}_N\right) \tag{13}$$

in which $\oslash$ denotes the element-wise division operation and $l^{(D)}$ will be used in Algorithm 2 to train the transmitter network of the DNN scheme. After obtaining the transceivers from the proposed two learning schemes, we evaluate the block error rate (BLER) of the transceivers on the same test set for various peak optical power. The simulation results are illustrated in Fig. 4, where the BLER of the SPAM with MLD is appended as a baseline. In the high optical power regime, the performances of the AE and DNN transceivers are almost consistent with each other and slightly better than that of SPAM with MLD. However, In the low optical power regime, we can observe an evident performance loss of the AE transceiver; this can be explained by Property 1. Low optical power causes the parameter λ so small that it hinders the effectiveness of SR transform. Additional results of comparing the constellations of the learned transceivers with that of SPAM are shown in Fig. 5. It can be noticed that those constellations of the learned transceivers are graphically similar to that of SPAM. As it can be seen, the distance between arbitrary two neighbour points of these constellations increasing with the their power; such nonuniform constellation structure

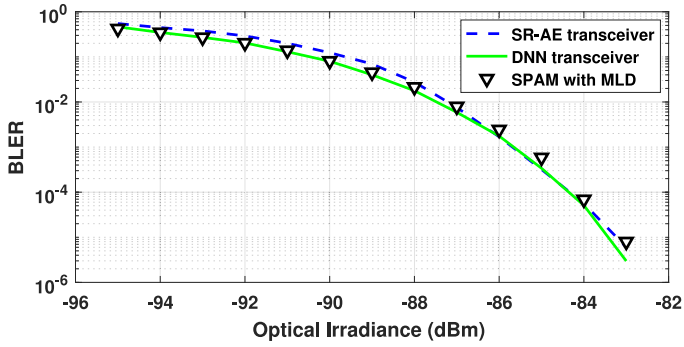
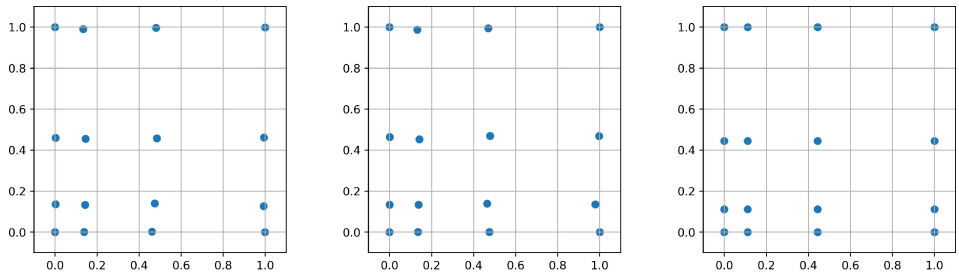
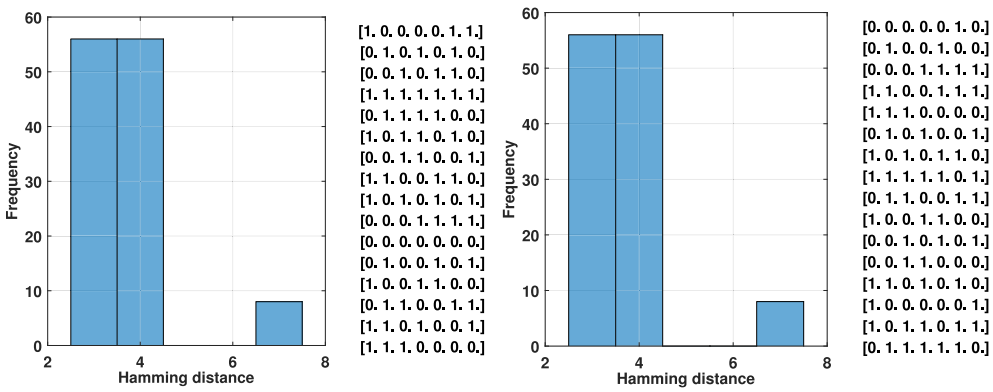


Fig. 4. BLER of the leaned transceivers and the SPAM with MLD.



(a) DNN (b) SR-AE (c) SPAM

Fig. 5. Constellations of the learned transceiver and the SPAM.



