DTU Library

Gradient-free training of autoencoders for non-differentiable communication channels

Jovanovic, Ognjen; Yankov, Metodi Plamenov; Da Ros, Francesco; Zibar, Darko

Link back to DTU Orbit

Gradient-free training of autoencoders for non-differentiable communication channels

Ognjen Jovanovic, *Student Member, IEEE,* Metodi P. Yankov, *Member, IEEE,* Francesco Da Ros, *Senior Member, IEEE,* and Darko Zibar, *Member, IEEE*

Abstract—Training of autoencoders using the back-propagation algorithm is challenging for non-differential channel models or in an experimental environment where gradients cannot be computed. In this paper, we study a gradient–free training method based on the cubature Kalman filter. To numerically validate the method, the autoencoder is employed to perform geometric constellation shaping on differentiable communication channels, showing the same performance as the back-propagation algorithm. Further investigation is done on a non–differentiable communication channel that includes: laser phase noise, additive white Gaussian noise and blind phase search-based phase noise compensation. Our results indicate that the autoencoder can be successfully optimized using the proposed training method to achieve better robustness to residual phase noise with respect to standard constellation schemes such as Quadrature Amplitude Modulation and Iterative Polar Modulation for the considered conditions.

Index Terms—Optical fiber communication, cubature Kalman filter, end-to-end learning, geometric constellation shaping, phase noise.

I. INTRODUCTION

COMMUNICATION systems consist of a transmitter and receiver designed with the aim to reliably transfer information from one end to another over a physical channel, e.g. air or optical fiber. Typically, both the transmitter and receiver are structured as a chain of multiple independent signal processing blocks, such as channel coding, modulation, pulse shaping, and equalization. Even though this block-wise approach has proven to be efficient, it is uncertain if it achieves the best possible end-to-end performance, e.g. highest throughput.

End-to-end learning of a communication system, which optimizes the transmitter and receiver (including all of their processes) jointly for a specific channel model and performance metric was introduced in [1]. A communication system is perceived as an autoencoder (AE) [2] by representing the transmitter and receiver as neural networks (NNs) and considering the channel model as a non-trainable layer. Optimizing the AE is typically done by applying gradient–based algorithms, which require a *differentiable* channel model. In [3], the AE was trained numerically for a wireless link and its benefits demonstrated experimentally. The idea has been expanded and applied to orthogonal frequency-division multiplexing (OFDM) [4] and multiple–input multiple–output (MIMO) [5]. In order to optimize full waveforms for optical

O. Jovanovic, M. P. Yankov, F. Da Ros, and D. Zibar are with the Department of Photonic Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, e-mail: ognjo@fotonik.dtu.dk

fiber transmission, the AE has been applied to a dispersive linear fiber channel [6], [7], and nonlinear frequency division multiplexing (NFDM) [8], [9] for transmission over the nonlinear dispersive fiber channel model.

The AE approach can be utilized for geometric constellation shaping (GCS) in order to close the gap to the theoretically achievable information rate, which is experienced with uniformly distributed signalling (such as conventional quadrature amplitude modulation (QAM)). Geometric constellation shaping is a process of optimizing the positions of the constellation points on the I/Q plane. The main goal of GCS is to achieve the best possible trade–off between Euclidean distance and energy distribution of the constellation for the given channel. Embedding the AE with different optical fiber channel models to learn geometric constellation shapes was demonstrated for perturbative models of the nonlinear fiber channels [10], [11], for non-dispersive channels [12] or for linear links with up to 1.2 dB of optical signal-to-noise ratio (SNR) gain with respect to standard QAM [13].

All the aforementioned works fulfilled the requirement that the channel model is differentiable in order to perform classical gradient–based optimization of an AE. This requirement is often too strict because 1) not all channel models are differentiable; 2) approximating a channel with a simple differentiable model results in inaccuracies [14]; 3) an accurate differentiable channel model can be too complicated for reliable optimization.

As an alternative, it was proposed to use generative adversarial networks (GANs) to provide a simple differentiable channel model of the complex or non-differentiable observed system [15]. The GANs are used to train the AE as demonstrated in [16] for wireless communication and [14] for non-coherent fiber optic communication. However, using a GAN as a channel model adds an extra step to the AE optimization. For every optimization step of the AE, new data needs to be obtained from the non-differentiable channel model and used to train the GAN. The GAN requires a lot of training samples to learn an accurate channel approximation. Therefore, the overall process of AE optimization using GANs can be time-consuming [14]. Moreover, the data generated by the GAN during the AE optimization is synthetic and it is just an approximation of the probability distribution of the actual channel.

In [17], [18], a two–phase alternating algorithm for training of the AE without a channel model was presented. The algorithm is based on observing the transmitter and receiver as two separate NNs, applying a variant of reinforcement learning

to train the transmitter and supervised learning to train the receiver. This approach is in contrast with a joint optimization since the optimization of the transmitter and receiver is not done simultaneously, and it requires more samples to converge [19]. A simultaneous optimization approach for AE without channel knowledge using simultaneous perturbation stochastic approximation (SPSA) [20] for gradient estimation was demonstrated in [19]. However, it was shown that the variance of the gradient estimation increases with the number of parameters [18]. As a consequence, SPSA struggles to train AEs with a large number of parameters, e.g. a large constellation size if targeting constellation shaping.

In this paper, the differentiable channel model requirement is lifted by proposing a derivative-free optimization method for AEs. This allows the encoder and the decoder to be optimized simultaneously for *arbitrary black-box channels*, including non-numerical ones (e.g. experimental test-beds). Also, the method has the potential for online optimization. The proposed method is exemplified by adopting the cubature Kalman filter (CKF) [21] for the optimization of an AE. In order to show that method can be used for differentiable channel models, the AE performs GCS for a differentiable AWGN and nonlinear phase noise channels, resulting in nearly identical performance to typically used gradient–based optimization. Then, the AE is trained to perform GCS on a phase noise channel with residual phase noise, resulting from a non–differentiable carrier phase recovery algorithm.

The remainder of the paper is organized as follows. In Section II the basic principles of estimating the mutual information and the key concepts of using an AE for GCS are described. Section III explains how the weights of the AE can be optimized using CKF. A detailed description of the system, the AE architecture and channel models, is provided in Section IV. Section V provides results on the mutual information achieved by the AE for different channel models. In Section VI the simulation results and future work are discussed. The conclusions are summarized in Section VII.

Notations: Boldface denotes multivariate quantities such as vectors (lowercase) and matrices (uppercase). The sets of real and complex numbers are denoted as $\mathbb{R}$ and $\mathbb{C}$, respectively. The subscript k and j indicate time and iteration step, respectively. The covariance matrix of matrix $\mathbf{A}$ is denoted as $\mathbf{P_{AA}}$, whereas the cross–covariance matrix of matrices $\mathbf{A}$ and $\mathbf{B}$ is denoted as $\mathbf{P_{AB}}$. The i-th column of a matrix $\mathbf{A}$ is represented by $\mathbf{A}_i$, whereas the i-th scalar element of a vector $\mathbf{a}$ is represented by $\mathbf{a}^{(i)}$. The subscript $j|j-1$ is used to indicated that the current value of the matrix (vector) is conditional to the previous iteration. If two subscripts occur for the same matrix they are separated by a comma, e.g. $\mathbf{A}_{i,j|j-1}$ or $\mathbf{P}_{\mathbf{AA},j|j-1}$.

II. FUNDAMENTALS OF GEOMETRIC CONSTELLATION SHAPING

A. Mutual information

Consider X to be a sequence of complex constellation points (symbols) that take values from $\mathcal{X} = \{x_1, x_2, \ldots, x_M\}$ with a uniform probability mass function $P_X(x) = \frac{1}{M}$, where M is

the number of constellation points. The entropy of the constellation $H(X) = -\sum_{x \in \mathcal{X}} P_X(x) \log_2(P_X(x)) = \log_2(M) = m$ represents the number of bits carried by a symbol. Let Y be a continuous complex output of a memoryless channel with X as its input. The input-output relation of the channel is governed by the channel transition probability density $p_{Y|X}(y|x)$. The amount of information that Y contains about X in bits per symbol is represented by mutual information (MI)

$$
\begin{aligned}
I(X;Y) &= H(X) - H_p(X|Y) = m - H_p(X|Y) \\
&= \sum_{x \in \mathcal{X}} P_X(x) \int_{\mathbb{C}} p_{Y|X}(y|x) \log_2 \frac{p_{Y|X}(y|x)}{p_Y(y)} dy,
\end{aligned} \quad (1)
$$

where $\mathbb{C}$ denotes the set of complex numbers, $H_p(X|Y) = \mathbb{E}[p_{X|Y}(x|y)]$ is the conditional entropy of X given Y and $p_Y(y)$ is the probability density distribution of Y.

