Fibre Optic Nonlinear Technologies [FONTE] - A European Industrial Doctorate [GA766115]

Document Details

Title	Deliverable D4.3
Deliverable number	D4.3
Deliverable Type	Report (public)
Deliverable title	Capacity limits of NFDM optical fibre networks
Work Package	WP4 - Network applications of the NFT technology
Description	Nonlinear Fourier transform (NFT) has provided fiber-optic communications with a novel powerful tool for data transmission in the nonlinear regime. However, a number of important challenges lie ahead of bringing NFT-based communications out of labs and into operating in real-world systems. As a related approach to the problem, we have considered a deep learning based method to mitigate fiber nonlinearity. In this path, equalization was considered using several methods in the literature as well as a proposed model. The proposed model achieves BER of digital back-propagation and a state-of-the-art CNN, with fewer model parameters and lower memory requirement.
Deliverable due date	30/05/2020
Actual date of submission	11/03/2021
Lead beneficiary	Telecom Paris
Version number	V1.0
Status	FINAL

Dissemination level

PU	Public	X
CO	Confidential, only for members of the consortium (including Commission Services	

Project Details

Grant Agreement	766115
Project Acronym	FONTE
Project Title	Fibre Optic Nonlinear Technologies
Call Identifier	H2020-MSCA-ITN-2017
Project Website	fonte.astonphotonics.uk
Start of the Project	1 June 2018
Project Duration	48 months

Consortium

Aston University

NOKIA Bell Labs

TELECOM ParisTech

TU Delft

DTU Technical University of Denmark

EC Funding

European Commission

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 766115

Executive Summary

The demand for information traffic has risen significantly in recent years, leading to the so-called "capacity crunch" in optical fiber. Nonlinear frequency-division multiplexing (NFDM) has been proposed to help increase transmission rates. This method is similar to the frequency division multiplexing, where Fourier transform is replaced with the nonlinear Fourier transform (NFT). The NFT is based on the key observation that there is a linear operator associated with the propagation equation whose spectrum is invariant during propagation. However, a number of important challenges lie ahead of bringing NFT-based communications out of the labs and into operating in real-world systems. For instance, computing NFT efficiently and accurately at high powers is challenging. A variety of approaches have been proposed to address these challenges and improve NFDM. We, however, consider a machine learning approach to compensate nonlinearity in fiber-optic communications.

In this deliverable, after reviewing the current deep learning trends in fiber-optic communications, we consider the equalization in point-to-point long-haul dispersive fiber-optic communication systems using chromatic dispersion compensation, digital back-propagation, multilayer perceptrons (MLPs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), and a proposed neural architecture. We discuss the experimentation results of these solutions and the gain of our proposed approach. It is explored that with lower complexity, the proposed model achieves the BER of the state-of-the-art CNN approach and DBP.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

AiPT	Aston Institute of Photonic Technologies
BP	Backpropagation
BER	Bit Error Rate
CDC	Chromatic Dispersion Compensation
CNN	Convolutional Neural Network
DBP	Digital-Backpropagation
DL	Deep Learning
DNN	Deep Neural Network
ESR	Early Stage Researcher
EC	European Commission
EID	European Industrial Doctorates
FONTE	Fibre Optic Nonlinear Technologies
GAN	generative adversarial network
ML	Machine Learning
MLP	Multilayer perceptron
NN	Neural Network
NBL	Nokia Bell Labs
NFT	Nonlinear Fourier transform
NFDM	Nonlinear Frequency-Division Multiplexing
NLS	Nonlinear Schrödinger
RL	Reinforcement Learning
RNN	Recurrent Neural Networks
RX	Receiver
SSFM	Split-Step Fourier Method
TX	Transmitter
WDM	Wavelength Division Multiplexing

1 Introduction

Stochastic nonlinear Schrödinger (NLS) equation models signal propagation in single-mode optical fiber with distributed amplification. This equation can be normalized as [1, Eq. 3]

$$j\frac{\partial q}{\partial z} = \frac{\partial^2 q}{\partial t^2} + 2|q|^2 q + n(t,z), \tag{1}$$

where $q(t,z)$ is the complex envelope of the signal, as a function of time t and distance z, and n(t,z) is zero-mean white circular symmetric complex Gaussian noise.

Nonlinear frequency-division multiplexing (NFDM) consists of three steps in both transmitter (TX) and receiver (RX). For a multi-user system with N_u users, each sending N_s symbols, total (linear) bandwidth B Hz, and (total) average power P, NFDM, firstly, computes

$$u(\tau,0) = \sqrt{2}\sum_{k=-\frac{N_u}{2}}^{\frac{N_u}{2}-1}\sum_{k=-\frac{N_s}{2}}^{\frac{N_s}{2}-1} s_l^k \phi(\tau\ell - T_0)e^{j2\pi k W_0 \tau} \tag{2}$$

where $\phi(t)$ is a root-raised-cosine function with the bandwidth W_0 Hz and the roll-off factor r. τ is the generalized time in Fourier transform relation with the generalized frequency $\lambda \in \mathbb{R}$. $T_0 = 1/W_0$, and $\{s_l^k\}_l$ are symbols of user k chosen from a constellation Ξ . In the second phase, NFDM calculates the transmit signal in the nonlinear Fourier domain as follows:

$$\hat{q}(\lambda,0) = (e^{|U(\lambda,0)|^2} - 1)^{\frac{1}{2}} e^{j\angle U(\lambda,0)}. \tag{3}$$

In (3), $U(\lambda,0) = \frac{1}{\sqrt{2}}\mathfrak{F}(u(\tau,0))$, where $\mathfrak{F}$ denotes Fourier transform. This process is followed by the final operation $q(t,0) = \text{INFT}(\hat{q}(\lambda,0))$. [1]

Although NFT is a novel transmission method paving the way for the advancement of new modulation and multiplexing methods in optical fiber communication, it faces a number of limitations in practice. One obstacle is the implementation penalty owing to distortions introduced by hardware components. Furthermore, computing NFT efficiently and accurately at high powers is challenging.