(a) Linear codeword (b) Nonlinear codeword

Fig. 6. The codebooks and distance spectrum for the learned two kinds of code with the BN+RL activation.

benefits to resist the signal-dependent Poisson noise. The above results indict the effectiveness of our proposed two end-to-end learning schemes over Poisson channel.

4.1.2 Case 2: $K = 4, N = 7$: Interestingly, as it can be seen in Fig. 7, when the proposed learning schemes are trained with $K = 4, N = 7$ and the activation functions in the last layer of their transmitter networks being sigmoid functions, the finally learned DNN and SR-AE transceivers perform worse than the traditional transceiver, i.e., the (7, 4) Hamming code with MLD. Although one could

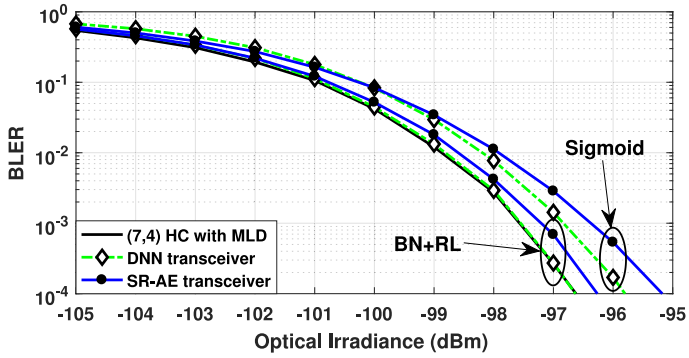


Fig. 7. BLER of the (7,4) Hamming code (HC) with MLD and the transceivers learned by the proposed learning schemes with various activation functions, i.e. the currently available sigmoid and the proposed BN+RL.

TABLE 3
Specific Network Structures of the AE and DNN Schemes for $K = 4, N = 7$

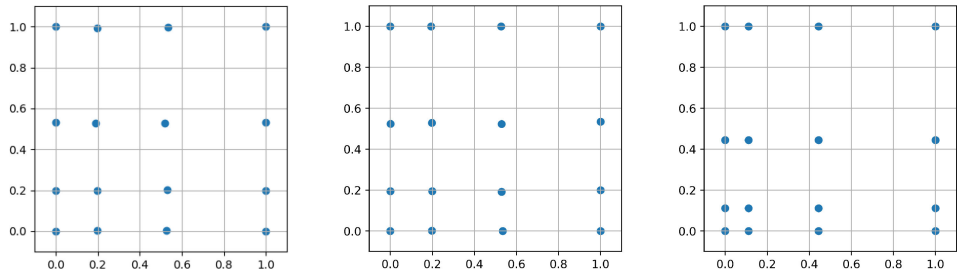
AE	
Layer	Number of units
Dense+Relu	70
Dense+Relu	70
Dense+Relu	35
Dense+BN+RL (or Dense+sigmoid)	7
AWGN	7
Concatenation	8
Dense+BN+Relu	70
Dense+Relu	70
Dense+Relu	35
Dense+Softmax	16

DNN		
	layer	Number of units
Transmitter network	Dense+Relu	70
	Dense+Relu	70
	Dense+Relu	35
	Dense+BN+RL (or Dense+sigmoid)	7
Receiver network	Concatenation	9
	Dense+BN+Relu	70
	Dense+Relu	70
	Dense+Relu	35
	Dense+Softmax	16

easily envision that the performance decisive criterion of the code over Poisson channel may differ from that of Gaussian, i.e., Hamming distance criterion; current coding theory is inadequate to provide an allied "distance criterion" of Poisson channel. Here we do not claim any optimality of the (7, 4) Hamming code for our considered case, but it indeed performs best according to our current experiments. We find that the learned constellations usually consist of 16 binary vectors but none of them has the same performance and code distance as the (7, 4) Hamming code. Such phenomenon also appears in our previous work [29] and results in about 1.5 dB performance loss compared with the (7,4) Hamming code with MLD. This dilemma should be blamed to the vanishing gradient problem of the sigmoid activation, which is located in the last layer of the transmitter component and leads to the update cessation of the transmitter network when it learns a kind of binary permutation. To alleviate this problem, we introduce the following improvements:

1) A combination of the batch normalization (BN) and the linear rectification (LR) function is adopted to replace the sigmoid activation in the last layer of transmitter network. The function of these two components can be expressed as $x_{out} = \mathbf{BN}(x_{in})$ and $x_{out} = \mathbf{LR}(x_{in}) = 0.5 \min(\max(x_{in} + 1, 0), 2)$, where $\mathbf{BN}(\cdot)$ denotes the batch normalization (BN) operation without moving average, offset and scaling [30];
2) We embed another BN layer as default of Keras before the Relu activation of the first layer in the receiver component since its inputs have large dynamic range and stabilizing them by BN will benefit the training convergence;
3) We increase the number of layers and neurons of network since enlarging the size of network decreases the probability of finding a 'bad' local minimum [31].