In order to calculate Eq. (1), the transition probability $p_{Y|X}(y|x)$ must be known. Since the main topic of this paper is black-box channels (which do not have known explicit analytical expressions), the transition probability is unknown. In such cases, Eq. (1) needs to be bound. A lower bound on the MI, also known as the achievable information rate (AIR), can be obtained by using the mismatched decoding approach and assume the transition probability $q_{Y|X}(y|x)$ of an auxiliary channel instead of the true $p_{Y|X}(y|x)$ [22],

$$
\begin{aligned}
I(X;Y) &\geq H(X) - \hat{H}_q(X|Y) = m - \hat{H}_q(X|Y) \\
&= \sum_{x \in \mathcal{X}} P_X(x) \int_{\mathbb{C}} p_{Y|X}(y|x) \log_2 \frac{q_{Y|X}(y|x)}{q_Y(y)} dy,
\end{aligned} \quad (2)
$$

where $\hat{H}_q(X|Y) = \mathbb{E}[q_{X|Y}(x|y)]$ is the upper bound of the true conditional entropy $H_p(X|Y)$. The inequality turns to equality only when $q_{Y|X}(y|x) = p_{Y|X}(y|x)$.

B. Geometric constellation shaping with autoencoders

Typically, GCS involves the optimization of the constellation points in the complex plane with the aim to maximize the MI $I(X;Y)$. The MI can be maximized without explicit channel knowledge or assumption by leveraging the AEs to perform GCS. The AE accomplishes this by making an approximation $\hat{p}_{X|Y}(x|y)$ of the true posterior distribution $p_{X|Y}(x|y)$ using the decoder NN. It was shown that an AE can be used to improve the lower bound of the MI [12].

The considered communication system employing an AE for geometrical constellation shaping is shown in Fig. 1. The encoder and the decoder are represented by feed-forward neural networks $NN_e(\mathbf{w}_e)$ and $NN_d(\mathbf{w}_d)$, parameterized with trainable weights $\mathbf{w}_e \in \mathbb{R}^{N_e}$ and $\mathbf{w}_d \in \mathbb{R}^{N_d}$, respectively. The number of weights, including all trainable weights and biases of the NN, in the encoder is denoted by N_e, whereas the number of weights in the decoder is N_d. The overall goal is to find the corresponding $NN_e(\mathbf{w}_e)$ and $NN_d(\mathbf{w}_d)$ topologies and the weight set, $\mathbf{w} = \{\mathbf{w}_e, \mathbf{w}_d\} \in \mathbb{R}^N$, where $N = N_e + N_d$ is the total number of trainable weights, that would maximize the MI between the transmitted and the received symbols for the considered channel. In this paper, the goal of the encoder is to shape the input sequence to a geometrically optimized constellation that is robust to channel impairments, whereas
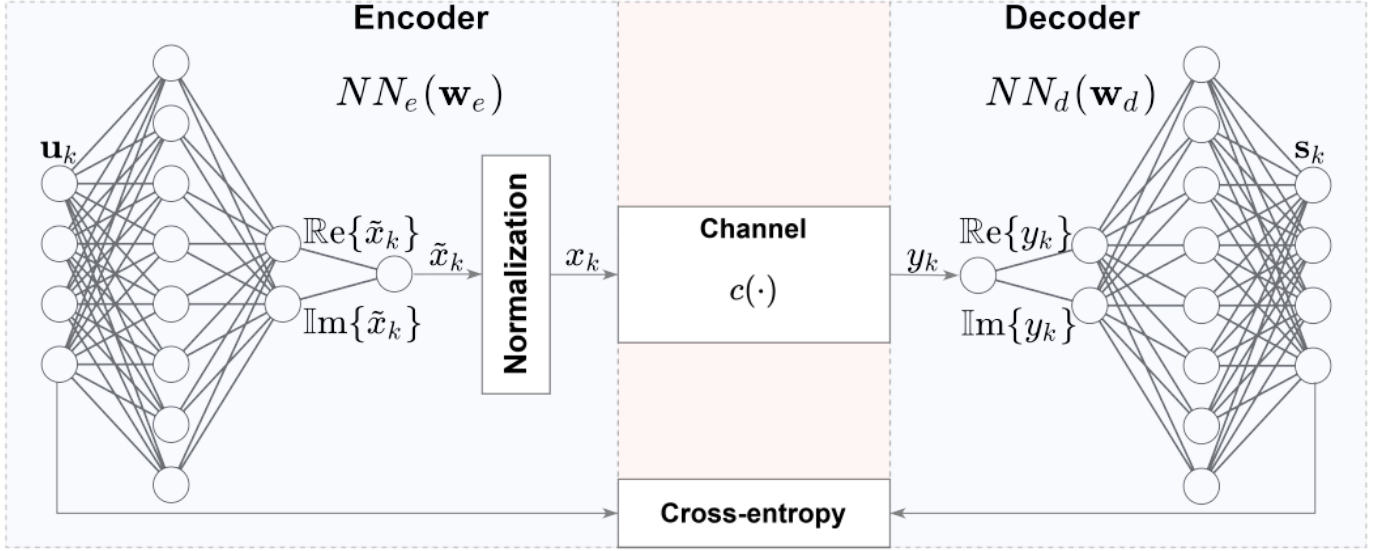
Fig. 1. Example of autoencoder model for geometrical constellation shaping. The number of input/output nodes of the AE and the hidden layers are used for illustration purposes.

the decoder learns to reconstruct the transmitted symbols with high fidelity.

The encoder $NN_e(\mathbf{w}_e)$ performs a mapping of the input one–hot encoded vector $\mathbf{u}_k \in \mathbb{R}^M$ to an output $\tilde{x}_k = \mathbb{Re}\{\tilde{x}_k\} + i \cdot \mathbb{Im}\{\tilde{x}_k\} \in \mathbb{C}$, representing a point in a constellation plane, where k is a running index representing time, $\mathbb{Re}\{\cdot\}$ is the real part and $\mathbb{Im}\{\cdot\}$ is the imaginary part of the complex point. Normalized to an average power of 1, the complex symbol x_k is then sent through the channel $c(\cdot)$. It should be emphasised that channel $c(\cdot)$ can be any channel for which an output can be generated with a given input. The output of the channel, denoted by y_k is then applied to the decoder $NN_d(\mathbf{w}_d)$ which uses a softmax output layer to output a vector of posterior probabilities $\mathbf{s}_k \in \mathbb{R}^M$. The full data propagation throughout the AE is represented with $\mathbf{h}(\cdot)$ such that $\mathbf{s}_k = \mathbf{h}(k, \mathbf{w}, \mathbf{u}_k)$.

The described AE structure can be considered an $M-$ class pattern-classification problem. For such a problem, it is appropriate to optimize the weights $\mathbf{w}$ by minimizing the cross–entropy cost function [2]

$$
\begin{aligned}
J_{CE}(k, \mathbf{w}) &= -\sum_{i=1}^{M} \mathbf{t}_k^{(i)} \log \mathbf{s}_k^{(i)} \\
&= -\sum_{i=1}^{M} \mathbf{t}_k^{(i)} \log \mathbf{h}^{(i)}(k, \mathbf{w}, \mathbf{u}_k),
\end{aligned}
\tag{3}
$$

where $\mathbf{t}_k$ is the target AE output and (i) denotes the i-th output node of the decoder neural network NN_d. Even though the target AE output is $\mathbf{t}_k = \mathbf{u}_k$, the notation $\mathbf{t}_k$ will be used in the rest of the paper for clarity. The cross–entropy cost function can be used to calculate an AE-based upper bound $\hat{H}_{\hat{p}}(X|Y) = \mathbb{E}[\hat{p}_{X|Y}(x|y)] = \sum_{k=1}^{K} J_{CE}(k, \mathbf{w})$ to the true conditional entropy $H_p(X|Y)$, where K is the number of symbols transmitted. Replacing $\hat{H}_q(X|Y)$ in Eq. 2 with $\hat{H}_{\hat{p}}(X|Y)$ an AE-based lower bound on the MI is obtained. Therefore, minimizing the cross–entropy cost function is equivalent to maximizing the MI between transmitted and received symbols,

therefore satisfying the overall goal of the system. The weights $\mathbf{w}$ can be optimized by applying Bayesian filtering techniques [23] as discussed in the following.

III. TRAINING OF AUTOENCODERS USING BAYESIAN FILTERING

In order to apply Bayesian filtering techniques for the optimization, the AE structure depicted in Fig. 1 has to be described using a state–space modelling framework. The state–space model is described by a pair of equations, known as the *process* equation and *measurement* equation. The process equation describes the evolution of the states which are non–observable variables that we would like to estimate ($\mathbf{w}$ for the considered case). The corresponding measurement equation relates observable variables ($\mathbf{s}_k$ for the considered case) to the states [23].

