

# OCATA: a deep-learning-based digital twin for the optical time domain

D. SEQUEIRA,<sup>1</sup>  M. RUIZ,<sup>1</sup>  N. COSTA,<sup>2</sup> A. NAPOLI,<sup>3</sup> J. PEDRO,<sup>2,4</sup>  AND L. VELASCO<sup>1,\*</sup> 

<sup>1</sup>Universitat Politècnica de Catalunya, Barcelona, Spain

<sup>2</sup>Infinera Unipessoal Lda., Carnaxide, Portugal

<sup>3</sup>Infinera, Munich, Germany

<sup>4</sup>Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal

\*Corresponding author: luis.velasco@upc.edu

Received 4 October 2022; revised 22 December 2022; accepted 22 December 2022; published 12 January 2023

The development of digital twins to represent the optical transport network might enable multiple applications for network operation, including automation and fault management. In this work, we propose a deep-learning-based digital twin for the optical time domain, named OCATA. OCATA is based on the concatenation of deep neural network (DNN) modeling of optical links and nodes, which facilitates representing lightpaths. The DNNs model linear and nonlinear noise, as well as optical filtering. Additional DNN-based models are proposed to extract useful lightpath metrics, such as lightpath length, number of optical links, and nonlinear fiber parameters. OCATA exhibits low complexity, thus making it ideal for real-time applications. Illustrative results for the application of OCATA to disaggregated and mixed disaggregated-proprietary optical network scenarios reveal remarkable accuracy. © 2023 Optica Publishing Group

<https://doi.org/10.1364/JOCN.477341>

## 1. INTRODUCTION

During the past few years, optical networks have evolved from static and manually operated networks to more dynamic and flexible ones to support traffic dynamicity requirements. This evolution has been possible, among others, thanks to technologies like advanced reconfigurable optical add/drop multiplexers (ROADMs) architectures in the data plane and software-defined networking and monitoring and data analytics in the control plane [1,2].

In addition, with the introduction of 5G and beyond services, more autonomous, efficient, and reliable optical networks are required in all network segments, from core to metro and access [3]. In that regard, machine learning (ML) has received special attention, and several works have demonstrated its benefit to improve optical communications and for network automation (see e.g., [4–6]). The low computational requirements of ML models for inference, once trained, make them very attractive to solve hard computational problems. For example, the authors of [7] proposed deep neural networks (DNNs) for nonlinear interference (NLI) noise equalization, and the authors of [8] proposed ML-based models for quality of transmission (QoT) estimation in single-domain network scenarios, while the authors of [9] extended it to multidomain scenarios. In contrast, the well-known analytical model split-step Fourier method (SSFM) [10] solves the nonlinear Schrödinger equation numerically with high accuracy but has

high computational requirements, which limits its utilization for real-time operations and in dynamic optical network scenarios.

ML, the SSFM, and other methods can help in the control and management of optical networks and are suitable tools to solve dedicated, isolated network problems. However, they fail to provide a holistic representation of the network with high accuracy and, simultaneously, low computational complexity. In this context, the use of a digital twin has been proposed to improve lightpath provisioning, QoT estimation, fault management, etc. [11]. Another initiative is the open-source project GNPy [12], which implements the Gaussian noise model [13] to estimate the signal's linear interference (LI) noise and NLI noise powers. GNPy provides independent models for fiber propagation, optical amplifiers, and ROADMs, which can be concatenated to estimate the QoT of an optical connection. The use of GNPy as a digital twin of the optical network was suggested by the authors of [14], where the output of the models was analyzed and used for soft-failure prediction, localization, and identification.

In fact, the possibility to have independent models enables the development of specific models for devices with different characteristics, and it is key for the application of GNPy in open disaggregated network scenarios [15]. Independent modeling also allows the characterization of the optical signal not only at the receiver, but also at any intermediate location. In addition, different models can work together to represent the coexistence of different optical fiber types, optical channels

working on nonflat optical power profiles and heterogeneous data traffic, based on different modulation formats, symbol rates, channel spacing, etc. Note that such heterogeneity originates different signal-to-noise interactions along the lightpaths. Not only the LI noise contribution, but also the signal-to-nonlinearity noise interactions play an important role in current and future optical networks [16].

However, in addition to estimating the QoT, a digital twin should generate *expected signals* that can be compared with those obtained from the network [17,18]. For example, effects of the LI noise on the signal can be observed in the time domain as an increased dispersion of the received symbols. In that way, deviations between the observed and the expected signals can be detected and used for, e.g., soft-failure detection. In our previous paper [19], we proposed DNNs to model the noise contribution of ROADM and optical links in the time domain, i.e., to estimate the in-phase and quadrature (IQ) optical constellations. As in the case of GNPy, such specific models can be concatenated to represent lightpaths. In addition to the benefits discussed above regarding specific models, in the case of ML, this allows one to have a database of pre-trained models for elements with different characteristics like fiber length and type and number of spans in the case of optical links. Then, creating a digital twin for a lightpath consists of concatenating models for the specific elements in the route of the lightpath. In this paper, we extend [19] with models for the NLI noise, whose effects on the signal can be observed in the time domain and are different from those introduced by the LI noise. Thus, in this work, we consider LI noise from optical amplifiers, NLI noise from fiber Kerr effects, and optical filtering impairments from the wavelength selective switches (WSSs) inside ROADM. We call such deep-learning-based digital twins for the optical time domain OCATA. We evaluate OCATA for disaggregated and mixed disaggregated proprietary network scenarios.

The rest of the paper is organized as follows. Section 2 motivates the use of an optical digital twin and presents the considered network scenarios. Section 3 introduces the features defined to extract useful characteristics from IQ optical constellations and explains how we develop, train, and concatenate feedforward DNN-based models for IQ optical constellations. Section 4 explains the proposed models for optical constellation analysis focusing on lightpath modeling under the considered optical network scenarios. Illustrative numerical results are shown and discussed in Section 5, from the generation of the datasets to the OCATA performance. Finally, Section 6 draws the main conclusions of this work.

## 2. OPTICAL TIME DOMAIN DIGITAL TWIN AND APPLICATION SCENARIOS

Developing an accurate and low-complex optical transmission and networking time domain digital twin can lead to significant improvements in network management, from lightpath provisioning to network maintenance, which would have significant impact on operational costs.