Specific network structures of the proposed learning schemes are listed in Table 3, where two kinds of activation functions (i.e., BN+RL and sigmoid) are considered in last layer of transmitter networks. The training needs more epochs than before due to the increment of the network



(a) DNN (b) SR-AE (c) SPAM

Fig. 8. Constellations of the learned transceiver and the SPAM with $\eta = 7$.

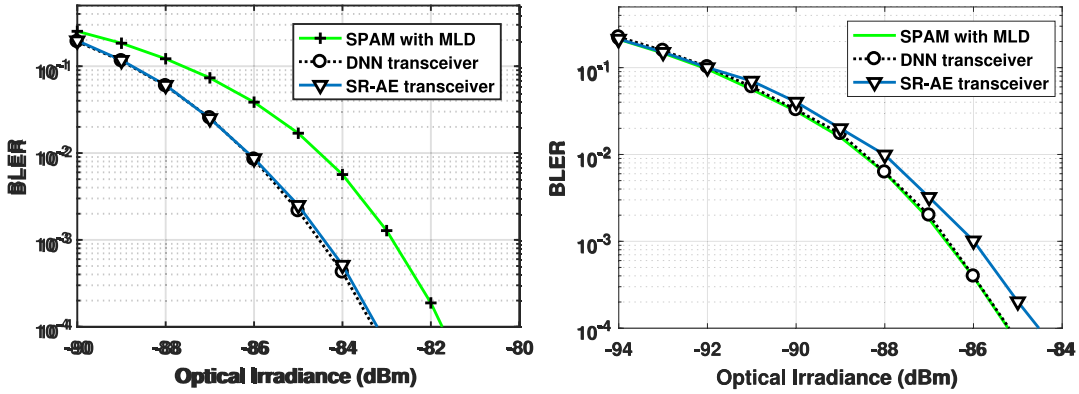
scale. In the learning procedure of the proposed schemes, their networks are trained with three phases and various learning rate of Adam optimizer. The collocation set for the three training phases, batchsize\epochs\learning-rate, are 200\15\0.03, 1000\15\0.01 and 10000\15\0.01 respectively.The simulation results are illustrated in Figs. 6 and 7, which mainly include the results for the case where the proposed BN+RL activation is adopted. Additionally, simulation results for the case of sigmoid activation are also added in Fig. 7 as performance references. As shown in Figs. 6, the proposed schemes learn two kinds of code, i.e., the linear and nonlinear codes. As it can be seen, only the linear code has the same codebook as the (7, 4) Hamming code; nevertheless, the nonlinear code can also achieve the same performance as the (7, 4) Hamming code due to the identity between their distance spectrum. The BLER curves of the learned transceivers and the traditional transceiver, i.e., the (7, 4) Hamming code and MLD, are illustrated in Fig. 7. It shows that the DNN transceiver has the same performance as the (7, 4) Hamming code with MLD but outperforms the AE transceiver; this is because the AE transceiver has the same codebook as the DNN transceiver but suffers a performance loss induced by the SR transform. A further discussion of this loss will be hold in the next subsection.

4.2 The Convergence of SR Transform

Obviously, the performance of the SR-AE scheme will be strongly influenced by the convergence of the SR approximation. In light of the Property 1, this convergence will be guaranteed by enlarging $\lambda = Ax + \eta$. One can always increase λ by enlarging the transmitted power when x is not zero but will be stranded for the case of $x = 0$. Unfortunately, in a usual case, a constellation excluding zero point is not energy-efficient. Thus, the system parameters T and N_{bcr} become the decisive factors of the performance since $\lambda = \eta$ when $x = 0$. To further validate this, we simulate the performance of SR-AE and DNN with various η and $N = 2$, $K = 4$ in the following subsection, where the SPAM with MLD is set as a baseline. The training setups except N_{bcr} remain unchanged as before. For different N_{bcr}, we first train the networks and then carry out the BLER simulation of the learned transceiver.