In order to correctly perform the optimization of the AE using Bayesian filtering techniques, the state–space model has to be defined specifically for the system under consideration. In the following subsection III-A, the space–state model will be constructed so that it fulfills the requirements of the AE structure and the desired performance metric (MI for the considered case). First, the process equation will be defined, followed by an adaptation of the measurement equation to fit the system requirements (cross–entropy cost function and batch optimization). In subsection III-B, the optimization procedure will be detailed based on the established state–space model.

A. Autoencoder state–space model

For the AE, the weights associated with the encoder and decoder NNs form column vectors constructed by stacking the weights associated with each neuron, starting with the first neuron in the first hidden layer and progressing first in width (next neuron) and then in depth (next layer). The two column

vectors are then stacked to form a single column weight vector $\mathbf{w}_j = [\mathbf{w}_{e,j}^T, \mathbf{w}_{d,j}^T]^T$, where j is the j-th iteration step. The AE weights are considered as states and their evolution is described by the first–order auto-regressive equation [23]:

$$\mathbf{w}_j = \mathbf{w}_{j-1} + \mathbf{q}_{j-1} \tag{4}$$

where $\mathbf{q}_j$ represents process–noise term which corresponds to white Gaussian noise of zero mean and covariance matrix $\mathbf{Q_{j-1}}$. For simplicity reasons, the covariance matrix is defined as a diagonal matrix $\mathbf{Q_{j-1}} = Q_{j-1}\mathbf{I}$, where $\mathbf{I}$ is an identity matrix. The process-noise term is intentionally included to avoid the states being trapped in a local minimum during the initial training stage.

The measurement equation for the state–space model describing the AE is defined such that it assumes that the target output $\mathbf{t}_j$ is a noisy measurement of the actual AE output:

$$\mathbf{t}_j = \mathbf{h}(j, \mathbf{w}_j, \mathbf{u}_j) + \mathbf{r}_j, \tag{5}$$

where $\mathbf{r}_j$ is the measurement–noise term which corresponds to white Gaussian noise of zero mean and covariance matrix $\mathbf{R}_j$. The term $\mathbf{u}_j$ is the input of the AE at iteration j. With the measurement equation in its current form, Bayesian filtering techniques implicitly minimize the sum of squared errors cost function [23],

$$J(j, \mathbf{w}_j) = \sum_{i=1}^{M} (\mathbf{t}_j^{(i)} - \mathbf{h}^{(i)}(j, \mathbf{w}_j, \mathbf{u}_j))^2. \tag{6}$$

However, such a cost function does not comply with the goal of maximizing MI between the transmitted and received symbol. Therefore, the measurement equation needs to be adapted to a form where the weight estimation $\mathbf{w}_j$ is obtained by minimizing cross-entropy cost function. The adaptation is done following the steps explained in [23]. First, the measurement equation (5) can be rewritten as

$$\mathbf{0} = (\mathbf{t}_j - \mathbf{h}(j, \mathbf{w}_j, \mathbf{u}_j)) + \mathbf{r}_j, \tag{7}$$

where the measurement is forced to take a vector-value $\mathbf{0}$. Then, the vector-valued measurement equation can be reformulated as a scalar-valued equation

$$0 = \sqrt{\sum_{i=1}^{M} (\mathbf{t}_j^{(i)} - \mathbf{h}^{(i)}(j, \mathbf{w}_j, \mathbf{u}_j))^2} + r_j = \sqrt{J(j, \mathbf{w}_j)} + r_j, \tag{8}$$

where r_j is now a single dimension zero mean Gaussian noise with a variance R_j.

The scalar reformulation of the measurement equation offers significant benefits: 1) Reduces the computational complexity of the Bayesian filter optimization; 2) improves the numerical stability of the Bayesian filter [23]; 3) The cost function is directly incorporated into the measurement equation. The new formulation of the measurement equation allows various cost functions to be fitted and used with Bayesian filtering techniques. The final step of the adaptation is just incorporating

the required cross–entropy cost function from Eq. (3) into Eq. (8)

$$\tilde{t}_j = 0 = \sqrt{-\sum_{i=1}^{M} \mathbf{t}_j^{(i)} \log \mathbf{h}^{(i)}(j, \mathbf{w}_j, \mathbf{u}_j)} + r_j \tag{9}$$
$$= \tilde{h}(j, \mathbf{w}_j, \mathbf{u}_j, \mathbf{t}_j) + r_j.$$

The Bayesian filtering techniques typically perform a single weight-vector update on the basis of a single input-output data pair. In the case of the AE, due to the normalization of the constellation, it would be advantageous to coordinate the weight update on a batch of data. Batch optimization is referred to as multistream training in Kalman filter related work and it is described in [24]. Multiple instances of the input, a batch of size B, are propagated through the system with the same weight set $\mathbf{w}_j$. The input to the AE is now $\mathbf{U}_j = [\mathbf{u}_{j \cdot B}, \mathbf{u}_{j \cdot B+1}, \dots, \mathbf{u}_{j \cdot B+(B-1)}]^T$, each are propagated through the AE and form a vector-valued measurement,

$$\tilde{\mathbf{t}}_j = \mathbf{0} = [\tilde{t}_{j \cdot B}, \tilde{t}_{j \cdot B+1}, \dots, \tilde{t}_{j \cdot B+(B-1)}]^T = \tag{10}$$
$$= \tilde{\mathbf{h}}(j, \mathbf{w}_j, \mathbf{U}_j, \mathbf{T}_j) + \mathbf{r}_j,$$

where $\mathbf{T}_j$ is the target output corresponding to the AE input $\mathbf{U}_j$ and $\mathbf{r}_j = [r_{j \cdot B}, r_{j \cdot B+1}, \dots, r_{j \cdot B+(B-1)}]^T$ is a vector that represents different independent noise realizations. A diagonal covariance matrix $\mathbf{R}_j = R_j \mathbf{I} \in \mathbb{R}^{B \times B}$ is used to describe $\mathbf{r}_j$.

Now that both process and measurement equations are defined to fulfill the requirements of the AE, a Bayesian filtering technique can be used to estimate the weights. Based on the Bayesian filtering paradigm, a complete statistical description of the state at iteration j is provided by the posterior density of the state. Exploiting the new measurement, the old posterior density of the state is updated in two steps, *prediction* and *correction*. In the correction step, the posterior density of the state is computed exploiting the predictive density obtained in the prediction step. The weight estimation within the state–space model given by Eq. (4) and (9) can be solved by using various nonlinear Bayesian state estimation techniques [23]. In this paper, the focus was put on using cubature Kalman filter (CKF) [21]. The advantage of using CKF is its accuracy and most importantly that it does not require computations of gradients providing a gradient–free training of AEs.

B. Cubature Kalman filter

The assumption of CKF is that the predictive and the posterior densities of the state are Gaussian distributions described by their means and covariances. In the prediction step, the predictive density of the current process state is computed based on the posterior density of the previous process state. The mean and covariance of the predictive density of the process state are denoted as $\hat{\mathbf{w}}_{j|j-1} \in \mathbb{R}^N$ and $\mathbf{P}_{j|j-1} \in \mathbb{R}^{N \times N}$, respectively. Since the process equation (4) yields a linear transition, $\hat{\mathbf{w}}_{j|j-1}$ and $\mathbf{P}_{j|j-1}$ are defined as:

$$\hat{\mathbf{w}}_{j|j-1} = \mathbf{w}_{j-1}, \tag{11}$$
$$\mathbf{P}_{j|j-1} = \mathbf{P}_{j-1} + \mathbf{Q}_{j-1}. \tag{12}$$