Throughout this work, we propose an optical time-domain digital twin named OCATA and investigate two different optical network scenarios: (1) disaggregated optical networks

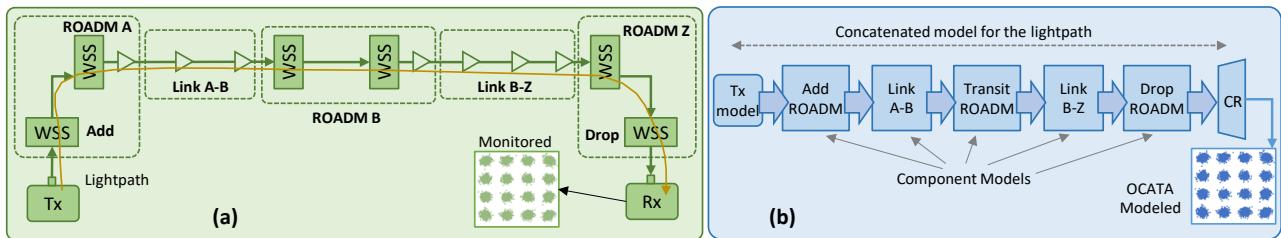
[15], with transponders, ROADM, and optical amplifiers from multiple vendors [Fig. 1(a)], and (2) mixed disaggregated proprietary networks, with single-vendor proprietary islands providing transparent optical connectivity to disaggregated segments. The example in Fig. 2(a) represents two disaggregated segments (S1 and S3) connected through a vendor-proprietary one (S2).

In the disaggregated scenario, information regarding the network topology, the type of fibers, etc., as well as the configuration and monitoring data from every optical component is accessible. In this scenario, a lightpath from site A to site Z can be modeled by concatenating models for the different components supporting such a lightpath, i.e., transponders, ROADM, and optical links [19]; see the example in Fig. 1(b), where output IQ optical constellation features of one component model are the input features of the following one. Every component model modifies the input features according to the noise that the specific physical network component introduces. Specifically, a transmitter (Tx) model generates the initial constellation features following a Tx configuration. Then, models for ROADM and optical links are concatenated in the same order in which the respective network components appear in the route of the lightpath.

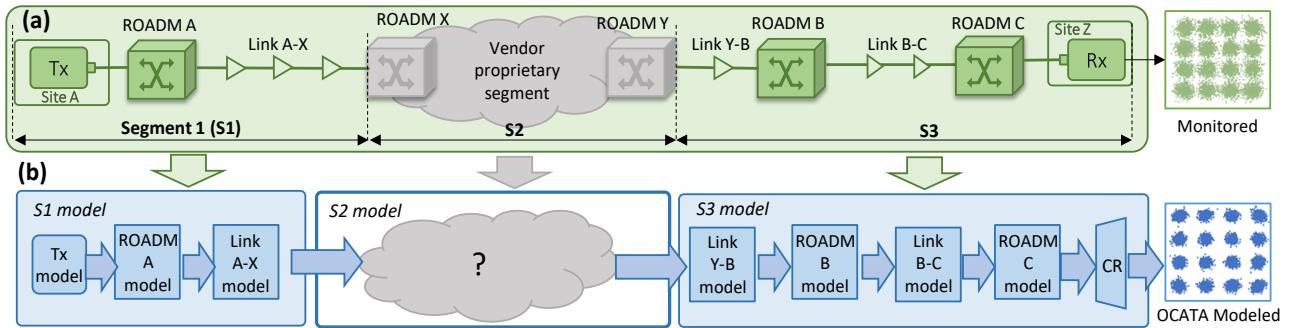
To minimize complexity and ensure component model availability at lightpath *provisioning time*, such models are trained beforehand using datasets collected from the network and/or coming from simulation. Then, at provisioning time, the specific concatenated model for the lightpath is created by selecting trained component models for the network components in the route of the lightpath from a model database. Finally, to reduce complexity even more, only the features of a few selected constellation points are propagated. Consequently, a constellation reconstruction (CR) module generates the features of the nonpropagated constellation points based on the received features to complete the IQ optical constellation. If the models are accurate enough, the features of constellation samples collected from the optical transponder in Z would match the expected optical constellation features obtained with OCATA. As we consider both LI and NLI noise, extensions to the concatenated model proposed in [19] are needed.

In the mixed scenario, we assume that vendor-specific optical network management systems (NMSs) provide software interfaces for provisioning but hide relevant information of the underlying optical line systems, e.g., the route of the lightpath. Since no information about the ROADM and optical links supporting the lightpath in the proprietary domain are available, a concatenation model cannot be built in the same way as in the disaggregated scenario. A possible solution is to create an *ad hoc* DNN model for the whole vendor-proprietary segment, as suggested in Fig. 2(b) for segment S2, and use it to create a concatenated model for the end-to-end lightpath.

Different NMSs will provide different partial information about the segment, e.g., the distance from the input to the output port for the optical connection or the number of intermediate ROADM. A possibility would be to infer the model for segment S2 by analyzing the monitored IQ optical constellations and finding a model for S2 that results in similar observed and expected optical constellations.



**Fig. 1.** (a) Example of a lightpath in disaggregated scenarios and (b) the DNN-based concatenation model for lightpath abstraction.



**Fig. 2.** (a) Example of a lightpath in mixed disaggregated proprietary scenarios and (b) the DNN-based concatenation model with per-segment analysis.

The next sections introduce the lightpath IQ optical constellation modeling, including both LI and NLI noise, OCATA, and some use cases for the defined network scenarios.

### 3. DEEP-LEARNING-BASED CONSTELLATION MODELING

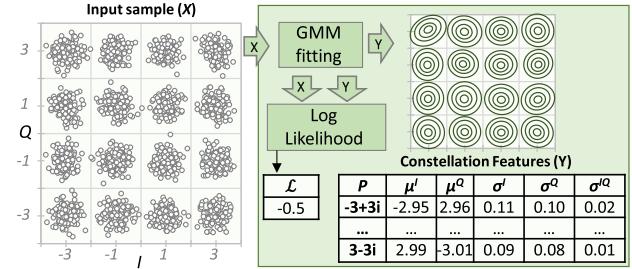
In this section, we first elaborate on the IQ optical constellation feature extraction. Recall that features are propagated along the concatenated DNN models representing the lightpath. Since different network components impair the optical signal traveling along the lightpath differently, specific DNNs are used to model links and ROADMs and LI and NLI noise. Table 1 summarizes the notation used in the rest of the paper.

