

Concept and Experimental Demonstration of Optical IM/DD End-to-End System Optimization using a Generative Model

Boris Karanov^{1,2}, Mathieu Chagnon², Vahid Aref², Domanić Lavery¹, Polina Bayvel¹,
and Laurent Schmalen³

¹Optical Networks Group, Dept. of Electronic & Electrical Engineering, UCL, WC1E 7JE London, U.K.

²Nokia Bell Labs, 70435 Stuttgart, Germany

³Communications Engineering Lab, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

boris.karanov.16@ucl.ac.uk

Abstract: We perform an experimental end-to-end transceiver optimization via deep learning using a generative adversarial network to approximate the test-bed channel. Previously, optimization was only possible through a prior assumption of an explicit simplified channel model. © 2020 The Authors

OCIS codes: (060.4510) Optical communications, (060.2330) Fiber optics communications, (060.4080) Modulation

1. Introduction

The optimization of a complete communication system via deep learning has attracted great interest since the introduction of the idea in [1]. The application of such end-to-end neural network-based *autoencoders* is of particular importance in communication scenarios where the optimum transmitter-receiver pair is not known, or is computationally prohibitive to implement. An example is optical fibre communication based on intensity modulation and direct detection (IM/DD), where the joint effects of chromatic dispersion, introducing intersymbol interference (ISI), and square-law detection render the channel nonlinear with memory. The absence of optimal, computationally feasible, processing algorithms for IM/DD systems prompted the introduction of end-to-end deep learning in [2], where it was shown experimentally that a simple feed-forward neural network (FFNN) scheme can outperform pulse-amplitude modulation (PAM) systems with simple feed-forward linear equalizers. Moreover, sequence processing by a recurrent neural network (RNN) was used in [3] to tailor the autoencoder to the dispersive channel properties. This led to an improved system performance with lower computational complexity than PAM transmission with receivers using nonlinear processing [3] or maximum likelihood sequence detection [4].

The framework of learnable communication systems considers the channel as a part of the deep neural network. The application of autoencoders has thus been limited to scenarios where the channel model is known and differentiable. In addition, when applied to an actual system, a transceiver representation learned on a prior assumption of a specific model can result in sub-optimal performance [5, Sec. IV] [2, Sec. VI-C]. Fine-tuning the receiver part of the autoencoder with collected data from measurements was used in [5]. However, optimizing the transmitter without the explicit knowledge of the underlying channel model remains an open problem. To circumvent using a model, a reinforcement learning method was proposed and verified for simple memoryless channels [6]. Alternatively, a generative model of the channel can be obtained via a generative adversarial network (GAN) and used for gradient backpropagation during end-to-end optimization [7–9].

We employ a GAN to acquire a model for our experimental IM/DD test-bed. We perform iterative steps of training the generative model on experimental data and use it to optimize the transceiver. To the best of our knowledge, this is the first end-to-end optimization of an optical communication system via deep learning based on measured data.

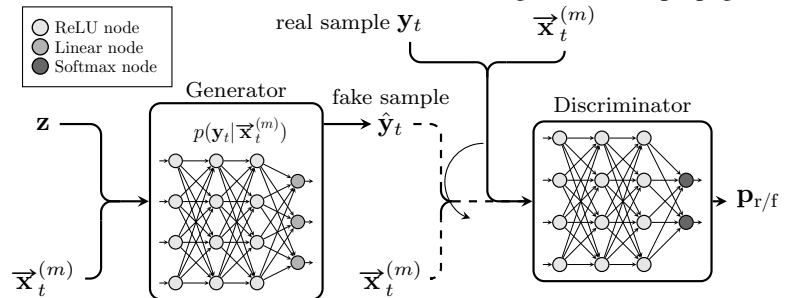


Fig. 1: Schematic of the conditional GAN structure for training of the generative model.

2. Generative Adversarial Network Design for the Optical Autoencoder

Generative adversarial networks are an effective tool for the learning of generative models [10]. A GAN typically employs two artificial neural networks (ANNs) with competing objective functions, which are trained iteratively. The schematic of the GAN we used is shown in Fig. 1. The generator network aims at translating its input into a (possibly high-dimensional) output sample, mimicking some ground truth distribution of the data. The discriminator

Table 1: Generator (bold) and discriminator ANNs.