In the correction step, the posterior density of the state is obtained from the predicted measurement density. In order
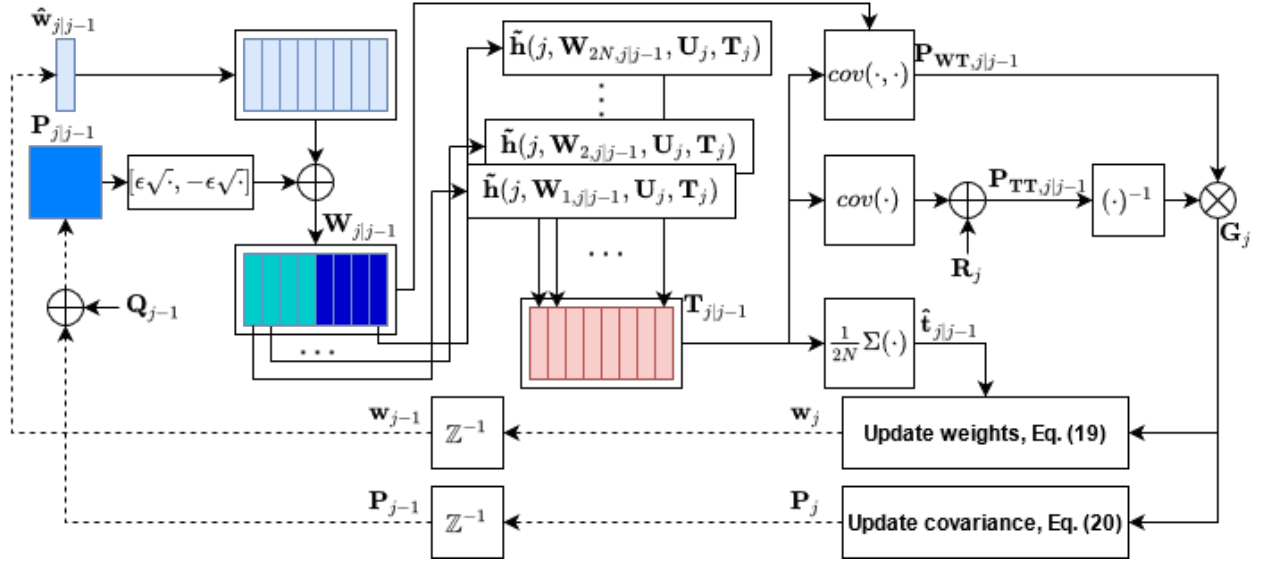
Fig. 2. Schematic of the optimization process. The covariance matrix calculation is denoted as $cov(\cdot)$, whereas $cov(\cdot, \cdot)$ denotes the cross-covariance matrix calculation. A bar is used to represent a column vector, a square to represent a square matrix and multiple bars encapsulated a rectangular matrix. Two colors have been used with $\mathbf{W}_{j|j-1}$ to indicated that one half of the matrix is a a result of an addition and the other the result of a subtraction. The unit time delay operator is denoted as $\mathbb{Z}^{-1}$.

to calculate the predicted measurement density, $2N$ cubature points are formed

$$\mathbf{W}_{j|j-1} = [\hat{\mathbf{w}}_{j|j-1} + \epsilon\sqrt{\mathbf{P}_{j|j-1}} \quad \hat{\mathbf{w}}_{j|j-1} - \epsilon\sqrt{\mathbf{P}_{j|j-1}}] \in \mathbb{R}^{N\times 2N},$$
(13)

where $\sqrt{(\cdot)}$ is the square root of a matrix satisfying the relation $\mathbf{P}_{j|j-1} = \sqrt{\mathbf{P}_{j|j-1}}\sqrt{\mathbf{P}_{j|j-1}}^T$, and the unit cubature point $\epsilon = \sqrt{N}$. Each column of the cubature points $\mathbf{W}_{i,j|j-1}$, where $i = 1, 2, \ldots, 2N$ represents the i-th column of $\mathbf{W}_{j|j-1}$, is a different realizations of the AE weights and the input is propagated through each of them to obtain the output of each realization $\mathbf{T}_{j|j-1} \in \mathbb{R}^{B\times 2N}$. The obtained outputs are used to calculate the predicted measurement $\hat{\mathbf{t}}_{j|j-1} \in \mathbb{R}^B$:

$$\mathbf{T}_{j|j-1} = \tilde{\mathbf{h}}(j, \mathbf{W}_{j|j-1}, \mathbf{U}_j, \mathbf{T}_j)$$
(14)

$$\hat{\mathbf{t}}_{j|j-1} = \frac{1}{2N}\sum_{i=1}^{2N}\mathbf{T}_{i,j|j-1}$$
(15)

where $\mathbf{T}_{i,j|j-1}$ represents the i-th column of $\mathbf{T}_{j|j-1}$. The covariance associated with the predicted measurement density $\mathbf{P}_{\mathbf{TT},j|j-1} \in \mathbb{R}^{B\times B}$, also known as the *innovation covariance*, is estimated

$$\mathbf{P}_{\mathbf{TT},j|j-1} = cov(\mathbf{T}_{j|j-1}) + \mathbf{R}_j$$
$$= \frac{1}{2N}\sum_{i=1}^{2N}(\mathbf{T}_{i,j|j-1} - \hat{\mathbf{t}}_{j|j-1})(\mathbf{T}_{i,j|j-1} - \hat{\mathbf{t}}_{j|j-1})^T + \mathbf{R}_j,$$
(16)

where $cov(\cdot)$ denotes the covariance matrix calculation. The cross–covariance $\mathbf{P}_{\mathbf{WT},j|j-1} \in \mathbb{R}^{N\times B}$ of the state and the measurement is calculated

$$\mathbf{P}_{\mathbf{WT},j|j-1} = cov(\mathbf{W}_{j|j-1}, \mathbf{T}_{j|j-1})$$
$$= \frac{1}{2N}\sum_{i=1}^{2N}(\mathbf{W}_{i,j|j-1} - \hat{\mathbf{w}}_{j|j-1})(\mathbf{T}_{i,j|j-1} - \hat{\mathbf{t}}_{j|j-1})^T,$$
(17)

where $cov(\cdot, \cdot)$ denotes the cross–covariance matrix calculation. The Kalman gain $\mathbf{G}_j \in \mathbb{R}^{N\times B}$ is then calculated

$$\mathbf{G}_j = \mathbf{P}_{\mathbf{WT},j|j-1}\mathbf{P}_{\mathbf{TT},j|j-1}^{-1},$$
(18)

and used to update the weights $\mathbf{w}_j$ conditional on the measurement

$$\mathbf{w}_j = \hat{\mathbf{w}}_{j|j-1} + \mathbf{G}_j(\tilde{\mathbf{t}}_j - \hat{\mathbf{t}}_{j|j-1})$$
$$= \hat{\mathbf{w}}_{j|j-1} + \mathbf{G}_j(-\hat{\mathbf{t}}_{j|j-1})$$
(19)

Based on Eq. (10), the target output is $\tilde{\mathbf{t}}_j = \mathbf{0}$, therefore it can be neglected in Eq. (19). The Kalman gain $\mathbf{G}_j$ is also used to update the covariance $\mathbf{P}_j$

$$\mathbf{P}_j = \mathbf{P}_{j|j-1} - \mathbf{G}_j\mathbf{P}_{\mathbf{TT},j|j-1}\mathbf{G}_j^T.$$
(20)

Observing Eq. (13–18), it can be noticed that the complexity of the algorithm depends on the number of weights N and batch size B. The noise covariance matrices $\mathbf{Q_{j-1}}$ and $\mathbf{R_j}$ are hyperparameters of the CKF algorithm and its performance depends on how they are chosen. Fig. 2 illustrates the optimization process described by Eq. 11-20.

IV. COMMUNICATION SYSTEM DESCRIPTION

A. AE model

In [25], it was demonstrated that state-of-the-art generalized mutual information (GMI) performance can be achieved with an encoder NN with no hidden layers and with biases set to zero. Since GMI is a good indicator of system performance [26], an encoder NN with no hidden layers and with biases set to zero has been used in this paper. As a result of a coarse optimization, the decoder neural network has a single hidden layer of $\frac{M}{2}$ nodes and *Leaky Relu* as the activation function. The AE architecture is summarized in Table I. The initial weight vector $\hat{\mathbf{w}}_0$ is initialized using *Glorot initialization* [27], whereas the initial covariance $\mathbf{P}_0$ is an identity matrix.

TABLE I
PARAMETERS OF THE ENCODER AND DECODER NEURAL NETWORK

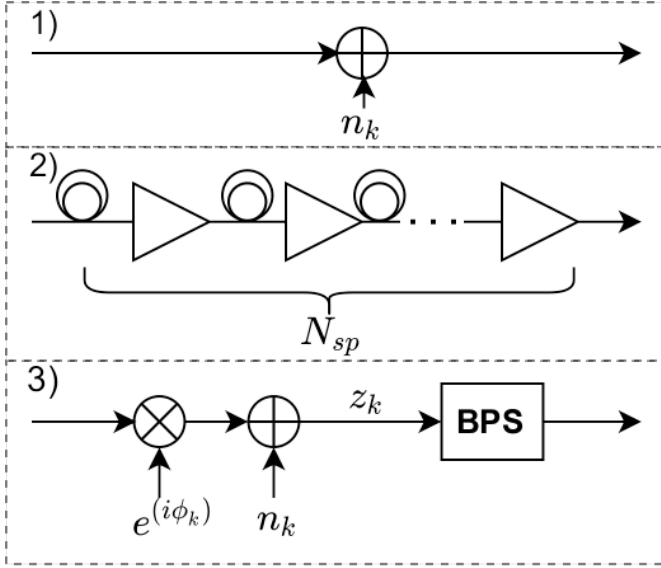|	Encoder NN	Decoder NN
# of input nodes	M	2
# of hidden layers	0	1
# of nodes per hidden layer	0	$M/2$
# of output nodes	2	M
Bias	No	Yes
Hidden layer activation function	None	Leaky Relu
Output layer activation function	Linear	Softmax

Fig. 3. Embedded channel models consist of: 1) additive white Gaussian noise (AWGN); 2) nonlinear phase noise (NLPN) 3) phase noise, AWGN and blind phase search algorithm for phase recovery.