#### A. IQ Optical Constellation Feature Engineering

Let us first define the IQ optical constellation features used in this paper. To that end, we start from an optical constellation sample  $X$  collected from the network [Fig. 1(a)]. A constellation sample is defined by a sequence of complex symbols,

**Table 1. Notation**

$X$	IQ optical constellation from an $m$ QAM signal
$M$	Set of constellation points in sample $X$ ( $m =  M $ )
$Y$	Set of constellation features for sample $X$ ( $\{Y^i, \forall i \in M\}$ )
$Y_{LI}$	Set of constellation features related to the LI noise
$\Delta Y_{NLI}$	Set of residuals features related to the NLI noise
$Y^i$	Vector of features for constellation point $i$ ( $Y^i = [y^k]^i$ )
$y^k$	$k$ th feature in vector $Y^i$ , where $y^k \in \mu^I, \mu^Q, \sigma^I, \sigma^Q, \sigma^{IQ}$
$\mu^I, \mu^Q$	Real and imaginary mean
$\sigma^I, \sigma^Q$	Real and imaginary variance
$\sigma^{IQ}$	Symmetric covariance



**Fig. 3.** GMM fitting for optical IQ constellation feature extraction.

where the real and imaginary parts represent the I and Q components of the optical signal, respectively. Such a sequence is commonly represented as an IQ optical constellation, where every symbol belongs to one of the constellation points  $M$ ,  $m$  being the number of constellation points, i.e.,  $m = |M|$ .

Figure 3, left, depicts an input sample  $X$  from an  $m = 16$  quadrature amplitude-modulated (QAM) optical signal. As in our previous work [19], we apply Gaussian mixture models (GMMs) [20] to characterize a given optical constellation sample as a set of  $M$  bivariate Gaussian distributions, i.e., one distribution for each constellation point. For instance, in the case of 16QAM signals, we obtain 16 different distributions.

Figure 3, right, represents the GMM fitting of the optical constellation sample  $X$ . The objective is to generate the set of semisupervised constellation features  $Y$  that summarizes  $X$  with a number of clear, unique, and predefined statistical characteristics. Specifically, GMM facilitates characterizing the actual centroid and dispersion of symbols around each centroid, which are relevant and significant characteristics that are affected by both LI and NLI noise. Let us denote  $Y^i$

the vector of features characterizing the constellation point  $i$ , i.e.,  $Y^i = [y^k]^i$ , where  $y^k$  represents the  $k$ th feature. Then,  $Y$  is formally defined as the vector that specifies all constellation point features, i.e.,  $Y = [Y^i, \forall i \in M]$ . In particular, GMM fitting is used to characterize every constellation point  $i$  by means of a two-component vector  $[\mu^I, \mu^Q]$  representing the mean position in the constellation and a three-component vector  $[\sigma^I, \sigma^Q, \sigma^{IQ}]$  that captures the real and imaginary variance and symmetric covariance terms that the symbols belonging to the constellation point  $i$  experience around the mean. Therefore, each constellation point  $i$  is characterized by five features, i.e.,  $Y^i = [y^k]^i = [\mu^I, \mu^Q, \sigma^I, \sigma^Q, \sigma^{IQ}]^i$ . For an  $m$ QAM signal, the GMM fitting is initialized to find  $m$  different bivariate Gaussian distributions and mean values, i.e., one for each of the expected constellation centroids, e.g.,  $(-3, 3)$ .

Note that in the case of highly impaired optical signals, symbols would be very dispersed, which makes constellation characterization more challenging. In addition, by forcing the constellation points to be modeled as bivariate Gaussian distributions and initializing the GMM fitting with the expected constellation centroids, the obtained features are strongly conditioned. For this reason, we compute the log-likelihood value  $L$  as a goodness-of-fit metric to estimate GMM fitting accuracy [20].

## B. Linear Links and ROADMs Models

As in [19], the DNN-based models target to propagate the LI features  $Y_{LI}$ . Hence, to develop the DNN-based models, we use the features obtained after GMM fitting considering the LI noise only. Note that the propagation of features under LI noise introduces nonlinear changes in the features.

To reduce the complexity of the DNNs while maintaining good performance, we limit the constellation points that are propagated to a selected subset. In particular, for 16QAM signals, we selected 4 out of 16 constellation points, which greatly reduces the number of input neurons. Specifically, two inner and two outer constellation points are selected to obtain knowledge from higher and lower power constellation points. Consequently, each DNN for 16QAM signals receives four constellation points  $\times$  5 features = 20 inputs. Different DNNs need to be trained for different optical link configurations, i.e., length and number of spans. In addition, we consider two models for the ROADMs: one for the add and transit ROADMs that include a booster optical amplifier (OA) at the output and a second one for the drop ROADM [see Fig. 1(a)].

After each individual DNN is trained, the main approach is to concatenate the desired models to emulate the desired lightpath. For example, we model the lightpath in Fig. 1 by concatenating three DNN models: add ROADM A, link A-B, and drop ROADM B. It is worth noting that we can extract the propagated Gaussian features at the output of each model for optical constellation reconstruction purposes (see details in Subsection 3.D).

## C. Nonlinear Residuals Models

For the sake of clarity, let us assume that we can obtain the features ( $Y$ ) of an IQ constellation sample  $X$  that includes LI and NLI noise as defined in the previous subsection. Let us assume also that we can predict the features related to the LI noise, i.e., without nonlinear impairments,  $Y_{LI\_pred}$ , by using the linear models defined in the previous subsection. Then, we compute what we call NLI residual features,  $\Delta Y_{NLI}$ , as

$$\Delta Y_{NLI} = Y - Y_{LI\_pred}. \quad (1)$$

In line with the linear models, DNNs are used to model NLI residual features. We assume that ROADMs do not add any NLI noise. In addition, NLI noise depends on the link lengths, and therefore, DNNs, modeling NLI noise in optical fibers, require the distance as additional input. However, to follow the concatenation model, we need models for both ROADMs and optical links, and hence (i) ROADM models just copy the inputs to the outputs, and (ii) DNNs for the optical links require two additional inputs for the accumulated distance until the link, denoted as  $d$ , and for the link length, denoted as  $l$ . Figure 4 represents the concatenated model for a lightpath considering both NLI and LI noise; for simplicity, we only represent a Tx, a ROADM, and a link.

After obtaining both the NLI residuals and the linear features, the predicted features  $Y_{pred}$  for the signal are

$$Y_{pred} = Y_{LI\_pred} + \Delta Y_{NLI\_pred}. \quad (2)$$

## D. Constellation Reconstruction

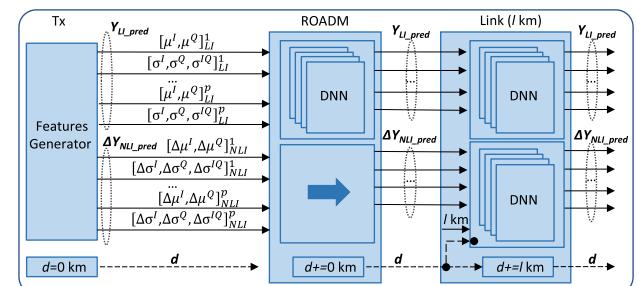
As previously stated, to reduce the size of the DNNs, we select and propagate a selected subset of constellation points  $C \subseteq M$  from a given IQ optical constellation sample  $X$  of an  $m$ QAM signal. CR is the process that predicts the features of every constellation point as a function of the features of the propagated ones; i.e.,

$$Y^i = F^i([Y^j, \forall j \in C]) \quad \forall i \in M. \quad (3)$$

Note that  $F^i(\cdot)$  for constellation points  $i \in C$  is defined as

$$F^i([Y^j, \forall j \in C]) = Y^i \quad \forall i \in C. \quad (4)$$

Recall that  $Y^i = [\mu^I, \mu^Q, \sigma^I, \sigma^Q, \sigma^{IQ}]^i = [y^k]^i$ . In [19], we proposed finding functions  $F^i(\cdot) = [f^k(\cdot)]^i$  as linear combinations of the propagated LI features. Consequently, we need to train  $(|M| - |C|) \times 5$  linear models. For instance, for 16QAM, this method requires training  $(16 - 4) \times 5 =$