As alternative solutions, neural network based receivers [4–11] and end to end deep learning of communication systems [12–14] are emerging fields addressing the nonlinearity problem in optical fiber communication via a deep learning approach. In the following sections, we discuss these areas.

2 Learning by neural networks

Neural networks are the enablers of deep learning owing to their ability to capture the complex input-output correlations and mathematical structures in the presented data. Neural networks are hierarchically organized into groups of basic processing units or neurons.

In a general grouping, based on the scheme that information propagate fro the input layer to the output, neural networks are categorized into groups:

- **Feed-forward Neural Networks** (FNNs). In FNNs each neuron is only linked to the neurons in the next layer. Thefore, the information only propagates forward from the input layer to the output without any feedback loop.
- **Recurrent Neural Networks** (RNNs). In contrast to FNNs, there exists feedback links in RNNs and therefore the information can also be passed to the previous layers in the network via feedback links.

Although these two categories are different in architecture, for both of them the back-propagation algorithm [15] is used for the training process to find the best set of weights. In back-propagation algorithm, the error is calculated at each iteration t and then it is back-propagated to the preceding layers in order to adjust their weights by the gradient descent as

$$W_{t+1} = W_t - \alpha\Delta\hat{L}(W_t), \tag{4}$$

[1] The achievable rates of the linear and nonlinear FDM are compared in [2] and [3] for normal dispersion and anomalous dispersion fiber, respectively.

where $\hat{L}$ is the calculated resulting loss caused by the current weights, $\Delta\hat{L}$ is the gradient of $\hat{L}$, and $\alpha > 0$ is the learning rate.

Multi-layer perceptrons (MLPs), also called dense neural networks, and convolutional neural networks (CNNs) are the two main architectures in the category of the FNNs. These two architectures were discussed in the deliverable D4.2, thus we do not explain them in this document again. In the following subsection the RNN architecture is briefly described.

2.1 Recurrent neural networks

The RNNs are developed to operate on the temporal sequences of data with correlated samples. In RNNs, the links between the nodes forms a loop, thus the information does not merely propagate forward. An RNN consists of recurrent layers composed by recurrent cells. A recurrent cell state is influenced by the current input as well as the past (and the next) states via the feedback connection. The general mathematical expression of a recurrent cell is as follows:

$$\mathbf{h}_t = \sigma(\mathbf{w_h}\mathbf{h}_{t-1} + \mathbf{w_x}\mathbf{x}_t + \mathbf{b}) \tag{5}$$

$$\mathbf{y}_t = \sigma'(\mathbf{h}_t) \tag{6}$$

where $\mathbf{x}_t$, $\mathbf{y}_t$, $\mathbf{h}_t$ are respectively, the input, the output, and the hidden state (the recurrent information) of the time step t, $t = 1, 2, \cdots$. Also, $\mathbf{w_h}$, $\mathbf{w_x}$, $\mathbf{b}$ denotes the weight matrix for the recurrent state, the weight matrix for the input state and the bias vector shared among all the recurrent cells. $\sigma(.)$ and $\sigma'(.)$ are two different activation functions.

The recurrent cell discussed above is called Simple RNN cell. There are variations of this format as well. The two primary variations are Long-short-term-mermory (LSTM) [16] cell and Gated recurrent unit [17] (GRU) cell. The detail of these variations is out of the scope of this document.

Different forms of RNNs can be designed by arranging the recurrent cells in different formats. Therefore an RNN is mainly characterized by the recurrent cell structure and their arrange (network format) [18]. The main RNN network formats are many-to-many (or seq2seq), many-to-one (many inputs, one output), and one-to-many artictectures. The input of the many-to-many format is a sequence of data and its output is also a sequence of data. The input of the many-to-one format is a sequence of inputs but the output is a single value (the output of last recurrent cell). In contrast, in the one-to-many format, the input is a single value and the output is a sequence of data.

In the field of neural based receiver for optical fiber communication systems the goal is to design the most accurate (in terms of bit-error-rate) and efficient neural network (in terms of performance-complexity trade-off) capable of equalizing the channel effects such as chromatic dispersion, Kerr-nonlinearity, polarization mode dispersion, etc [19]. For example, [20] uses an MLP to perform equalization in a 200 Gbps DP-16QAM optical transmission system for distances 100km to 400 km. This approach has reported a Q-factor gain of 0.5 dB to 1 dB compared to linear equalization. [4] uses an MLP to equalize the received waveform in the nonlinear Fourier domain. This approach has demonstrated a six-times BER improvment compared to absence of a NN-equalizer for a 1000km fiber.

[5] and [21] use an RNN for equalization with focus on nonlinear and intra-channel effects mitigation, showing that RNN based transcrivers outperms the MLPs. [6–8] has proposed an CNN based for classification of different PAM classes based on the received signal as input. [9–11] also has proposed a CNN approach, but their model is based the computational graph generated by SSFM. By this technique, termed learned DBP (LDBP), they have succeded in achieving the BER of DBP [22] with almost 50% lower complexity.