Layer	Outputs
ReLU	$30n$
ReLU	$20n$
ReLU	$13n$
ReLU	$8n$
ReLU	$5n$
Linear	n
ReLU	$16n$
ReLU	$10n$
ReLU	$6n$
Softmax	2

acts as a binary classifier between real (ground truth) and fake (generated) samples. Related to communication systems, GANs can be used to train a generative model which mimics the function of the channel by approximating its conditional probability distribution. In the framework of autoencoder design, the model can be used in the end-to-end system optimization. In this work we employ a simple FFNN autoencoder. The transmitter maps the input messages $s_t \in \{1, \dots, S\}$, each of which carrying $\log_2(S)$ bits, into blocks (symbols) of n samples, which we denote as $\mathbf{x}_t \in \mathbb{R}^{1 \times n}$. After propagation through the channel, the symbols are fed to the receiver as $\mathbf{y}_t \in \mathbb{R}^{1 \times n}$. Since ISI in optical IM/DD stems from both preceding and succeeding symbols, the goal of the generative model is to approximate $p(\mathbf{y}_t | \bar{\mathbf{x}}_t^{(m)})$, where m , an odd integer, is the modeled symbol memory and $\bar{\mathbf{x}}_t^{(m)} := (\mathbf{x}_{t-(m-1)/2}, \dots, \mathbf{x}_{t+(m-1)/2}) \in \mathbb{R}^{1 \times m \cdot n}$. To mimic the probability distribution, the generator takes as an input the concatenation of a vector of uniformly-distributed random samples $\mathbf{z} \sim \mathcal{U}_{1 \times m \cdot n}^{(0,1)}$ with $\bar{\mathbf{x}}_t^{(m)}$. It transforms it into the fake symbol $\hat{\mathbf{y}}_t = G_\theta((\mathbf{z}, \bar{\mathbf{x}}_t^{(m)}))$, where $\hat{\mathbf{y}}_t \in \mathbb{R}^{1 \times n}$. The discriminator is first fed with the real \mathbf{y}_t and then the fake symbol $\hat{\mathbf{y}}_t$, both conditioned on $\bar{\mathbf{x}}_t^{(m)}$, producing the output probability vectors $\mathbf{p}_r = D_\phi((\mathbf{y}_t, \bar{\mathbf{x}}_t^{(m)}))$ or $\mathbf{p}_f = D_\phi((\hat{\mathbf{y}}_t, \bar{\mathbf{x}}_t^{(m)}))$, respectively, where $\mathbf{p}_{r/f} \in \mathbb{R}^{1 \times 2}$. Table 1 shows the layers of the two ANNs, trained in a supervised manner. The sets of parameters ϕ (discriminator) and θ (generator) are iteratively updated via stochastic gradient descent (SGD), using the Adam algorithm, aimed at minimizing the losses

$$\overline{\mathcal{L}_D}(\phi) = \frac{1}{B} \sum_{i=1}^B [\ell(\mathbf{l}_r, \mathbf{p}_{r,i}) + \ell(\mathbf{l}_f, \mathbf{p}_{f,i})], \quad \overline{\mathcal{L}_G}(\theta) = \frac{1}{B} \sum_{i=1}^B \ell(\mathbf{l}_f, \mathbf{p}_{f,i}), \quad (1)$$

over a batch of $B = 10^3$ elements from the training set, where $\mathbf{l}_r := (0, 1)$ and $\mathbf{l}_f := (1, 0)$ are the labels for real and fake symbols, respectively, and $\ell(\mathbf{x}, \mathbf{y}) = -\sum_i x_i \log(y_i)$ is the cross entropy. Note that the objective of the discriminator is to classify \mathbf{y}_t and $\hat{\mathbf{y}}_t$ correctly. By minimizing $\overline{\mathcal{L}_G}$, the generator learns representations $\hat{\mathbf{y}}_t$ that are statistically similar to \mathbf{y}_t . We train the GAN over 10^4 steps, each of 4 consecutive discriminator updates with the learning rate of 10^{-3} , followed by a generator update with the rate gradually reduced every 200 steps from $5 \cdot 10^{-4}$ to 10^{-5} .