**Fig. 4.** Details on LI and NLI feature propagation.

$60 f^k(\cdot)$  functions. Moreover, we cannot guarantee that linear models will provide accurate results when NLI noise is considered. For these very reasons, in this work, we propose a different approach based on one single DNN that takes as input propagated features  $[Y^j, \forall j \in C]$  and outputs nonpropagated features  $[Y^i, \forall i \in M \setminus C]$ . For instance, for 16QAM, the DNN has 20 inputs and 60 outputs.

#### 4. DIGITAL TWIN FOR THE OPTICAL TIME DOMAIN AND CONSTELLATION ANALYSIS

This section presents the algorithms required to create a time-domain digital twin of a given lightpath considering the two different optical network scenarios described in Section 2. Some examples of constellation analysis that take advantage of OCATA are presented afterwards.

##### A. OCATA for Disaggregated Optical Networks

Algorithm 1 presents the general algorithm of OCATA for a given lightpath. OCATA receives (i) the complete or a partial route of the lightpath (*Route*); it includes the needed information for the Tx, the ROADM, and the optical links and returns the IQ constellation  $X'$  that we could expect at the last element in the route; (ii) the reference to a database (DB) that contains objects with already-trained DNN models for the different elements in the route of a lightpath (*ModelDB*); and (iii) the length of the pseudo-random binary sequence (PRBS) to be generated for the initial optical constellation (*n\_bits*). Individual models are trained for every element and with specific characteristics: (i) Txs with different configurations of modulation format, symbol rate, power, etc.; (ii) optical links with different lengths and spans; and (iii) add-transit and drop ROADM.

The algorithm assumes that the first element in the route is a Tx. Therefore, the description of the Tx is used to identify the model in the DB that fits the characteristics of the Tx detailed in the configuration. Such a model is then used to generate the initial features to be propagated (lines 1–3 in Algorithm 1). Then, the distance of lightpath ( $d$ ) and the length of the last element ( $l$ ) are initialized (line 4). Next, for each element in the route (lines 5–15),  $d$  is updated with the length added by the last element and  $l$  is reset (line 6); it will be updated only in the case of traversing an optical link (line 11). Then, the specific model for the current element in the route is retrieved from the DB, i.e., an add-transit or drop ROADM, or an optical link model with the defined configuration in terms of length and number of spans (lines 7–13). Once a model has been found, the LI and NLI features are propagated separately from the current ones (lines 14–15). Note that the propagation of the NLI residuals features depends on the fiber length  $l$  of the element and accumulated distance  $d$ . In the case of a ROADM, the NLI residual feature propagation consists of just forwarding the input values without modification.

After propagating the features through all elements in the route, the CR model is retrieved from the DB and used to predict the complete set of constellation point features  $Y'$  from the propagated ones  $Y_{\text{pred}}$ . Next, features  $Y'$  are sampled and the IQ constellation  $X'$  is obtained (lines 16–18). Then, a

---

##### Algorithm 1. OCATA Main Algorithm

---

```

INPUT: Route, ModelDB, n_bits
OUTPUT:  $X'$ ,  $Y'$ ,  $[\mathcal{L}_{\min}, \mathcal{L}_{\max}]$ 
1:  $e \leftarrow \text{Route.pop}()$ 
2:  $Txmodel \leftarrow \text{ModelDB("Tx," } e.\text{config})$ 
3:  $Y_{\text{LI}}, \Delta Y_{\text{NLI}} \leftarrow \text{featuresGen}(Txmodel, e.\text{config}, n\_bits)$ 
4:  $d, l \leftarrow 0$ 
5: for each  $e$  in Route do
6:    $d \leftarrow d + l; l \leftarrow 0$ 
7:   if  $e.\text{type} = \text{"add-transit"}$  then
8:      $model \leftarrow \text{ModelDB("ROADM," "add-transit")}$ 
9:   else if  $e.\text{type} = \text{"link"}$  then
10:     $model \leftarrow \text{ModelDB("link," } e.\text{config})$ 
11:     $l \leftarrow e.\text{config.length}$ 
12:   else
13:      $model \leftarrow \text{ModelDB("ROADM," "drop")}$ 
14:      $Y_{\text{LI}} \leftarrow model.\text{propagateLI}(Y_{\text{LI}})$ 
15:      $\Delta Y_{\text{NLI}} \leftarrow model.\text{propagateNLI}(\Delta Y_{\text{NLI}}, d, l)$ 
16:    $CRmodel \leftarrow \text{ModelDB("CR," null)}$ 
17:    $Y' \leftarrow CRmodel(Y_{\text{pred}} = Y_{\text{LI}} + \Delta Y_{\text{NLI}})$ 
18:    $X' \leftarrow \text{sampling}(Y')$ 
19:    $LLHmodel \leftarrow \text{ModelDB("Log-Likelihood," } route)$ 
20: return  $X', Y', LLHmodel.\text{getRange}()$ 

```

---

##### Algorithm 2. Feature Generator

---

```

INPUT: Txmodel, Txconfig, n_bits
OUTPUT:  $Y_{\text{LI}}, \Delta Y_{\text{NLI}}$ 
1:  $bitSequence \leftarrow \text{PRBS}(n\_bits)$ 
2:  $X \leftarrow Txmodel.\text{generateConst}(bitSequence, Txconfig)$ 
3:  $Y_{\text{LI}} \leftarrow \text{GMMfitting}(X)$ 
4:  $\Delta Y_{\text{NLI}} \leftarrow [0, \dots, 0]$ 
5: return  $\text{selectFeatures}(Y_{\text{LI}}, \Delta Y_{\text{NLI}})$ 

```

---

model of feasible log-likelihood values for lightpaths matching the characteristics of the given one is retrieved from the DB (line 19). Finally, the IQ constellation, its features, and the range of log-likelihood values for the lightpath are returned (line 20).