3 End to end deep learning of communication systems

End to end learning of communication systems for optical fiber channels has gained attention recently [23]. The purpose of this approach is to learn full transmitter and receiver implementations optimized for a certain channel model given a specific objective performance measure. In this approach, the transmitter and receiver are implemented by neural networks and the systems is interpreted as an autoencoder [24, Ch. 14]. The schematic representation of the system is illustrated in Fig. 1. In this figure, $\mathbf{m} = (m_1, ..., m_N)$ is a sequence of bits. The neural transmitter maps this sequence to the sequence of symbols $\{S_i\}_1^n$. Then they are pulse shaped with the

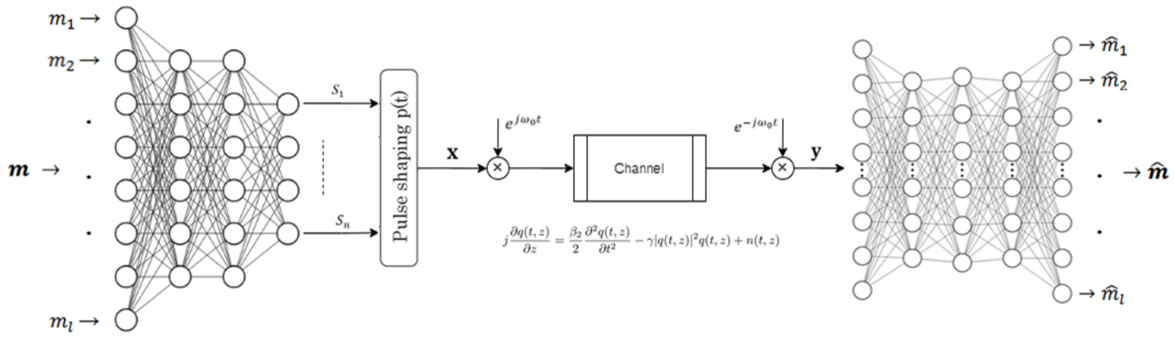

$m_1 \rightarrow$
$m_2 \rightarrow$
$\boldsymbol{m} \rightarrow$
$m_l \rightarrow$
s_1
s_n
Pulse shaping p(t)
x
$e^{j\omega_0 t}$
Channel
$e^{-j\omega_0 t}$
y
$j\frac{\partial q(t,z)}{\partial z} = \frac{\beta_2}{2}\frac{\partial^2 q(t,z)}{\partial t^2} - \gamma|q(t,z)|^2 q(t,z) + n(t,z)$
$\rightarrow \hat{m}_1$
$\rightarrow \hat{m}_2$
$\rightarrow \hat{\boldsymbol{m}}$
$\rightarrow \hat{m}_l$

Fig. 1. Schematic representation of an autoencoder for optical fiber communication systems

filters $\{\phi_i(t)\}_1^n$ to form the transmitted signal $\mathbf{x}$. At the RX, the neural reciver try to recover $\mathbf{m}$ but processing the received signal $\curvearrowright$ as input.

However, although end-to-end learning of communications systems [12], is theoretically an interesting idea, the practicability of it is in doubt. This is in view of the fact that this method requires the channel model to be differentiable [25]. More clearly, the gradient of the instantaneous channel transfer function has to be precisely known. For an actual system, this is quite rarely feasible as the channel is generally a black box for which only inputs and outputs can be observed. In addition, the channel typically contains certain components of the transceiver, such as quantization, which are non-differentiable and therefore prevent gradient-based training via backpropagation [26].

One proposed solution is to assume a common channel model, e.g., a Gaussian model. The idea is to first learn on the basis of this generic model and then refine the receiver, i.e., the weights of the decoding component of the NN, according to the received signals. This strategy has been discussed in [27].

Using generative adversarial networks (GANs) [28] is another proposed approach [29], [30] to mimic the channel. In the following subsection, we discuss the general of idea GANs.

3.1 Generative adversarial networks

GANs are algorithmic architectures consists of two opposing neural networks, a generative NN and a discriminative NN. In general, a discriminative NN learns the boundary between classes, and a generative NN models the distribution of individual classes. To put it simply, discriminative NNs map features to labels while generative NNs capture the probability of features given a label or category. In GANs, these two components, let's call them G and D, compete against each other so that the GAN generates new, synthetic instances of data that can not be distinguished from real data. In this algorithm, G captures the data distribution, and D estimates the probability that a sample came from the training data rather than G. The training objective of G is to maximize the probability of D making a mistake [28], and the training objective of D is to maximize the probability of detection when a data came from G. This framework corresponds to a minimax two-player game, with value function $V(G, D)$:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data(x)}}[\log D(x)] + \mathbb{E}_{z\sim p_{z(z)}}[\log(1 - D(G(z)))], \tag{7}$$

where z is the generated data. According to the Nash's Theorem this framework has at least one solution (Nash equilibrium) [31].

In [29] and [30], GANs are utilized to generate an artificial channel model that approximates the actual channel distribution and thus can be used for autoencoder training.

Addressing the mentioned challenge of the autoencoder idea via a reinforcement learning (RL) solution is another approach to the problem [32], [33]. The principles of these solutions and RL and is elaborated in the following subsection.

3.2 Reinforcement learning

Reinforcement learning is about an agent interacting with the environment to learn an optimal policy for sequential decision-making problems, by trial and error. Trial-and-error learning is associated with so-called long-term rewards. In RL, the algorithm (agent) analyses the current situation (state), makes a decision, and receives feedback (reward) from the environment. Positive feedback denotes a reward, and negative feedback denotes a punishment due to the wrong decision. The main objective of RL is to determine the best sequence of decisions that will enable an agent to solve a problem while optimizing long-term rewards [34, Ch. 1]. This is carried out via the following process.