B. Embedded channel models

In this paper, three channel models that operate on one sample per symbol basis, with symbol rate R_s, are embedded into the autoencoder. Those channel models are illustrated in Fig. 3:

1) AWGN channel: The noise variance is determined by the signal-to-noise ratio (SNR): $\sigma_N^2 = \frac{1}{SNR}$. As the AWGN channel model is differentiable, a fair comparison in the performance of the AE trained using the CKF and the gradient–based methods can be obtained. For the considered case, we employ the backpropagation algorithm using the Adam optimizer as a benchmark gradient–based training method [28].

2) Nonlinear phase noise channel: A memoryless fiber channel model that includes multiple spans N_{sp} with ideal lumped amplification is considered. Erbium-Doped Fiber Amplifiers (EDFAs), described by the noise figure NF and amplifier gain G, are used and they introduce *amplified spontaneous emission* (ASE) noise n_{ASE}. Similar to [12], the channel model is obtained from the nonlinear Schrödinger equation by neglecting the dispersion. The channel model shows the impact of a data-dependent nonlinear phase shift known as the *nonlinear phase noise* (NLPN) and it is defined by the recursion

$$z_k^{(i+1)} = z_k^{(i)} e^{i\gamma L_{eff}|z_k^{(i)}|^2} + n_{ASE,k}^{(i+1)}, \quad 0 \le i < N_{sp}, \quad (21)$$

where (i) is i-th fiber span, γ is the nonlinearity parameter, and $L_{eff} = (1 - e^{\alpha L})/\alpha$ is the effective length of a span. The actual length of a span is L and α is the attenuation coefficient. The noise term n_{ASE}^{i+1} is a zero–mean Gaussian distribution with variance $P_n = hF_cR_s(G \cdot NF - 1)/2$ [29], where h is Planck's constant and F_c is the carrier frequency. The input to the channel is a power rescaled signal $z_k^{(0)} = \sqrt{P_{in}}x_k$, where P_{in} the launch power. The output of the channel $y_k = z_k^{(N_{sp})}/P_{in}$ is normalized before processed by the decoder.

The NLPN channel model serves as the second differentiable channel model to use for comparison of the CKF and gradient–based methods. As for the previous channel model, the Adam optimizer is used as a benchmark gradient–based method.

3) Non–differentiable phase noise channel: As a validation of our proposed optimization method, we consider a channel model for which the gradients cannot be computed. For this purpose, the phase noise channel with blind phase search (BPS) [30], which is a standard phase noise compensation algorithm, is considered.

The encoder NN output x_k is distorted by phase noise and additive noise

$$z_k = x_k e^{i\phi_k} + n_k, \quad (22)$$

where the AWGN term n_k is characterized by noise power σ_N^2. The phase noise ϕ_k is modeled as a Wiener process

$$\phi_k = \phi_{k-1} + \Delta\phi_k, \quad (23)$$

where $\Delta\phi_k$ is the random phase increment with a zero-mean and variance $\sigma_\phi^2 = 2\pi\Delta\nu T_s$. The combined transmitter and receiver oscillator linewidth is denoted by $\Delta\nu$, and T_s is the symbol period.

The BPS is used as the phase recovery algorithm and it is the non-differentiable part of this channel model due to its hard-decision directed nature [30]. The BPS is a pure feedforward phase recovery algorithm based on rotating the received symbol by N_s test phases defined by:

$$\theta_i = \frac{i}{N_s} \cdot 2\pi, \quad i \in \{0, 1, \ldots, N_s - 1\}, \quad (24)$$

where i represents the i-th test phase. Decisions are made on each of the rotated symbols and the distance between the decided symbol $\hat{z}_{k,i}$ and the rotated symbol $z_{k,i}$ is calculated. Afterwards, distances of symbols rotated by the same test phase are summed over a window of size W_{BPS}

$$d_{k,i} = \sum_{j=1}^{W_{BPS}} |z_{k,i}^{(j)} - \hat{z}_{k,i}^{(j)}|^2, \quad (25)$$

where (j) is the j-th sample in the window. This step mitigates the effect of the AWGN on the quality of the decision. Finally, the optimal test phase is chosen by the minimum sum of distances [30]

$$\hat{\phi}_k = \underset{\theta_i}{\operatorname{argmin}} \quad d_{k,i}, \quad (26)$$

where argmin is a non–differentiable operation. The received symbol is rotated by the chosen test phase to output the phase compensated sample

$$y_k = z_k e^{-i\hat{\phi}_k}. \tag{27}$$

The performance of the BPS algorithm is determined by the parameters N_s and W_{BPS}.

C. Gaussian receiver

For optical communication, it is a common approach to use a mismatched Gaussian receiver [31], [32] and assume the transition probability $q_{Y|X}(y|x)$ in Eq. (2) is of an auxiliary Gaussian channel

$$q_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi\sigma_G^2}} \exp\left(-\frac{||y-x||^2}{2\sigma_G^2}\right), \tag{28}$$

where σ_G^2 is the noise variance and $||y - x||^2$ is a squared Euclidean distance between the symbols. Applying the Bayes' theorem, the conditional probability that a specific x was sent observing y is given as

$$q_{X|Y}(x|y) = \frac{p_X(x)q_{Y|X}(y|x)}{\sum_{x^i \in \mathcal{X}} p_X(x=x^i)q_{Y|X}(y|x=x^i)}. \tag{29}$$

The Monte Carlo approach can be used to evaluate Eq. (29). The noise variance σ_G^2 can be estimated from the channel input-output pairs.

In the case of the AWGN channel model, the assumed transition probability $q_{Y|X}(y|x)$ is identical to the true $p_{Y|X}(y|x)$. Therefore, the Gaussian receiver is the maximum likelihood (ML) receiver for the AWGN channel and the estimated noise variance $\sigma_G^2 = \sigma_N^2$. Therefore, on the AWGN channel, the performance of the Gaussian receiver is an upper bound for the performance of the NN decoder.

However, in the case of channel models 2) and 3), the true transition probability $p_{Y|X}(y|x)$ is approximated by $q_{Y|X}(y|x)$. The combined distortion of the nonlinear phase noise and the additive noise of channel model 2) is assumed to be purely Gaussian and approximated with the noise variance σ_G^2. The same assumption and approximation is made for the combined distortion of the residual phase noise and the additive noise of channel model 3). Therefore, the Gaussian receiver is a mismatched receiver for both of these channel model.

V. NUMERICAL RESULTS

The system symbol rate is $R_s = 32$ GBd and the size of the constellation is $M = 64$. At each training iteration j, a batch $\mathbf{U}_j$ of $B = 32 \cdot M$ one-hot encoded input vectors $\mathbf{u}_k$ is generated. The hyperparameters of the CKF algorithm Q_{j-1} and R_j are coarsely optimized using a standard grid search method. Both hyperparameters are sampled from the set $\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$ and the CKF algorithm is applied with each combination. The combination that achieves the best mutual information is chosen. The training was done for each SNR value. The AE is trained until the cost function converges. Afterwards, the weights of the AE

are fixed and testing is performed. The testing was done by running 100 simulations with 10^5 symbols per simulation. Each of the trained AE was tested with the same channel parameters as it was trained.

The constellations learned by AE will be compared to iterative polar modulation (IPM) [33] and square QAM (referred to simply as QAM in the following). The IPM was chosen as a benchmark because it is a near-optimal constellation shape for the AWGN channel [33].

A. AWGN channel

The studied SNR region includes values from the interval SNR = $\{10, 11, \ldots, 25\}$ dB. Since the Gaussian receiver is optimal for the AWGN channel, the presented results for this channel model only include the Gaussian receiver.

The simulation results of the testing for the AWGN channel are shown in Fig. 4, which illustrates the performance in MI with respect to the SNR. The learned constellations using the AE trained with the CKF and the backpropagation are denoted as AE-CKF and AE-BP, respectively. This notation will be used throughout the rest of this section. The learned constellations AE-CKF and AE-BP result in a similar performance in the MI with an average difference of around 0.01 bits/symbol. Compared to QAM, the two constellations achieve higher MI in the low SNR region, as expected, whereas the difference is marginal compared to the IPM.