Algorithm 2 presents the generation of the initial LI and NLI residual features that will be propagated afterwards. The algorithm receives as input the Tx model, the required Tx configuration, and the length of the PRBS to be generated. After generating the bit sequence (line 1 in Algorithm 2), the initial IQ optical constellation  $X$  is obtained by sampling, shaping, and modulating the bit sequence [21] following the received Tx configuration (line 2). Next, GMM fitting is performed to compute the set of bivariate Gaussian distribution that better fits the generated optical constellation  $X$  (line 3). Since such features come from an IQ optical constellation without NLI noise, they are stored in  $Y_{\text{LI}}$ , whereas the NLI residual features are initialized to 0 (line 4). Finally, subset  $C$  of constellation point features to be propagated is selected and returned (line 5).

##### B. Tuning OCATA with Received Samples

In this subsection, we present a methodology that allows us to identify an unknown network configuration by tuning

OCATA to force the generated IQ optical constellations to be as close as possible to the received ones. One possible application of this methodology is in mixed disaggregated proprietary optical networks, where the configuration of the proprietary segment is not completely known.

Let us take the example of mixed disaggregated proprietary optical networks in Fig. 2, where the only known characteristic of segment S2 is the distance. Conversely, the number of optical links (hops) and their main characteristics (number of spans, fiber characteristics, etc.) are unknown. Our objective is to find a configuration of segment S2 so that when OCATA generates IQ constellations with such a configuration, the results in the IQ optical constellations are similar to the measured ones.

We compare two IQ optical constellations  $X_1$  and  $X_2$  by computing the Euclidean distance of the difference of their constellation point features  $Y_1$  and  $Y_2$ , i.e.,

$$\text{diff}(X_1, X_2) = \|Y_1 - Y_2\|_2. \quad (5)$$

To find the configuration of segment S2, we can train DNN models that use features  $Y$  as input and predict the number of hops of the lightpath and the coefficient gamma. Hence, such DNNs have  $|M| \times 5$  inputs (e.g., for 16QAM, the DNNs have 80 inputs) and one single output.

Algorithm 3 presents the procedure that finds the most likely proprietary segment configuration. The algorithm receives a sampled constellation  $X$ , the route, an initial proprietary segment with two ROADM, and one single optical link with the total length of the proprietary segment, the models' DB, and a list with realistic values for the nonlinear gamma coefficient. First, the features  $Y$  of the received sample are extracted using a GMM fitting (line 1) and the log-likelihood value is computed to estimate the GMM fitting accuracy (line 2). The predicted number of hops of the lightpath and end-to-end nonlinear gamma coefficient are computed using DNN models from the models' DB (lines 3–6). The gamma coefficient that better approximates that of the complete lightpath is selected for all hops of the segment. The proprietary segment in the *Route* is then expanded with the predicted number of hops of the segment and the value of the gamma coefficient (lines 7–9). OCATA is then used to generate the estimated constellation  $X'$  and its features  $Y'$  using a PRBS with the same length as and the difference between the constellation sample, and the one generated with OCATA is computed (line 12). Since some of the optical links in the proprietary segment may have different optical fiber types, and thus different gamma coefficients, a search is started to find the configuration that minimizes the difference between the sampled and generated IQ constellations (lines 13–20). The log-likelihood computed for the constellation sample is verified to be in the range of log-likelihood values for lightpaths with the same characteristics as those found; otherwise, an error is generated (lines 21–22). Finally, the found configuration for the proprietary segment is returned (line 23).

### C. Constellation Analysis

The opportunities for the analysis of optical constellations that OCATA opens are broad. For instance, similar to the

---

### Algorithm 3. Find Proprietary Segment Model

---

**INPUT:**  $X$ , *Route*, *propSegment*, *ModelDB*, *GammaList*  
**OUTPUT:** *propSegment*

```

1:  $Y \leftarrow \text{GMMfitting}(X)$ 
2:  $\mathcal{L} \leftarrow \text{logLikelihood}(X, Y)$ 
3:  $\text{LPNHopmodel} \leftarrow \text{ModelDB}(\text{"LPNHops,"} \text{Route})$ 
4:  $nHopsSample \leftarrow \text{LPNHopmodel}(Y)$ 
5:  $\text{LPGammamodel} \leftarrow \text{ModelDB}(\text{"LPGamma,"} \text{Route})$ 
6:  $\text{LPgammaSample} \leftarrow \text{LPGammamodel}(Y)$ 
7:  $\text{gammaSegment} \leftarrow \text{get}(\text{GammaList}, \text{LPgammaSample})$ 
8:  $nHopsSegment \leftarrow nHopsSample \cdot \text{Route}.nHops$ 
9:  $\text{propSegment} \leftarrow \text{expandSegment}(\text{propSegment}, nHopsSegment,$ 
    $\text{gammaSegment})$ 
10:  $n\_bits \leftarrow \text{computeBits}(X)$ 
11:  $X', Y', L_{\min}, L_{\max} \leftarrow \text{OCATA}(\text{Route}, \text{ModelDB}, n\_bits)$ 
12:  $\text{diffX} \leftarrow \text{diff}(X, X')$ 
13: while true do
14:    $\text{gammaSegment} \leftarrow \text{getGammaConfig}(\text{GammaList},$ 
       $\text{propSegment}, \text{LPgammaSample})$ 
15:    $\text{newPropSegment} \leftarrow \text{expandSegment}(\text{propSegment},$ 
       $nHopsSegment, \text{gammaSegment})$ 
16:    $X', Y', \mathcal{L}_{\min}, \mathcal{L}_{\max} \leftarrow \text{OCATA}(\text{Route}, \text{ModelDB}, n\_bits)$ 
17:    $\text{newDiffX} \leftarrow \text{diff}(X, X')$ 
18:   if  $\text{newDiffX} > \text{diffX}$  then break
19:    $\text{diffX} \leftarrow \text{newDiffX}$ 
20:    $\text{propSegment} \leftarrow \text{newPropSegment}$ 
21: if  $\mathcal{L}$  not in  $[\mathcal{L}_{\min}, \mathcal{L}_{\max}]$  then
22:   return ERROR("log-likelihood value out of range")
23: return propSegment