3. End-to-End Optimization Algorithm and Experimental Performance Results

The schematic of the IM/DD test-bed is shown in Fig. 2, which we describe together with the end-to-end system optimization method, shown in Fig. 3 (left). At iteration $k=0$ the transmitter and receiver ANNs are initialized with parameters trained offline, following [2, Sec. III-D], using the IM/DD channel model of [3]. $N=500$ random sequences of $w=8 \cdot 10^4$ messages (\dots, s_t, \dots) , $s_t \in \{1, \dots, 8\}$ (3 bits), are generated and encoded by the transmitter into symbol ($n=6$ samples) sequences $(\dots, \mathbf{x}_t, \dots)$. These are filtered by a 32 GHz LPF and applied to a 84 GSa/s DAC, resulting in a 42 Gb/s data rate. The waveform is fed to an MZM, biased at the quadrature point, modulating a 1550 nm laser. The signal is launched at 1 dBm power into an un-amplified 20 km span of SSF. The power of the received waveform is directly detected by an AC-coupled PIN+TIA and real-time sampled by an ADC. After scaling and offset correction, the digital signal $(\dots, \mathbf{y}_t, \dots)$ is utilized in two ways: it is fed to the receiver for processing, symbol decision and BER calculation using an optimized bit mapping (see [4, Sec. III-C]). Moreover, the transmitted and received sequences are used for GAN training, followed by transceiver optimization. Part of the data ($q=10^3$ elements), grouped as $\mathbf{A} := [s_1; \dots; s_q] \in \mathbb{R}^{q \times 1}$ and $\mathbf{B} := [\mathbf{y}_1; \dots; \mathbf{y}_q] \in \mathbb{R}^{q \times n}$, was used for the single step of transceiver learning within this algorithm iteration. The remaining part, assuming $m=3$, was structured as $\mathbf{C}^{(3)} := [\bar{\mathbf{x}}_{q+2}^{(3)}; \dots; \bar{\mathbf{x}}_{Nw-1}^{(3)}] \in \mathbb{R}^{(Nw-q-2) \times 3n}$,

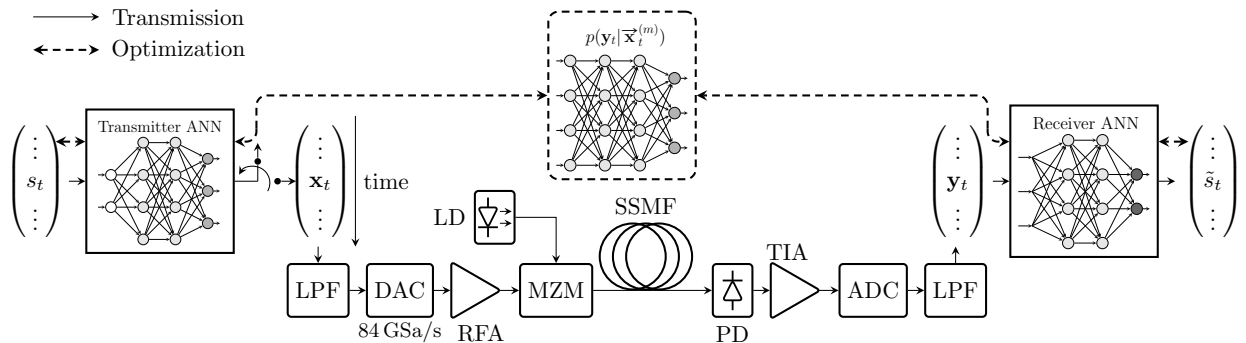


Fig. 2: Schematic of the IM/DD experiment, showing the optimization via the generative channel model. LPF: low-pass filter, DAC/ADC: digital-to-analog/analog-to-digital converter, MZM: Mach-Zehnder modulator, SSF: standard single mode fibre, TIA: trans-impedance amplifier.