At each time step t, the agent receives a state s_t in the state space $\mathcal{S}$ and selects an action a_t from the action space $\mathcal{A}$, according to the policy π. A policy is a mapping from states to probabilities of selecting each possible action, together with their associated rewards. In the event of an episodic problem, this cycle proceeds until the agent enters the terminal state, where it restarts. The return (accumulated reward) of a policy for the state s_t is calculated as

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k}, \tag{8}$$

where T is the time of termination, and r_t is the reward of the action at time step t, based on the policy. $\gamma \in [0, 1)$ is the discount factor. The goal of the agent is to maximize the expectation of the return from each state. This is performed by estimating the value function. The value function measures the expected return for a state under a policy. Therefore, mathematically, the value of a state s under a policy π, denoted by $v_\pi(s)$, is defined as

$$v_\pi(s) = \mathbb{E}_\pi[R_t|s_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{T} \gamma^k r_{t+k}|s_t = s\right]. \tag{9}$$

Similarly, the value of taking action a at state s under policy π, is defined by the action-value function for policy π as

$$q_\pi(s, a) = \mathbb{E}_\pi[R_t|s_t = s, a_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{T} \gamma^k r_{t+k}|s_t = s, a_t = a\right]. \tag{10}$$

A foundational property of the value functions in RL is that for any policy π and any state s, the following consistency condition holds between the value of s and the value of its possible successor states [34, Ch. 3]

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[R_t|s_t = s] \\ &= \mathbb{E}_\pi[r_{t+1} + \gamma R_{t+1}|s_t = s] \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma \mathbb{E}_\pi[R_{t+1}|s_{t+1} = s']\right] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_\pi(s')\right], \end{aligned} \tag{11}$$

where $\pi(a|s)$ denotes the probability that $a_t = a$ if $s_t = s$, under policy π.

Solving a RL task means finding the optimal policy π_*, whose expected return for all the states is higher than or equal to all the other policies. Optimal policies have the same state-value function, called the optimal state-value function. This function is defined as $v_*(s) = \max_\pi v_\pi(s)$ for all $s \in \mathcal{S}$. Optimal policies also have the same optimal action-value function $q_*(s, a) = \max_\pi q_\pi(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. $\mathcal{A}(s)$ is the set of all available actions for the state s. Therefore,

$$q_*(s, a) = \mathbb{E}_\pi[R_{t+1} + \gamma v_*(s_{t+1})|s_t = s, a_t = a]. \tag{12}$$

According to the Bellman optimality equation [35] and by (12),

$$\begin{aligned} v_*(s) &= \max_{a\in\mathcal{A}(s)} q_*(s, a) \\ &= \max_a \mathbb{E}_\pi[R_{t+1} + \gamma v_*(s_{t+1})|s_t = s, a_t = a] \\ &= \max_a \sum_{s',r} p(s', r|s, a) \left[r + \gamma v_*(s')\right]. \end{aligned} \tag{13}$$

Various solutions have been proposed to achieve π_* through iterations of policy improvement, the prominent of which are Q-learning [36], policy gradient [37] , DQN [38], and DDPG [39]. The policy improvement process finishes when v_* is reached [34, Ch. 4], which is the case when

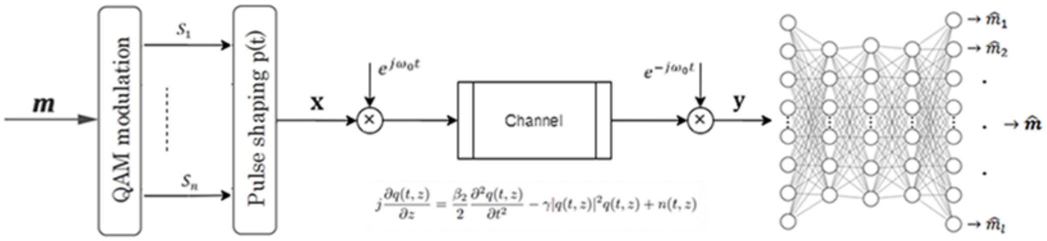


Fig. 2. Schematic representation of the implemented optical fiber communication systems with the NN-based receiver

TABLE 1. Fiber parameters

n_{sp}	1	spontaneous emission factor
α	$0\,\mathrm{km}^{-1}$	fiber loss
γ	$1.27 \quad \mathrm{W}^{-1}\mathrm{km}^{-1}$	nonlinearity parameter
β_2	$2.167 \times 10^{-23}\,\mathrm{s}^2\mathrm{km}^{-1}$	2^{nd}-order dispersion coefficient
σ_0^2	$5.906 \times 10^{-14}\,\mathrm{Wkm}^{-1}\mathrm{Hz}^{-1}$	noise PSD

$$\forall\, s \in \mathcal{S},\ \nexists\, a \in \mathcal{A}(s)\,[(a \neq \pi(s)) \wedge (q_\pi(s,a) > v_\pi(s))]\,. \tag{14}$$

[32], [33] has used RL through a policy gradient approach for autoencoder training. In this approach, the transmitter is regarded as an agent whose actions are the transmitted signals. Therefore, the transmitter is optimized through the process of receiving rewards via a feedback link connected to the actions. However, as a matter of fact, as it is apparent from what was discussed, RL is famous for being data-hungry, the overload of states, subject to instability in the learning process, and a laggard in terms of performance [40]. In consequence, although this approach is theoretically very interesting, the viability of it is intractable for real communication systems.