```

---

### Algorithm 4. Constellation Analysis

---

**INPUT:**  $X$ , *Route*, *ModelDB*  
**OUTPUT:** *results*, *diffX*

```

1:  $lpModels \leftarrow [\text{"LPLength,"} \text{"LPNHops,"} \text{"LPPower,"}$ 
    $\text{"LPGamma"}]$ 
2:  $Y \leftarrow \text{GMMfitting}(X)$ 
3:  $\mathcal{L} \leftarrow \text{logLikelihood}(X, Y)$ 
4:  $n\_bits \leftarrow \text{computeBits}(X)$ 
5:  $X', Y', \mathcal{L}_{\min}, \mathcal{L}_{\max} \leftarrow \text{OCATA}(\text{Route}, \text{ModelDB}, n\_bits)$ 
6: if  $\mathcal{L}$  not in  $[\mathcal{L}_{\min}, \mathcal{L}_{\max}]$  then
7:   return ERROR("log-likelihood value out of range")
8:  $\text{results} \leftarrow \{\}$ 
9: for each lpModel in lpModels do
10:    $\text{model} \leftarrow \text{ModelDB}(\text{i}lpModel, \text{Route})$ 
11:    $\text{results.add}(\text{lpModel}, \langle \text{model}(Y), \text{model}(Y') \rangle)$ 
12: return results, diff( $X, X'$ )

```

---

ideas presented in the previous subsection, one could devise algorithms that compare a predicted lightpath's metrics from measured optical constellations to the ones predicted from constellations generated with OCATA, e.g., to detect and identify soft failures. In this case, DNN models to predict metrics can be extended to include the total length of the lightpath and the optical power, in addition to the number of hops and the gamma coefficient.

Algorithm 4 compares the lightpath's metrics predicted from the features of received and generated optical constellations and returns the results of the comparison. The algorithm receives a sampled constellation  $X$ , the route, and the models'

DB. Four metrics will be predicted, being the names of the DNN models as defined in a vector (line 1 in Algorithm 4). Next, the features  $Y$  of the received sample are extracted using GMM fitting (line 2) and the log-likelihood value is computed to estimate the GMM fitting accuracy (line 3). OCATA is then used to generate the estimated constellation  $X'$  and its features  $Y'$  using a PRBS of the appropriate length (lines 4–5). The log-likelihood is then computed for the constellation sample and, if it is outside the expected range, an error is generated (lines 6–7). Next, the lightpath's metrics are computed for the received constellation sample and for the constellation generated with OCATA using the specific DNNs trained to predict each metric (lines 8–11). The obtained comparison results for each metric are eventually returned together with the difference between the sampled and generated optical constellations (line 12).

## 5. ILLUSTRATIVE NUMERICAL RESULTS

In this section, we first introduce the simulation scenario and the IQ optical constellation synthetic data generation for the numerical evaluation of OCATA. Then, we focus on evaluating the key aspects involved in the optical constellation characterization and analysis. Finally, the performance of the OCATA methodology for both disaggregated and mixed disaggregated proprietary scenarios is provided.

### A. Optical Signal Sample Generation

To evaluate OCATA, we generated optical signal samples by means of a coherent digital optical system simulator developed in MATLAB, which implements the SSFM. Specifically, an optical system consisting of 11 wavelength-division multiplexing optical channels was considered where all transmit 16QAM at 64 GBd with a 75 GHz channel spacing. The optical power per channel was set either to  $-1$  dBm, close to the optimal one, or to  $2$  dBm, leading to a higher nonlinear regime. The fiber was modeled as standard single-mode fiber characterized by an attenuation factor of  $0.21$  dB/km, chromatic dispersion of  $16.8$  ps/(nm · km), and nonlinear coefficient gamma (denoted as  $\gamma$ ) of  $1.14$  ( $\text{W} \cdot \text{km}$ ) $^{-1}$ . In addition, we considered  $\gamma = 0$ , i.e., without NLI noise, and  $\gamma = 2$  ( $\text{W} \cdot \text{km}$ ) $^{-1}$ , i.e., high nonlinear coefficient. The OAs were modeled as erbium doped fiber amplifiers, with a noise figure of  $4.5$  dB and ideal gain, introducing amplified spontaneous emission noise to the optical signal. The WSSs' shape inside the ROADM [see Fig. 1(a)]

were modeled by averaging real spectral transfer function measurements, including filter losses and power ripples [22]. At the optical coherent receiver, digital signal processing able to perform ideal chromatic dispersion compensation and carrier phase recovery was considered.

Regarding links and lightpath configurations, we considered seven different types of optical links changing the total length, span length, and number of spans, i.e., (i)  $100$  km (2 spans of  $50$  km), (ii)  $200$  km ( $4 \times 50$  km), (iii)  $240$  km ( $4 \times 60$  km), (iv)  $300$  km ( $4 \times 75$  km), (v)  $400$  km ( $5 \times 80$  km), (vi)  $500$  km ( $5 \times 100$  km), and (vii)  $560$  km ( $7 \times 80$  km). Then, by configuring from two to five cascaded ROADMs, i.e., from one to four hops, optical signal samples for lightpaths with total length ranging from  $100$  to  $2240$  km were generated.

Following the optical system configuration described above, we defined a number of scenarios changing the link and lightpath configuration (note that different types of links were assumed in the same lightpath), coefficient  $\gamma$  [ $0$  or  $2$  ( $\text{W} \cdot \text{km}$ ) $^{-1}$ ], and optical power per channel ( $-1$  and  $2$  dBm). For every scenario,  $10$  signal samples were generated using  $2^{13}$  PRBSs and shaped by a root-raised cosine filter with a roll-off factor of  $0.06$ . The whole dataset, containing  $3520$  signal samples, is fully available in [23].

### B. Constellation Features and Log-Likelihood Analysis

Figure 5 shows an example of the evolution of the optical constellation features as a function of the total lightpath length for constellation point  $1$  ( $-3 + 3i$ ) and for the several transmission regimes considered, i.e., LI regime ( $\gamma$  of  $0$ ) and NLI regimes with coefficient  $\gamma$  of  $1.14$  or  $2$  ( $\text{W} \cdot \text{km}$ ) $^{-1}$ . We observe the strong correlation between the feature value and the lightpath length, as well as noticeable differences among transmission regimes. As expected, for the LI regime, the feature value range (defined as the difference between values at the shortest and longest lightpath) is smaller than that for the NLI ones. Figure 5 also highlights the importance of having accurate models not only for LI noise, but also for the NLI noise, therefore allowing one to clearly distinguish between different NLI regimes.