4.2.1 Case 1: $\eta = 7$: Without loss of generality, let $T = 10^{-4}$ and $N_{bcr} = 7 \times 10^4$. The constellations learned by the proposed schemes are compared with the constellation of SPAM in Fig. 8 and appear quite different. In Fig. 9, the DNN and SR-AE transceivers evidently outperform the SPAM with MLD; since, in order to make the design problem tractable, the SPAM constellation is obtained under the assumption that the background noise count can be ignored. Accordingly, the SPAM dose not perform well in the case of $\eta = 7$. Furthermore, the BLER comparisons in Fig. 9 also show the performance similarity between SR-AE and DNN for large background noise, which accords with the previous expectation.

4.2.2 Case 2: $\eta = 0.01$: Without loss of generality, we simulate the performances of the SR-AE and DNN schemes with $T = 10^{-4}$ and $N_{bcr} = 100$. The results are illustrated in Fig. 9. Not surprisingly, the error performance of SR-AE transceiver worse than that of the DNN transceiver;



(a) $\eta = 7$ (b) $\eta = 0.01$

Fig. 9. BLER of the leaned transceivers and the SPAM with MLD for given background noise.

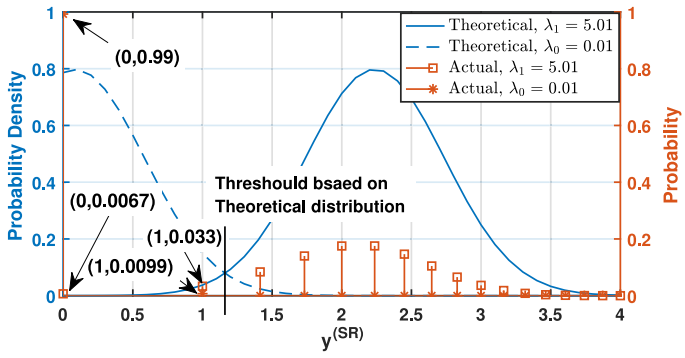


Fig. 10. Actual probability distribution and theoretical PDF of the SR transformed variable $y^{(SR)}$.

and the low background noise also benefits the SPAM to deliver the same performance as the DNN. Based on the current clues, we have known that 1) in the case of λ small, the theoretically assumed probability distribution of SR transform's output will deviate from the actual distribution; 2) this derivate performs differently depending on whether the symbol includes nonzero points since the noise is signal dependent. Thus, the remaining question is how this deviation reflects on the different symbols of a given constellation by the error performance. To answer this question, we study the following hypothesis detection for given constellation $\mathcal{S} = \{0, 1\}$. Consider a binary-input binary-output DTP channel and denote the average number of the counted photons by λ_0 if 0 is transmitted, otherwise, by λ_1. Additionally, let Pe(1|0) and Pe(0|1) denote the probabilities of mistaking 0 to 1 and mistaking 1 to 0 respectively. Then, for $\lambda_0 = \eta = 0.01$, assume that $\lambda_1 = 5.01$. Then, the SR transformed variable $y^{(SR)}$ is assumed to be Gaussian-distributed according to Property 1. Both of the theoretical PDF and the actual probability distribution is illustrated in Fig. 10. Based on the actual probability, an optimal threshold should be allocated in the interval [0,1] such that Pe(1|0) = 0.01 and Pe(0|1) = 0.0067. However, from the perspective of the theoretical PDF, the optimal threshold, as shown in Fig. 10, is situated on the right side of 1; this causes the total error probability rises up due to the fact that Pe(1|0) decreases to 0.0001 and Pe(0|1) increases to 0.0397. For the same reason, as it is shown in Fig. 11, an evident increment of the error numbers at the positions where the symbol includes the point nearest to zero in the error histogram of SR-AE transceiver. These error numbers are counted up over 10^6 samples with $P = -86$ dBm. The results in Fig. 11 indicate that the performance loss of the SR-AE transceiver mainly originates from the error caused by the transmitted symbol including the near-zero point.