The insets illustrate the learned constellations AE-CKF and AE-BP when training on SNR = 18 dB. The Euclidian distance between some of the points in AE-CKF is quite small, but this does not have a negative effect on the MI performance. Maximum shaping gain is not achieved by maximizing Euclidean distance but rather by optimizing the trade-off between the Euclidian distance and the energy distribution of the signal. As expected for an AWGN channel, the optimizer pushes points towards the origin in order to resemble the AWGN-optimal Gaussian distribution of the signal energy, thus providing overall gain.

B. NLPN channel

The training of the AE is performed by sweeping the launch power P_{in} in the interval $P_{in} = \{-8, -7.5, \ldots, 0\}$ dBm for the fiber parameters $\gamma = 1.27\frac{1}{W \cdot km}$, $\alpha = 0.2\frac{dB}{km}$, $NF = 5$ dB, $F_c = 193.41$ THz, $N_{sp} = 10$, and $L = 100$ km. The MI estimation for the QAM and IPM constellation is performed using the mismatched Gaussian receiver. The results for the constellations learn by applying CKF (AE-CKF) and backpropagation (AE-BP) include the MI estimation for both mismatched Gaussian receiver and decoder NN.

In Fig. 5, the MI performance with respect to launch power P_{in} is shown. For AE-CKF and AE-BP, the dashed lines represent the MI performance when the mismatched Gaussian receiver is used and the solid lines represent the MI performance when using the decoder NN. The learned constellations AE-CKF and AE-BP outperform both QAM and IPM in terms of MI, with both mismatched Gaussian receiver and decoder NN. Observing just the mismatched Gaussian receiver results, the learned constellations achieve a greater
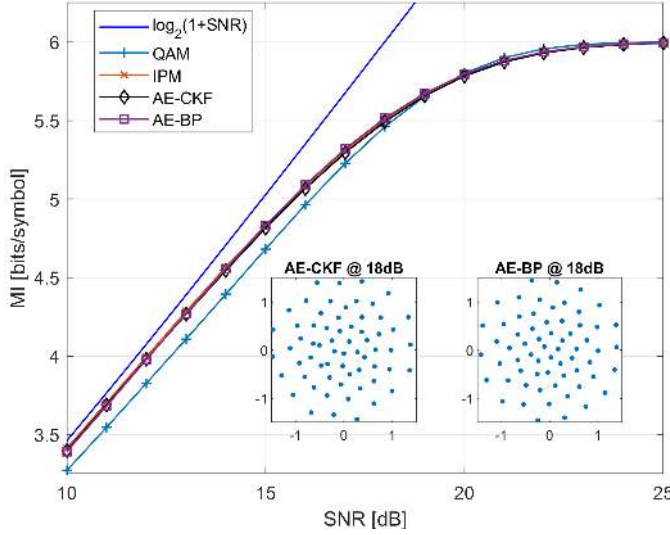
Fig. 4. *AWGN channel:* Mutual information as a function of SNR for constellation size $M = 64$. Inset: example of the learned constellations with CKF and BP at SNR = 18 dB.

Fig. 5. *NLPN channel:* Mutual information as a function of launch power P_{in} for constellation size $M = 64$. Inset: example of the learned constellations with CKF and BP for launch power $P_{in} = -2.5$ dBm.

maximum MI by up to 0.12 and 0.17 bits/symbol compared to QAM and IPM, respectively. Observing the studied launch power P_{in} region, the maximum gain in MI compared to QAM was obtained at $P_{in} = 0$ dBm and it amounts to around 0.26 bits/symbol. At the same launch power, the maximum gain in MI compared to IPM was achieved, reaching a value of up to 0.52 bits/symbol. It can be concluded that the learned constellations are more robust to nonlinear phase noise than QAM and IPM.

The two learned constellations AE-CKF and AE-BP have similar performance for both receivers in question. The average difference for both the mismatched Gaussian receiver and the decoder NN is around 0.01 bits/symbol. Observing the MI performance obtained with the decoder NN, similar mitigation and compensation was observed as what was shown in [12]. The insets illustrate the learned constellations AE-CKF and AE-BP when training on launch power $P_{in} = -2.5$ dBm.

The obtained results demonstrate that the proposed AE optimization method, CKF, can achieve similar performance to BP when the derivative of the cost function with respect to the encoder weights $\mathbf{w_e}$ can be calculated.

C. Non–differentiable phase noise channel

The training of AE is performed for each of the SNR values in the interval SNR = $\{15, 16, \ldots, 20\}$ dB combined with the BPS parameters $N_s = 36$ and $W_{BPS} = \{40, 64\}$ and linewidth $\Delta\nu = 100$ kHz. A mismatched Gaussian receiver is used instead of the decoder NN for the MI estimation during the testing phase in order to have a fair comparison between all the constellations.

In Fig. 6(a)-(b), the MI performance with respect to SNR is shown. The dashed lines show the performance of the constellations for the AWGN channel. The dashed line AE-CKF$_{AWGN}$ presents the MI obtained when the AE-CKF trained for channel model 3) is tested on channel model 1) for the respected SNR. These results are added in order to observe
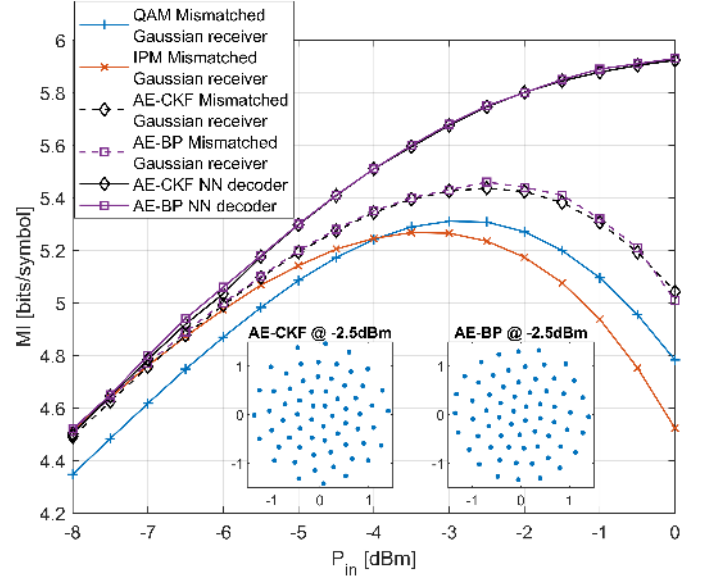
the penalty introduced by the phase noise. The solid lines represent the average MI value over the 100 test simulations at a given SNR, whereas the upper limit of the error bar is the maximum obtained MI value and the lower limit shows the 25th percentile. Therefore, the error bars represent the 75% of simulations with the highest MI. Constellations optimized on channel models 1) and 2) have been tested on channel model 3) and the curves representing the obtained results are denoted as "AE-CKF 1)" and "AE-CKF 2)", respectively. The insets illustrate the learned constellations AE-CKF when training on SNR= 18 dB.