Additionally, Fig. 6 shows the evolution of the features as a function of the number of hops for four lightpath scenarios. Every curve is for a single lightpath scenario with the same total lightpath length and coefficient  $\gamma$ , with the lightpath consisting of two, three, or four hops. Table 2 summarizes the specific configuration of each lightpath. Although correlation

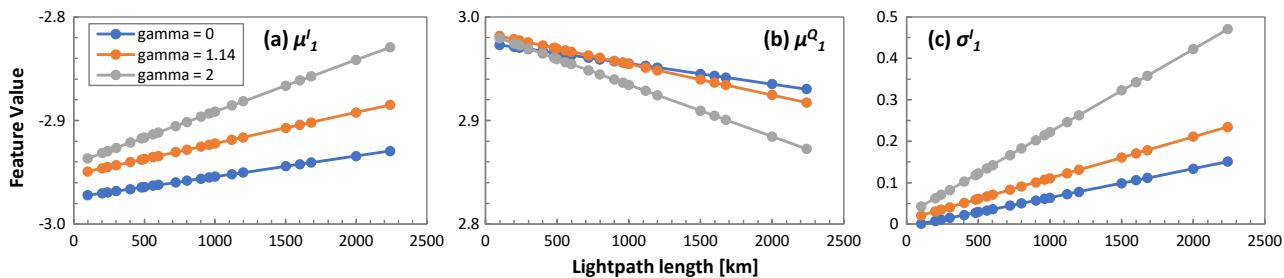
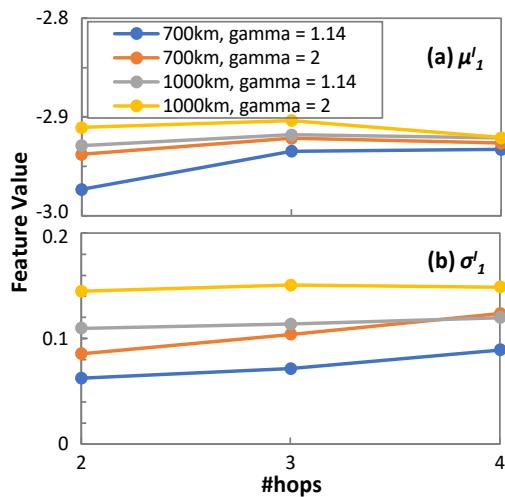


Fig. 5. Lightpath length impact on constellation features (a)  $\mu_1^I$ , (b)  $\mu_1^Q$ , and (c)  $\sigma_1^I$ .



**Fig. 6.** Effect of the number of hops on constellation features (a)  $\mu_1^I$  and (b)  $\sigma_1^I$ .

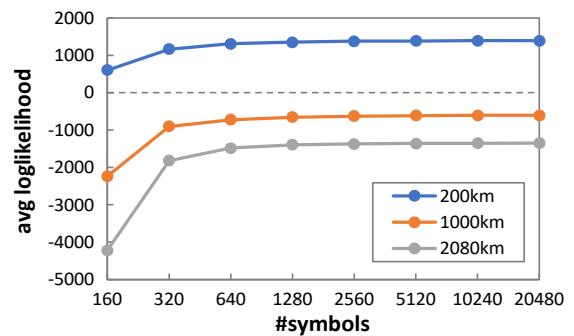
**Table 2. Links Lengths [km] for the Different Lightpath Scenarios**

Lightpath Length [km]	#Hops		
	2	3	4
700	400, 300	300, 200, 200	400, 100, 100, 100
1000	500, 500	400, 400, 200	400, 400, 100, 100

for hops is not as strong as for lightpath distance in Fig. 5, there are differences in the constellation features among the different scenarios. In consequence, we conclude that the selected bivariate Gaussian features provide an optical constellation characterization that makes it possible to distinguish among lightpaths with different lengths and NLI regimes.

Another important parameter for OCATA is the log-likelihood  $\mathcal{L}$  of the bivariate Gaussian distribution that characterizes a given optical signal sample. Figure 7 shows the evolution of  $\mathcal{L}$  as a function of the number of symbols, averaging all constellation points, for three different lightpath length scenarios: short (200 km), medium (1000 km), and long (2080 km). As expected,  $\mathcal{L}$  decreases with the lightpath length due to the loss of Gaussianity caused by NLI effects that distort the constellation points' shape. In other words, the expected  $\mathcal{L}$  clearly varies with the lightpath scenario, and therefore, it can be successfully used to perform constellation analysis validation like, e.g., that in lines 6–7 of Algorithm 4. We observe that  $\mathcal{L}$  reaches a stationary state after about 1000 symbols for all the considered scenarios. Note that the optical samples were generated with 2048 symbols each, which is well within the stationary region.

Finally, Fig. 8 evaluates the performance of the constellation comparison function  $diff$ , detailed in Eq. (5), as a function of several lightpath parameters, including total length, number of hops, and coefficient  $\gamma$ . Each curve belongs to a reference sample  $X_1$  that is compared against other samples  $X_2$  with the characteristics defined in the  $x$  axis. For instance, the blue curve in Fig. 8(a) presents the results of the comparison between a reference  $X_1$  sample of a 200 km lightpath and  $X_2$  samples of lightpaths with 200, 1000, and 2080 km. We



**Fig. 7.** Average log-likelihood as a function of the number of symbols.

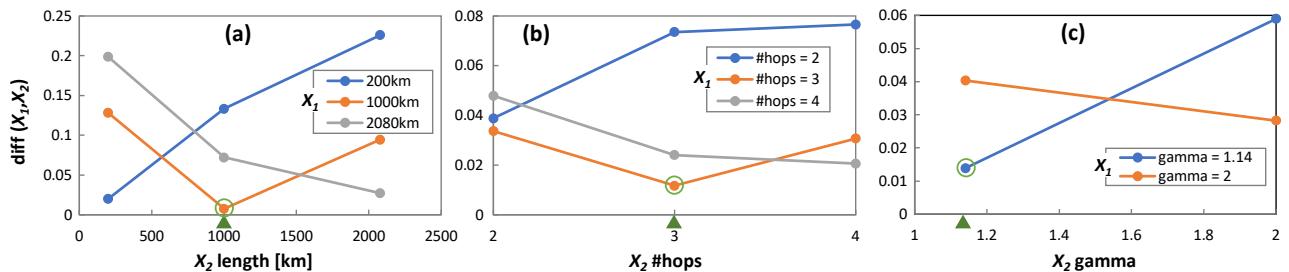
observe that the minimum  $diff$  value is achieved when  $X_1$  and  $X_2$  are from lightpaths with the same configuration. This fact is evident for all configurations of total length [Fig. 8(a)], number of hops [Fig. 8(b)], and coefficient  $\gamma$  [Fig. 8(c)]. These results highlight the usefulness of the  $diff$  function, which is essential when evaluating and finding proper lightpath configuration for proprietary segments in mixed disaggregated proprietary optical network scenarios.

### C. OCATA Performance Evaluation

To evaluate the performance of the OCATA algorithm detailed in Algorithm 1, let us concentrate on studying the accuracy of the models involved in the whole process, i.e., the DNN models in charge of characterizing optical components and the constellation reconstruction models. As detailed in Subsection 3.D, aiming at reducing DNN models' complexity, we selected a subset  $C$  of constellation points from the whole range of  $M$  points to be propagated. In particular, we selected two inner and two outer ones, i.e., constellation points 1 ( $-3 + 3i$ ), 7 ( $1 + 1i$ ), 10 ( $-1 - 1i$ ), and 15 ( $1 - 3i$ ).