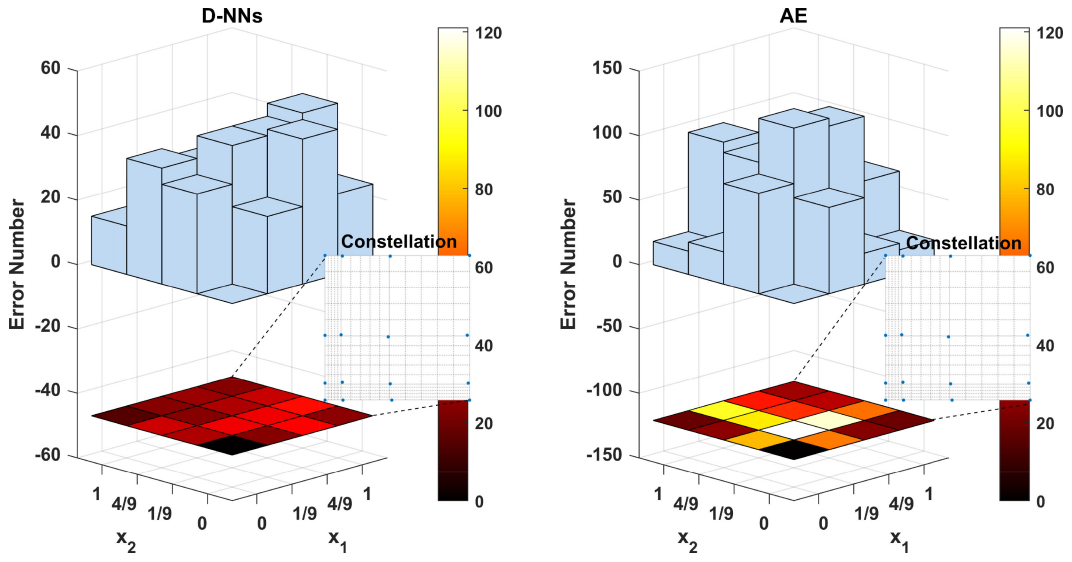


Fig. 11. The respective histograms of the error numbers for each symbol in the constellations shown by the inserted thumbnails.

5. Conclusion

In this paper, we develop two novel end-to-end learning schemes over the SISO DTP channel. First, following the design philosophy of the AE, we propose an SR-AE learning scheme with the aid of the SR transform. This scheme can lear n a transceiver model which shows a robust performance over the Poisson channel. Then, we propose another learning scheme called the DNN, which follows a different design philosophy from the AE scheme. This learning scheme is established based on a framework where the transmitter and the receiver are regarded as two separate networks. The training algorithm of the DNN scheme can realize joint transceiver optimization over any channel whose conditional PDF or PMF is differentiable w.r.t the channel input. These two schemes are proved effective by simulations and have their own features. Compared with the transceiver learned by DNN scheme, the SR-AE transceiver suffers performance loss in some cases where the SR approximation theory can not apply, but it has fewer network parameters and only computes a single loss function. The transceivers learned by DNN always achieve the best performance in comparison with the SR-AE transceivers and the considered hand-crafted transceivers, but it requires the computation of two loss functions to update the parameters of the transmitter and receiver neural networks. To the best of our knowledge, this is the first study focusing on the end-to-end learning over Poisson channel and provides a basis for further researches on other more complex techniques for OWC systems under Poisson regime.

Acknowledgment

The authors wish to thank the anonymous reviewers for their valuable suggestions.

References

[1] R. M. Gagliardi and S. Karp, *Optical communications*, 2nd ed. Hoboken, NJ, USA: Wiley, 1995.
[2] N. Farsad, H. B. Yilmaz, A. Eckford, C. Chae, and W. Guo, "A comprehensive survey of recent advancements in molecular communication," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1887–1919, 2016.
[3] I. Bar-David, "Communication under the Poisson regime," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 31–37, 1969.
[4] S. Verdú, "Poisson communication theory," in *Proc. International Technion Communication Day in honor of Israel Bar-David*, 25 Mar. 1999. [Online] Available: http://www.princeton.edu/~verdu/.