Fig. 6(a) shows MI as a function of SNR when the BPS parameters are $N_s = 36$ and $W_{BPS} = 64$. For the SNR values with a mean MI less than 3.5 bits/symbol, all points are removed from the figure for visual clarity. For the used BPS parameters, phase slips occur at SNR = 15 dB for QAM constellation resulting in a wide error bar. The IPM constellation is designed for the AWGN channel and it is not suitable for a phase noise channel. The BPS algorithm does not estimate the phase accurately resulting into wide error bars. Observing the studied SNR region, the AE learned constellation achieves the highest gain in MI compared to QAM at SNR = 15 dB, and it amounts to around 0.45 bits/symbol. The gain in MI compared to QAM is 0.15 bits/symbol at SNR = 16 dB and with the increase of SNR the gain is slowly decaying to 0.08 bits/symbol, which is achieved at SNR = 20 dB. Observing the width of the error bars, it can be noticed that the spread of MI performance is smaller for the AE learned constellation compared to QAM, especially for lower SNR values. It can also be noticed that the maximum achieved MI value of QAM is lower than the 25th percentile of the AE learned constellation, implying that the AE learned constellation has a better performance at least in 75% of the test cases. The AE-CKF and IPM constellations achieve similar maximum MI at high SNR but the AE-
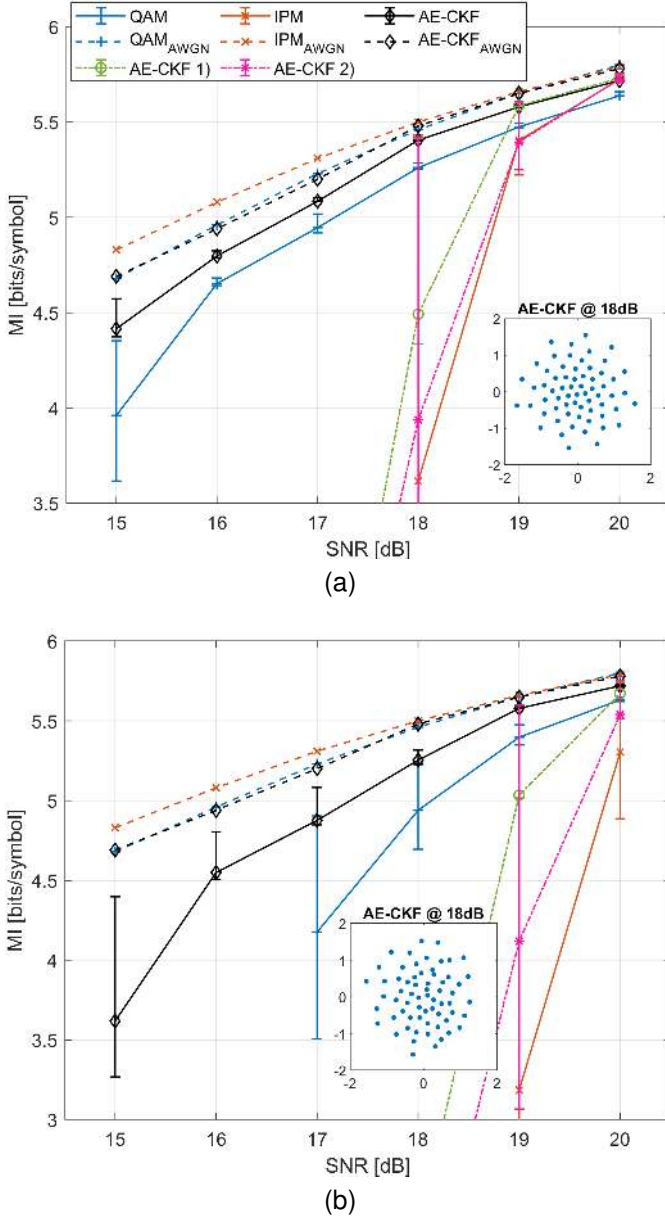
Fig. 6. *Nondifferentiable phase noise channel:* Mutual information as a function of SNR for constellation size M=64. The parameters of the BPS are: (a) $N_s = 36$ and $W_{BPS} = 64$ and (b) $N_s = 36$ and $W_{BPS} = 40$. The line shows the mean value, whereas the upper limit of the error bar is the max value and the lower limit is the 25th percentile. Points that have a mean less than 3.5 and 3 bits/symbol are omitted from the figure (a) and (b) for visual clarity, respectively. Inset: example of the learned constellation at SNR = 18 dB.

CKF is more robust and has a significantly higher mean MI value, whereas at SNR = 20 dB the two constellations have marginal differences in MI. The AE-CKF 1) achieves similar performance as AE-CKF for SNR = {19, 20} dB due to the fact that the main impairment at these SNR values is AWGN. Even though, both IPM and AE-CKF 1) are optimized for AWGN channel, the AE-CKF 1) performs better than IPM when applied to the non-differentiable channel. However, the AE-CKF 2) has similar performance as IPM. As expected both AE-CKF 1) and 2) do not perform well for the given non-differentiable channel since they were not optimized for it.

Next, the potential improvement at lower BPS complexity is analyzed. The window size was decreased to $W_{BPS} = 40$ and the results are illustrated in Fig. 6(b). All of the SNR points with a mean MI less than 3 bits/symbol were removed from the figure for visual clarity. Reducing the complexity of the BPS affects its performance, resulting in less accurate phase estimations. Therefore, the mentioned issues regarding QAM and IPM occur even for higher SNR values. The gain in MI achieved by the learned constellation compared to QAM at SNR = 20 dB is similar to the one achieved in the previous scenario, whereas compared to IPM for the same SNR = 20 dB greater gain was achieved, reaching a value of around 0.41 bits/symbol. The achieved gain is increasing with the lowering of the SNR. The gain at SNR = 17 dB already exceeds the highest gain achieved in the previous scenario, reaching a gain of around 0.7 bits/symbol. Similarly to the previous scenario, the error bars of the AE-CKF constellation are less spread than for QAM and IPM at the same SNR, indicating better robustness of the constellation. The less accurate phase estimation affects the performance of AE-CKF 1) and AE-CKF 2) which are in this scenario more penalized than QAM, but still less than IPM. At SNR = 20 dB, the AE-CKF 1) stands out with a higher MI than QAM, but still lower than AE-CKF.

The dashed lines in Fig. 6 show that the AE-CKF constellation learned by training on channel model 3) lost AWGN shaping gain compared to when it was trained directly on the AWGN channel. The AE-CKF$_{AWGN}$ and QAM$_{AWGN}$ now have similar performance in MI. Comparing situations with and without phase noise, the penalty in MI is significantly lower for AE-CKF than for QAM. This demonstrates that the AE-CKF constellation is more robust to phase noise at an expense of slight loss of AWGN shaping gain.

The previous figures show results when the AE was trained and tested on the same SNR value to show the best achieved MI performance at the given SNR. However, in practice an SNR estimation error of up to ~ 1 dB can be encountered [34]. Fig. 7 shows what the MI performance of constellations is when they are tested on different SNR values than what they were trained on. The results are only for the case when the BPS parameters are $N_s = 36$ and $W_{BPS} = 64$ and the curve denoted as "Envelope" is the same as "AE-CKF" curve from Fig. 6(a) and it indicates the training points as well. It has been expanded with the MI obtained when training at SNR = 14 dB, in order to show the robustness of the constellation trained at SNR = 15 dB. The results obtained when training on SNR = {18, 19, 20} dB are similar and therefore only the results obtained for SNR = 18 dB are shown. All of the constellations show at least up to ±0.5 dB of robustness to SNR estimation errors with less than 1% penalty in MI. The constellation trained at SNR = 18 dB can be used for the SNR = [17, 20] dB interval with less than 1% penalty in MI but at an expense of a considerable performance deterioration for SNR < 17 dB.

It has been established that AE-CKF constellations outperform QAM and IPM when using the mismatched Gaussian receiver. Now, a comparison between the mismatched Gaussian receiver and NN decoder for AE-CKF constellations
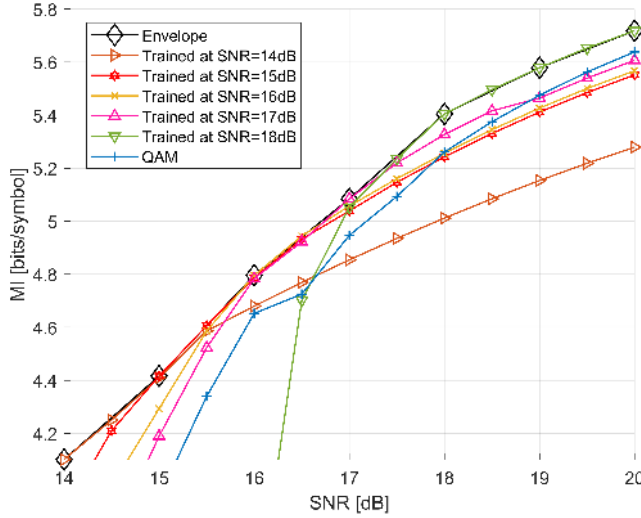
Fig. 7. *Nondifferentiable phase noise channel:* Mutual information as a function of SNR for constellation size $M = 64$ and BPS parameters $N_s = 36$ and $W_{BPS} = 64$. Sweeping the observed SNR interval with constellations trained with different SNR values.

Fig. 8. *Nondifferentiable phase noise channel:* Mutual information as a function of SNR for constellation size $M = 64$. The dashed lines represent the mismatched Gaussian receiver, whereas the solid lines represent the NN receiver.

is provided, since the encoder and decoder come as a pair. In Fig. 8 the MI is illustrated for the mismatched Gaussian receiver with dashed lines and NN decoder with solid lines for $W_{BPS} = 64$ and $W_{BPS} = 40$. For lower SNR, when the estimation of the phase noise is worse, the NN decoder outperforms the Gaussian receiver. As the SNR increases the estimation of the phase noise improves, resulting in AWGN being the main source of distortion. Thus, the gain in MI that NN decoders achieve compared to the mismatched Gaussian receiver decreases since the mismatched Gaussian receiver is optimal for the AWGN channel. For SNR= $\{19, 20\}$ dB the estimation of the phase noise is highly accurate for all the observed cases, therefore all have similar performance in regards to MI. In the case of $W_{BPS} = 64$, the estimation of the phase noise is still quite accurate at SNR= 15 dB and the NN decoder achieves around 0.09 bits/symbol gain compared to the Gaussian receiver. Whereas, in the case of $W_{BPS} = 40$, the estimation of the phase noise is significantly degraded at SNR= 15 dB and the NN decoder achieves up to around 0.44 bits/symbol gain compared to the Gaussian receiver.