4 Research and Experiments

The research field of neural based receivers was followed. We assumed a single-mode single-polarization point-to-point optical fiber communication system, shown in Fig.2, whose fiber is modeled by NLS equation

$$\frac{\partial q(t,z)}{\partial z} = -\frac{j\beta_2}{2}\frac{\partial^2 q}{\partial t^2} + j\gamma|q(t,z)|^2 q(t,z) + n(t,z), \quad 0 \leq z \leq \mathcal{L}. \tag{15}$$

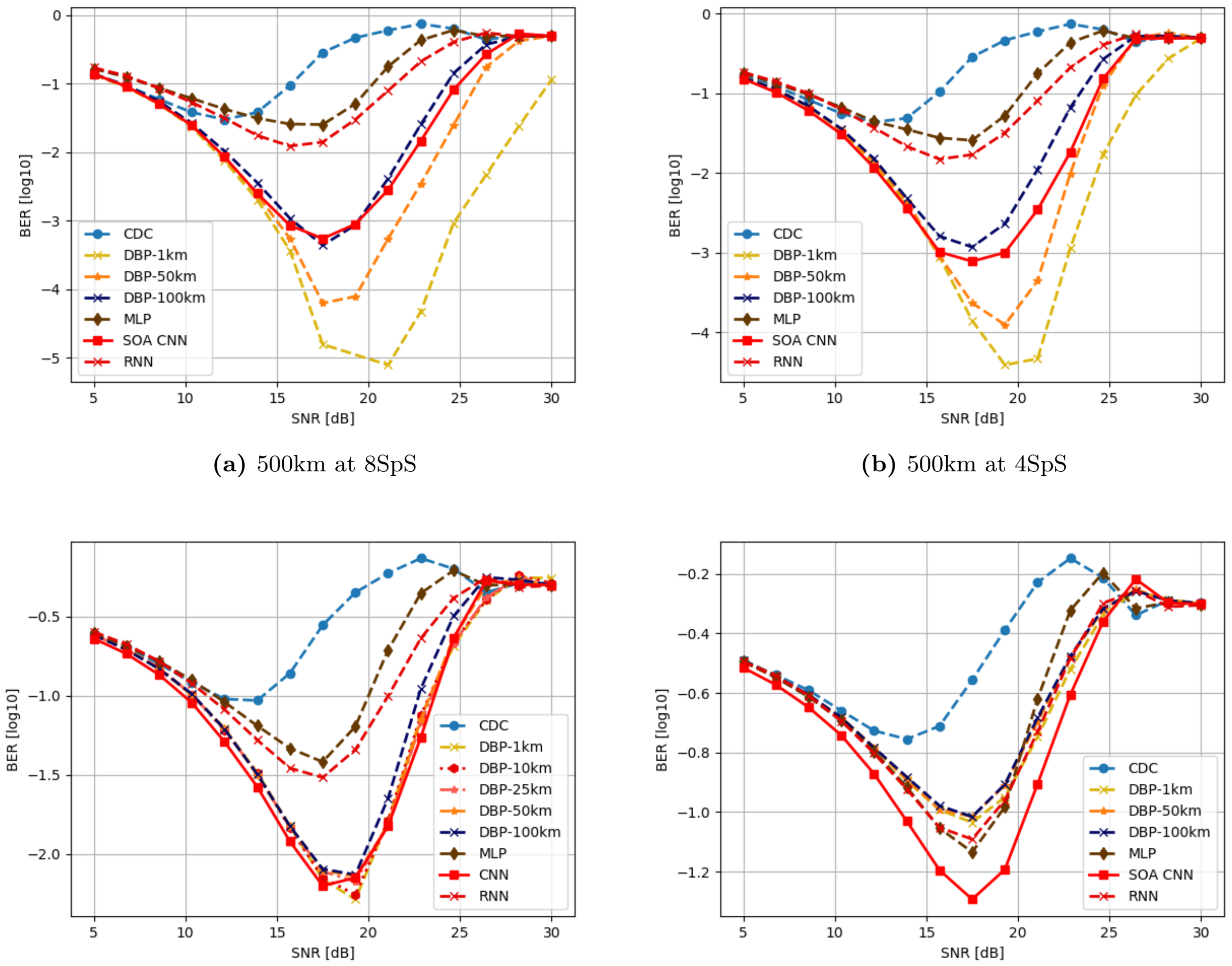
Here, $q(t,z)$ is the complex envelope of the signal propagating in fiber as a function of time t and distance z, β_2 is chromatic dispersion coefficient, γ is the nonlinearity parameter, $\mathcal{L}$ is fiber length and $n(t,z)$ is zero-mean white circularly symmetric complex Gaussian noise with the power spectral density σ^2. We implemented NLS equation using split-step Fourier method (SSFM). The fiber loss was presumed to be perfectly compensated by distributed Raman amplification.

In this system, the transmitter takes a sequence of bits $\mathbf{m} = (m_1, ..., m_N)$, where $(m_{2i}, m_{2i+1}) \in \{(0,1),(1,0)\}$, $i = 0, \ldots, N/2$, maps it to a sequence of symbols drawn from a quadrature amplitude (QAM) constellation, and modulates them using the pulse-amplitude modulation (PAM)

$$q(t,0) = \sum_{i=-\infty}^{\infty} S_i p(t - iT), \tag{16}$$

At the receiver, a neural equalizer and detector takes $q(t,\mathcal{L})$, denoted by $\mathbf{y}$ in the figure, and recovers the sequence of bits $\mathbf{m}$, denoted by $\hat{\mathbf{m}}$.

The system was set based on 16-QAM transmission at 25 Gbaud using RRC pulses (roll-off factor 0.1), and with the fiber parameters mentioned in Table. 1.