[5] B. Matuz, E. Paolini, F. Zabini, and G. Liva, "Non-binary LDPC code design for the Poisson PPM channel," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4600–4611, Nov. 2017.
[6] N. D. Chatzidiamantis, G. K. Karagiannidis, and M. Uysal, "Generalized maximum-likelihood sequence detection for photon-counting free space optical systems," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3381–3385, Dec. 2010.
[7] S. G. Wilson, M. Brandt-Pearce, Q. Cao, and J. H. Leveque, "Free-space optical MIMO transmission with Q-ary PPM," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1402–1412, Aug. 2005.
[8] M. L. B. Riediger, R. Schober, and L. Lampe, "Multiple-symbol detection for photon-counting MIMO free-space optical communications," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 5369–5379, Dec. 2008.
[9] Y. Li, M. Safari, R. Henderson, and H. Haas, "Optical OFDM with single-photon avalanche diode," *IEEE Photon. Technol. Lett.*, vol. 27, no. 9, pp. 943–946, May 2015.
[10] Z. Jiang, C. Gong, and Z. Xu, "Clipping noise and power allocation for OFDM-based optical wireless communication using photon detection," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 237–240, Feb. 2019.
[11] G.-C. Wang, C. Gong, and Z.-Y. Xu, "Signal characterization for multiple access non-line of sight scattering communication," *IEEE Trans. Commun.*, vol. 66, no. 9, pp. 4138–4154, Sept. 2018.
[12] C. Gong, Q. Gao, and Z.-Y. Xu, "Signal detection for superposition transmission protocols for optical wireless scattering broadcast channel," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5480–5493, Aug. 2018.
[13] J. Zhang, L. Si-Ma, B. Wang, J. Zhang, and Y. Zhang, "Low-complexity receivers and energy-efficient constellations for SPAD VLC systems," *IEEE Photon. Technol. Lett.*, vol. 28, no. 17, pp. 1799–1802, Sep. 2016.
[14] C. Abou-Rjeily, "Spatial-multiplexing for photon-counting MIMO-FSO communication systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5789–5803, Sept. 2018.
[15] C. Gong and Z.-Y. Xu, "Linear receivers for optical wireless scattering communication with multiple photon detectors," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2014, pp. 438–443.
[16] T. Oshea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognitive Commun. & Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
[17] N. Ye, X. Li, H. Yu, L. Zhao, W. Liu, and X. Hou, "DeepNOMA: A unified framework for NOMA using deep multi-task learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2208–2225, Apr. 2020.
[18] H. Ye, L. Liang, G. Y. Li, and B. Juang, "Deep learning-based end-to-end wireless communication systems with conditional GANs as unknown channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3133–3143, May 2020.
[19] F. A. Aoudia and J. Hoydis, "Model-free training of end-to-end communication systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2503–2516, Nov. 2019.
[20] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, 2018.
[21] M. Soltani, W. Fatnassi, A. Aboutaleb, Z. Rezki, A. Bhuyan, and P. Titus, "Autoencoder-based optical wireless communications systems," in *Proc. IEEE Globecom Workshops*, 2018, pp. 1–6.
[22] H. Lee, I. Lee, and S. H. Lee, "Deep learning based transceiver design for multi-colored VLC systems," *Opt. Exp.*, vol. 26, no. 5, pp. 6222–6238, 2018.
[23] B. Zhu, J. Wang, L. He, and J. Song, "Joint transceiver optimization for wireless communication PHY using neural network," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1364–1373, Jun. 2019.
[24] J. Song, B. Peng, C. Häger, H. Wymeersch, and A. Sahai, "Learning physical-layer communication with quantized feedback," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 645–653, 2020.
[25] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations (ICML)*, Banff, AB, Canada, Apr. 14–16, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114
[26] D. Sebastian, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132-143, Feb. 2018.
[27] B. Y. G. Ian and C. Aaron, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[28] J. H. Curtiss, "On transformations used in the analysis of variance," *Annal. Math. Statist.*, vol. 14, no. 2, pp. 107–122, Jun. 1943.
[29] Z. Zhu, J. Zhang, R. Chen, and H. Yu, "Autoencoder-based transceiver design for OWC systems in log-normal fading channel," *IEEE Photon. J.*, vol. 11, no. 5, Oct. 2019, Art. no. 7905912.
[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Jul. 2015, pp. 448–456.
[31] C. Anna, H. Mikael, and M. Michael, "The loss surfaces of multilayer networks," in *Proc. 18th Int. Conf. Artificial Intell. and Stat. (AISTATS)*, San Diego, CA, USA, Jun. 2017, pp. 192–204.