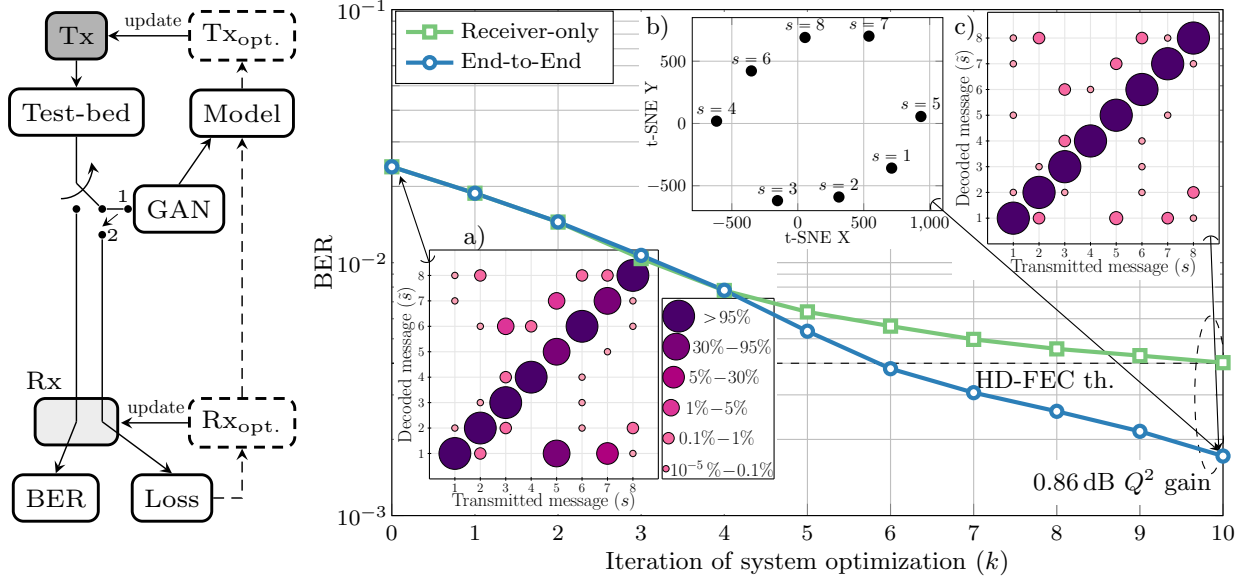


Fig. 3: (Left) Flow chart of the transmission and optimization algorithm. (Right) Experimental BER as a function of optimization iteration. Insets: a) error probabilities at $k=0$; b) 2D t-SNE representation of the waveforms output of the transmitter ANN at $k=10$; c) error probabilities at $k=10$.

with $\bar{\mathbf{x}}_i^{(3)} = (\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})$, $i \in \{q+2, \dots, Nw-1\}$, and $\mathbf{D} := [\mathbf{y}_{q+2}; \dots; \mathbf{y}_{Nw-1}] \in \mathbb{R}^{(Nw-q-2) \times n}$. First, the generative model is trained via the procedure in Sec. 2. Each row i from $\mathbf{C}^{(3)}$ and \mathbf{D} was used to condition the GAN and provide the ground truth symbol, respectively. Next, the transceiver learning is performed, minimizing the cross entropy $\mathcal{L}_{\text{sys}}(\psi_{\text{Tx/Rx}}) = \frac{1}{q} \sum_{i=1}^q \ell(s_i, f_{\text{Rx}}(\mathbf{y}_i))$, with s_i and \mathbf{y}_i rows from \mathbf{A} and \mathbf{B} , via SGD using Adam with 10^{-3} learning rate. To enable the gradient backpropagation, the generator model is applied in lieu of the test-bed, enabling the update of transceiver parameters ($\psi_{\text{Tx/Rx}}$) in a single process. This completes an iteration of the optimization algorithm. The transmitter and receiver representations are updated and we repeat transmission, GAN training and optimization. GAN training required 500 transmissions of different data sequences, which was the time-limiting step in our setup, as well as stable link parameters to ensure the model validity. Thus, as a proof of concept, we performed 10 optimization iterations, which already resulted in performance gain. The BER of the system during optimization is shown in Fig. 3 (right), with a monotonic reduction observed on each iteration. This improvement is illustrated via insets a) and c), showing the error probabilities at $k=0$ and $k=10$, respectively. No errors exceeding the 1% threshold occur at $k=10$, for which b) depicts the modulation constellation in 2D via t-SNE dimensionality reduction [1]. Compared to the previously possible receiver-only optimization on measured data, our method increases the Q^2 -factor by 0.86 dB.

4. Conclusions

We experimentally implemented deep learning of a transceiver based on measured data, an important step towards practical end-to-end optimized transmission. Instead of using an explicit channel model, a generative model of the test-bed was trained to approximate its conditional distribution. It enabled gradient backpropagation in an iteration of the end-to-end system learning. We observed a monotonic decrease in BER on each step of the optimization.

Work under the EU Marie Skłodowska-Curie project COIN (676448/H2020-MSCA-ITN-2015) & UK EPSRC TRANSNET.

References

- [1] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.* **3**(4), 563-575 (2017).
- [2] B. Karanov *et al.*, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.* **36**(20), 4843-4855 (2018).
- [3] B. Karanov *et al.*, "End-to-end optimized transmission over dispersive intensity-modulated channels using bidirectional recurrent neural networks," *Opt. Express* **27**(14), 19650-19663 (2019).
- [4] B. Karanov *et al.*, "Deep learning for communication over dispersive nonlinear channels: performance and comparison with classical digital signal processing," in *Proc. of 57th Allerton Conference on Communication, Control and Computing*, pp. 1-8 (2019).
- [5] S. Dörner *et al.*, "Deep learning-based communication over the air," *IEEE J. Sel. Topics Signal Process.* **12**(1), 132-143 (2018).
- [6] F. Ait Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *Proc. of 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 298-303 (2018).
- [7] T. O'Shea *et al.*, "Approximating the void: Learning stochastic channel models from observation with variational generative adversarial networks," in *Proc. of 2019 International Conference on Computing, Networking and Communications*, pp. 681-686 (2019).
- [8] H. Ye *et al.*, "Channel agnostic end-to-end learning based communication systems with conditional GAN," in *Proc. of GLOBECOM* (2018).
- [9] A. Smith and J. Downey, "A communication channel density estimating generative adversarial network," *NASA Technical Reports 2019*. Accessed on: Oct. 2, 2019. [Online]. Available: <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20190026982.pdf>.
- [10] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. of Conference on Neural Information Processing Systems (NIPS)*, pp. 1-9 (2014).