(a) 500km at 8SpS

(b) 500km at 4SpS

(c) 500km at 2SpS

(d) 500km at 1SpS

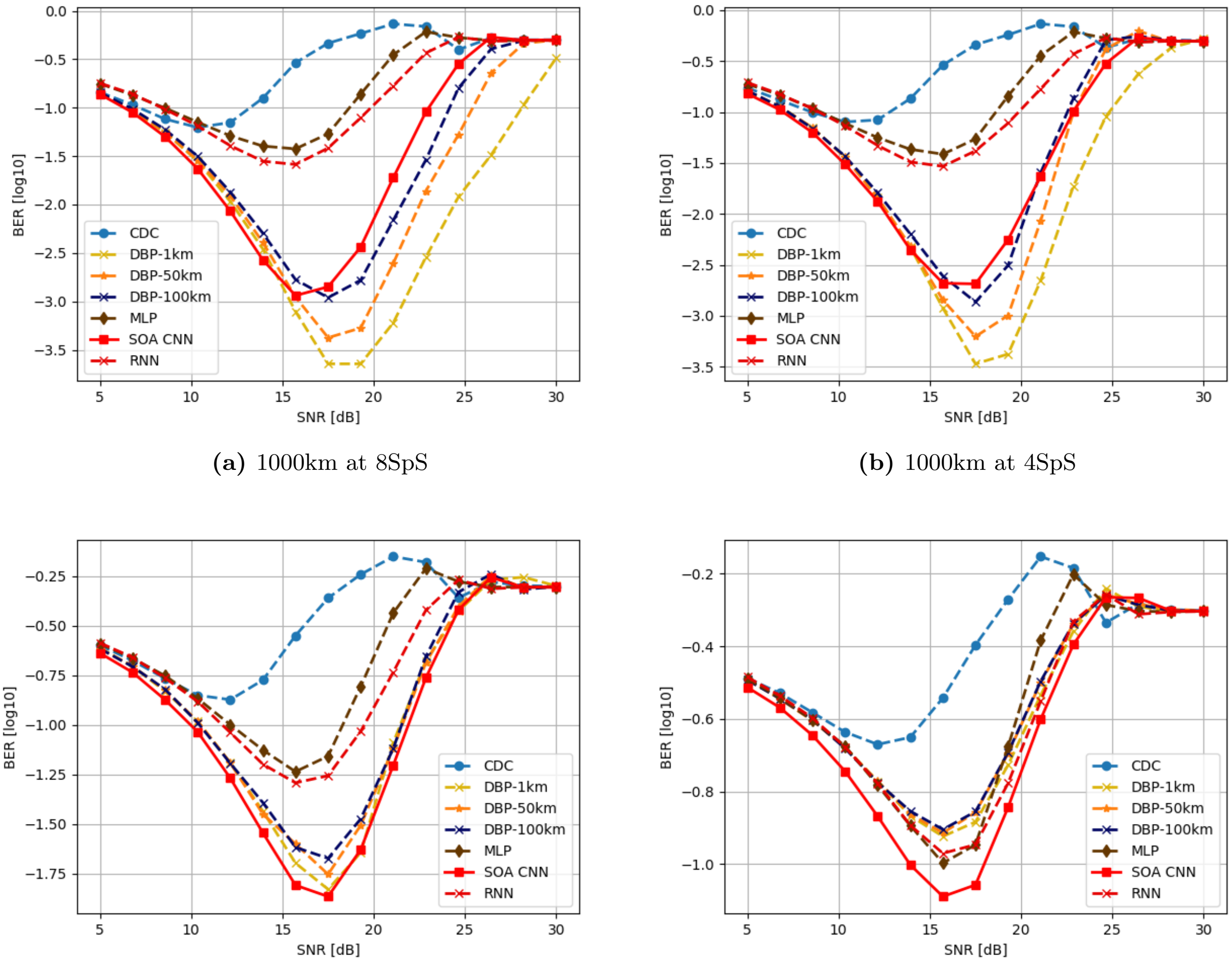
Fig. 3. The BER of the equalizers for the 500 km fiber-optic link, as a function of SNR for several values of sampling rates at RX and DBP step size.

We considered channel equalization in this system using chromatic dispersion compensation (CDC), digital back-propagation DBP (with step sizes 1km, 50km, 100km), MLPs, and the state-of-the-art CNN approach [11]. All models were created and trained using Tensorflow [41] in python. The BER performance of these approaches as a function of SNR, and simulation step size (for DBP), for a 500km, 1000km, and 2000km fiber-optic link, is illustrated in Fig.3, Fig.4, Fig.5, respectively.

Following the implementation of the methods in the literature review, we contemplated a neural network based equalizer model achieving roughly the BER performance of the state-of-the-art CNN approach and DBP with lower number of parameters and memory requirement for training. This model was based on the idea that although CNNs are powerful in capturing short temporal-distance features, they may not efficiently capture long-distance features. The details of this approach is confidential at the moment of writing this document.

Appendix

As discussed, although end-to-end deep learning of communications systems is theoretically an exciting idea, it faces some critical challenges regarding the channel transfer function. In this regard, during our research we also discerned a deep representation learning method that instead of learning the channel, learns the structure of the sequence of symbols that incurred the minimum damage while passing the channel. This structure will then try



(a) 1000km at 8SpS

(b) 1000km at 4SpS

(c) 1000km at 2SpS

(d) 1000km at 1SpS

Fig. 4. The BER of the equalizers for the 1000km fiber-optic link, as a function of SNR for several values of sampling rates at RX and DBP step size.

to be transferred to the sequence of symbols at TX to vaccinate them against error throughout the channel. That is to say, the content is preserved, but the pattern is taken to the pattern of Superior set $\mathfrak{S}$ in which there are sequences of symbols that were resilient against channel errors.