VI. Discussion

In this paper, the differentiable channel model requirement is lifted by proposing a derivative-free optimization method for AEs. The proposed method is exemplified by adopting CKF for the AE weights optimization. Most notably, the results of this study imply that the AE can be optimized with arbitrary black–box channels and not only with known differentiable ones. Although the training capabilities of the proposed method were demonstrated in few scenarios, the proposed method should allow the training of autoencoders on arbitrary channel models, e.g. experimental test–beds, which is the ultimate goal. This is left for future work.

The CKF algorithm can be applied for training of recurrent NNs, which is important if considering a channel with finite memory, such as the dispersive optical channel. The proposed
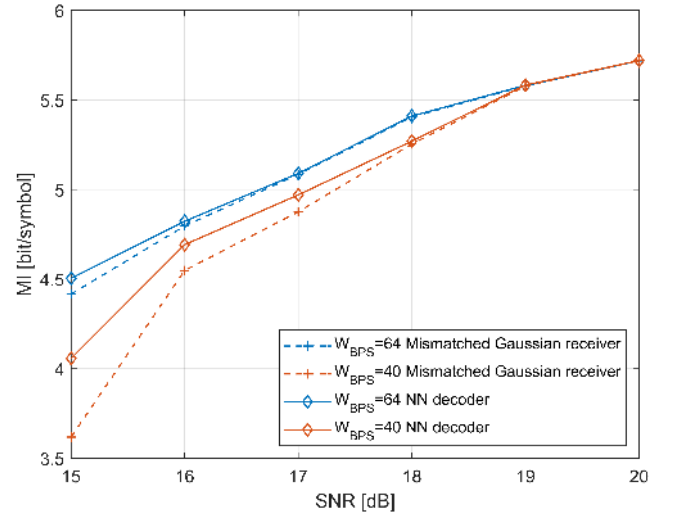
state spaced framework supports training of recurrent NNs [23], which showed superior performance compared to feed-forward NNs when applied to a channel with finite memory [7]. In such a case, the weight set is just expanded with the weights of the recurrent connections, meaning an identical state-based model for the kernel weights is used. The CKF derivation from this paper is therefore valid. This study is out of the scope of this paper, but is an interesting direction for future work.

VII. Conclusion

We have proposed and numerically demonstrated a derivative-free method for training autoencoders for geometrical constellation shaping. This is achieved by expressing the autoencoder weights and system as state-space models and then applying cubature Kalman filter (CKF) for state (encoder and decoder weights) estimation. For differentiable AWGN and NLPN channel models, it was shown that the performance, in terms of the mutual information of the learned constellations, is almost identical to when the training is performed using the standard backpropagation algorithm. The CKF trained autoencoder was also tested for a phase noise channel with a non-differentiable phase recovery algorithm. In such a case, the autoencoder-learned constellations achieved significant performance and robustness improvement with respect to conventional constellation shapes optimized for an AWGN channel. It should be emphasized that the proposed method can be applied to any AE structure and not just for GCS as it was used in this paper.

References

[1] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[2] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[3] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, "Deep Learning Based Communication Over the Air," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132–143, 2018.

[4] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. Ten Brink, "OFDM-Autoencoder for End-to-End Learning of Communications Systems," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[5] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017, pp. 76–80.

[6] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bulow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-End Deep Learning of Optical Fiber Communications," *Journal of Lightwave Technology*, vol. 36, no. 20, pp. 4843–4855, 2018.

[7] B. Karanov, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end optimized transmission over dispersive intensity-modulated channels using bidirectional recurrent neural networks," *Optics Express*, vol. 27, no. 14, p. 19650, 2019.

[8] S. Gaiarin, R. T. Jones, F. Da Ros, and D. Zibar, "End-to-end optimized nonlinear Fourier transform-based coherent communications," p. SF2L.4, 2020.

[9] S. Gaiarin, F. Da Ros, R. T. Jones, and D. Zibar, "End-to-End Optimization of Coherent Optical Communications Over the Split-Step Fourier Method Guided by the Nonlinear Fourier Transform Theory," *Journal of Lightwave Technology*, vol. 39, no. 2, pp. 418–428, 2021.

[10] R. T. Jones, T. A. Eriksson, M. P. Yankov, and D. Zibar, "Deep Learning of Geometric Constellation Shaping Including Fiber Nonlinearities," *European Conference on Optical Communication, ECOC*, 2018.

[11] R. T. Jones, M. P. Yankov, and D. Zibar, "End-to-end learning for GMI optimized geometric constellation shape," in *European Conference on Optical Communication, ECOC*, 2019.

[12] S. Li, C. Häger, N. Garcia, and H. Wymeersch, "Achievable Information Rates for Nonlinear Fiber Communication via End-to-end Autoencoder Learning," *European Conference on Optical Communication, ECOC*, 2018.

[13] M. Schaedler, S. Calabrò, F. Pittalà, G. Böcherer, M. Kuschnerov, C. Bluemm, and S. Pachnicke, "Neural network assisted geometric shaping for 800Gbit/s and 1Tbit/s optical transmission," *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, vol. Part F174-, no. DM, pp. 3–5, 2020.

[14] 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," *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, 2020.

[15] T. J. O'Shea, T. Roy, and N. West, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," in *2019 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2019, pp. 681–686.

[16] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–5.

[17] F. A. Aoudia and J. Hoydis, "End-to-End Learning of Communications Systems Without a Channel Model," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 298–303.

[18] ——, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.

[19] V. Raj and S. Kalyani, "Backpropagating Through the Air: Deep Learning at Physical Layer Without Channel Models," *IEEE Communications Letters*, vol. 22, no. 11, pp. 2278–2281, 2018.

[20] J. C. Spall *et al.*, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE transactions on automatic control*, vol. 37, no. 3, pp. 332–341, 1992.

[21] S. Haykin and I. Arasaratnam, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, 2009.

[22] D. M. Arnold, H. . Loeliger, P. O. Vontobel, A. Kavcic, and W. Zeng, "Simulation-Based Computation of Information Rates for Channels With Memory," *IEEE Transactions on Information Theory*, vol. 52, no. 8, pp. 3498–3508, 2006.

[23] I. Arasaratnam and S. Haykin, "Nonlinear Bayesian filters for training recurrent neural networks," in *Mexican International Conference on Artificial Intelligence*. Springer, 2008, pp. 12–33.

[24] L. A. Feldkamp and G. V. Puskorius, "A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering, and classification," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2259–2277, 1998.

[25] K. Gümüs, A. Alvarado, B. Chen, C. Häger, and E. Agrell, "End-to-end learning of geometrical shaping maximizing generalized mutual information," *Optical Fiber Communication Conference (OFC) 2020*, pp. 10–12, 2020.

[26] A. Alvarado, E. Agrell, D. Lavery, R. Maher, and P. Bayvel, "Replacing the Soft-Decision FEC Limit Paradigm in the Design of Optical Communication Systems," *J. Lightwave Technol.*, vol. 33, no. 20, pp. 4338–4352, Oct 2015.

[27] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research*, vol. 9, pp. 249–256, 2010.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] R. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity Limits of Optical Fiber Networks," *Journal of Lightwave Technology*, vol. 28, no. 4, pp. 662–701, 2010.

[30] T. Pfau, S. Hoffmann, and R. Noé, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M-QAM constellations," *Journal of Lightwave Technology*, vol. 27, no. 8, pp. 989–999, 2009.

[31] A. Lapidoth and S. S. Shitz, "On Information Rates for Mismatched Decoders," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1953–1967, 1994.

[32] M. P. Yankov, F. Da Ros, E. P. da Silva, S. Forchhammer, K. J. Larsen, L. K. Oxenløwe, M. Galili, and D. Zibar, "Constellation Shaping for WDM Systems Using 256QAM/1024QAM With Probabilistic Optimization," *Journal of Lightwave Technology*, vol. 34, no. 22, pp. 5146–5156, 2016.

[33] I. B. Djordjevic, H. G. Batshon, L. Xu, and T. Wang, "Coded polarization-multiplexed iterative polar modulation (PM-IPM) for beyond 400 Gb/s serial optical transmission," *Optical Fiber Communication Conference*, p. OMK2, 2010.

[34] F. N. Khan, Z. Dong, C. Lu, A. P. T. Lau, X. Zhou, and C. Xie, "Optical performance monitoring for fiber-optic communication networks," in *Enabling Technologies for High Spectral-Efficiency Coherent Optical Communication Networks*. Wiley Online Library, 2016.