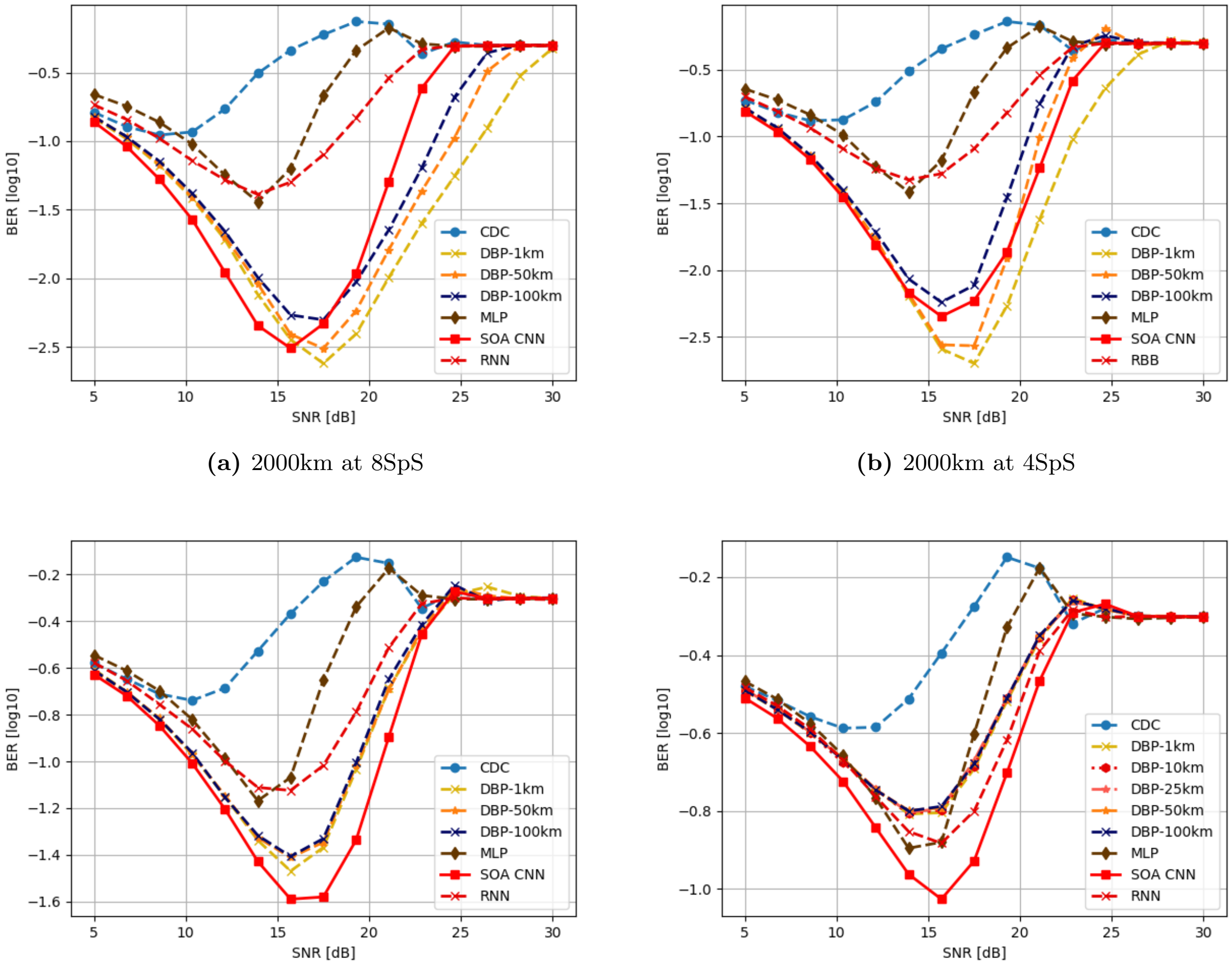
The detail of this approach is confidential at the time of writing this document. But To briefly discuss the general mathematical idea, the pattern similarity can be able to be measured by the cosine similarity of the featurized representation (embedding) of the input sequence x and the samples in $\mathfrak{S}$ as follows:

$$Similarity(x, \mathfrak{S}) = \frac{1}{|\mathfrak{S}|} \sum_{s \in \mathfrak{S}} \frac{x.s}{||x||||s||} = \frac{1}{|\mathfrak{S}|} \sum_{s \in \mathfrak{S}} \frac{\sum_1^n x_i s_i}{\sqrt{\sum_1^n x_i^2}\sqrt{\sum_1^n s_i^2}}. \tag{17}$$

The resulting similarity varies in the interval $[-1, 1]$. -1 denotes totally opposite, 1 denotes completely the same, 0 indicates orthogonality or decorrelation, and in-between values indicate intermediate similarity or dissimilarity. The content similarity could be obtained by measuring the mutual information of the random variable of the output sequence Y and the random variable of the input sequence X:

$$I(X;Y) = \int_{y \in Y} \int_{x \in X} P_{XY}(x, y) \log \frac{P_{XY}(x, y)}{P_X(x) P_Y(y)} dxdy, \tag{18}$$

where $P_{XY}(x, y)$ is the joint probability distribution and $P_X(x)$ and $P_Y(y)$ are the marginals. In the discrete



(a) 2000km at 8SpS

(b) 2000km at 4SpS

(c) 2000km at 2SpS

(d) 2000km at 1SpS

Fig. 5. The BER of the equalizers for the 2000km fiber-optic link, as a function of SNR for several values of sampling rates at RX and DBP step size.

domain we have

$$\begin{aligned} I(X;Y) &= \sum_{y\in Y}\sum_{x\in X} P_{XY}(x,y)\log\frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} \\ &= D_K L(P_{XY}||P_X P_Y) = E_{P_{XY}}\log\frac{P_{XY}}{P_X P_Y}, \end{aligned} \tag{19}$$

where E_P is the expected value over the distribution P. We know $P_X(x) = \sum_y P_{XY}(x,y)$ and $P_Y(y) = \sum_x P_{XY}(x,y)$.

The idea is to push the pattern similarity of the sequence at RX to $\mathfrak{G}$ while maintaining a high content similarity to its original form.

References

[1] M. I. Yousefi and F. R. Kschischang, "Information transmission using the nonlinear Fourier transform, part i, ii, iii," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4312–4369, 2014.

[2] M. Yousefi and X. Yangzhang, "Linear and nonlinear frequency-division multiplexing," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 478–495, 2020.

[3] X. Yangzhang, M. I. Yousefi, A. Alvarado, D. Lavery, and P. Bayvel, "Nonlinear frequency-division multiplexing in the focusing regime," in *Proceedings of the Optical Fiber Communication Conference*, p. Tu3D.1, 2017.

[4] O. Kotlyar, M. Pankratova, M. Kamalian-Kopae, A. Vasylchenkova, J. E. Prilepsky, and S. K. Turitsyn, "Combining nonlinear fourier transform and neural network-based processing in optical communications," *Optics Letters*, vol. 45, no. 13, pp. 3462–3465, 2020.

[5] S. Deligiannidis, A. Bogris, C. Mesaritakis, and Y. Kopsinis, "Compensation of fiber nonlinearities in digital coherent systems leveraging long short-term memory neural networks," *arXiv preprint arXiv:2001.11802*, 2020.

[6] P. Li, L. Yi, L. Xue, and W. Hu, "56 gbps im/dd pon based on 10g-class optical devices with 29 db loss budget enabled by machine learning," in *Optical Fiber Communications Conference and Exposition*, pp. 1–3, 2018.

[7] C. Chuang, L. Liu, C. Wei, J. Liu, L. Henrickson, W. Huang, C. Wang, Y. Chen, and J. Chen, "Convolutional neural network based nonlinear classifier for 112-gbps high speed optical link," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1–3, 2018.

[8] P. Li, L. Yi, L. Xue, and W. Hu, "100gbps im/dd transmission over 25km ssmf using 20g-class dml and pin enabled by machine learning," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1–3, 2018.

[9] C. Häger, H. D. Pfister, R. M. Bütler, G. Liga, and A. Alvarado, "Model-based machine learning for joint digital backpropagation and pmd compensation," in *Proceedings of the Optical Fiber Communication Conference*, p. W3D.3, 2020.

[10] C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, pp. 1–3, 2018.

[11] C. Häger and H. D. Pfister, "Deep learning of the nonlinear Schrödinger equation in fiber-optic communications," in *Proceedings of the IEEE International Symposium on Information Theory*, pp. 1590–1594, 2018.

[12] 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.

[13] 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.

[14] 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 Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3133–3143, 2020.

[15] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, pp. 65–93, Elsevier, 1992.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[18] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[19] T. Uhlemann, S. Cammerer, A. Span, S. Dörner, and S. t. Brink, "Deep-learning autoencoder for coherent and nonlinear optical communication," *arXiv preprint arXiv:2006.15027*, 2020.

[20] C. Catanese, R. Ayassi, E. Pincemin, and Y. Jaouën, “A fully connected neural network approach to mitigate fiber nonlinear effects in 200g dp-16-qam transmission system,” in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, 2020.

[21] B. Karanov, M. Chagnon, V. Aref, D. Lavery, P. Bayvel, and L. Schmalen, “Optical fiber communication systems based on end-to-end deep learning,” *arXiv preprint arXiv:2005.08785*, 2020.

[22] S. Wahls, S. T. Le, J. E. Prilepsk, H. V. Poor, and S. K. Turitsyn, “Digital backpropagation in the nonlinear Fourier domain,” in *Proceedings of the IEEE 16th International Workshop on Signal Processing Advances in Wireless Communications*, pp. 445–449, 2015.

[23] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, 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.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[25] R. Fritschek, R. F. Schaefer, and G. Wunder, “Deep learning for channel coding via neural mutual information estimation,” in *Proceedings of the IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications*, pp. 1–5, 2019.

[26] F. A. Aoudia and J. Hoydis, “Model-free training of end-to-end communication systems,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.

[27] 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.

[28] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2672–2680, 2014.

[29] H. Ye, G. Y. Li, B. F. Juang, and K. Sivanesan, “Channel agnostic end-to-end learning based communication systems with conditional gan,” in *Proceedings of the IEEE Globecom Workshops*, pp. 1–5, 2018.

[30] T. J. O’Shea, T. Roy, N. West, and B. C. Hilburn, “Physical layer communications system design over-the-air using adversarial networks,” in *Proceedings of the 26th European Signal Processing Conference*, pp. 529–532, 2018.

[31] J. N. Webb, *Game theory: decisions, interaction and Evolution*. Springer Science & Business Media, 2007.

[32] F. A. Aoudia and J. Hoydis, “End-to-end learning of communications systems without a channel model,” in *Proceedings of the 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 298–303, 2018.

[33] M. Goutay, F. A. Aoudia, and J. Hoydis, “Deep reinforcement learning autoencoder with noisy feedback,” *arXiv preprint arXiv:1810.05419*, 2018.

[34] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[35] R. Bellman, “On the theory of dynamic programming,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 38, no. 8, p. 716, 1952.

[36] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[37] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, pp. 1057–1063, 2000.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[39] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.

[40] G. Dulac-Arnold, D. Mankowitz, and T. Hester, “Challenges of real-world reinforcement learning,” *arXiv preprint arXiv:1904.12901*, 2019.

[41] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th symposium on operating systems design and implementation*, pp. 265–283, 2016.