Considering a disaggregated network scenario like the one represented in Fig. 1, we assume that full information about all network elements and their configuration is available and, consequently, we can concatenate pretrained DNN models to model a given lightpath with high accuracy. Table 3 shows the average relative error as a function of the model type for different optical components. The results are obtained by averaging all the predicted mean  $\mu$  and variance  $\sigma$  features of the propagated constellation points. As for  $\mu$ , the average relative error is small for all the DNN models considered (<2%). Regarding  $\sigma$ , errors are below 20% in all components, except for the ROADM add/transit, where the error increases. Although  $\sigma$  prediction errors might seem large, they are proportional to random variations in the training data (where  $\sigma$  is 10 times more variable than  $\mu$ ). Hence, we can validate this performance as sufficiently accurate for the use cases tackled in this paper.

After training the individual DNN-based models and performing concatenation and propagation of the selected constellation features, the reconstruction of the nonpropagated constellation points was carried out using the CR model. To show the performance of the CR model, Table 4 summarizes the average relative error as a function of the total lightpath length. We observe that errors are in line with those obtained



**Fig. 8.** Evaluation of optical constellation differences ( $\text{diff}$ ) as a function of the lightpath parameters.

**Table 3. Average Relative Error of Optical Component Models**

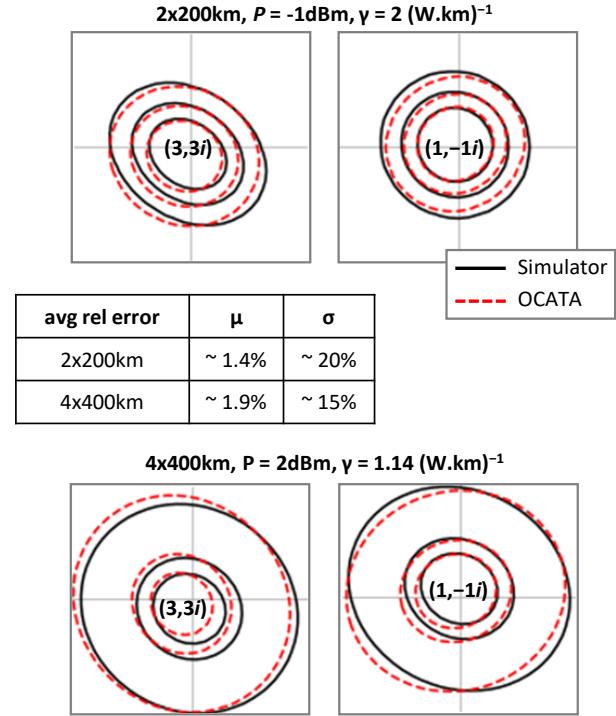
Model	Average Relative Error (%)			
	$\mu^I$	$\mu^Q$	$\sigma^I$	$\sigma^Q$
Optical link (100 km)	0.4	0.3	16.0	16.0
Optical link (200 km)	0.4	0.3	9.7	11.2
Optical link (400 km)	0.6	0.6	13.0	13.7
Optical link (560 km)	0.8	0.7	20.2	20.9
ROADM add/transit	1.2	1.1	32.4	33.2
ROADM drop	1.9	1.6	5.5	4.8

**Table 4. Average Relative Error of the CR Model**

Lightpath Length [km]	Average Relative Error (%)	
	$\mu$	$\sigma$
100	1	36
200	1	30
300	1	25
560	1	23
960	2	19
1200	2	16
2000	4	30

with the individual DNN models in Table 3. Therefore, we conclude that concatenation and reconstruction processes do not add significant incremental error in the received constellation, which is an essential result to validate the OCATA methodology.

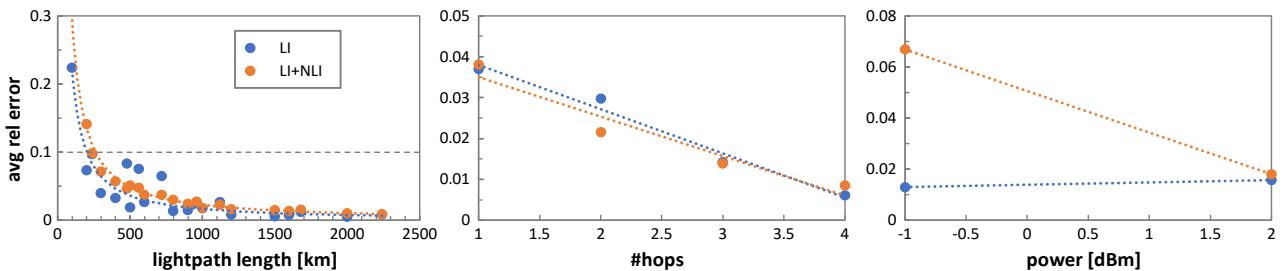
For illustrative purposes, Fig. 9 depicts the bivariate Gaussian distributions of constellation points 4 ( $3 + 3i$ ) and 11 ( $1 - 1i$ ) obtained with OCATA (red dashed lines) and by configuring the same lightpath in the MATLAB simulator detailed in Subsection 5.A (black solid lines). Note that these constellation points are not propagated by the optical component models; therefore, this figure allows one to visually evaluate the overall performance of both the propagation and reconstruction processes in the OCATA methodology. Two different lightpath scenarios were considered with different lightpath lengths, optical powers, and coefficient  $\gamma$ . Specifically, a lightpath of 400 km ( $2 \times 200$  km), power of  $-1$  dBm, and coefficient  $\gamma = 2$  ( $\text{W} \cdot \text{km})^{-1}$  is shown in the top of Fig. 9, whereas a lightpath of 1600 km ( $4 \times 400$  km), power of  $2$  dBm, and  $\gamma = 1.14$  ( $\text{W} \cdot \text{km})^{-1}$  is shown at the bottom. We observe that the shapes of the inner constellation point ( $1 - 1i$ ) are almost rounded, in contrast to the shapes of



**Fig. 9.** OCATA versus MATLAB-based simulator results.

the outer constellation point ( $3 + 3i$ ), which are more elliptical. Because the higher the power of the symbols, the higher the NLI noise, we can assume that the NLI noise is responsible for such elliptical impact, while the LI noise makes distributions more dispersed but circular, thus increasing  $\sigma^I$  and  $\sigma^Q$  almost equally. In any case, the similarity of Gaussian distributions validates the accuracy and usefulness of the OCATA methodology.

Let us now consider the mixed disaggregated proprietary optical network scenario illustrated in Fig. 2. We focus on the performance evaluation of the DNN-based models detailed in Algorithm 3, which are used to predict the lightpath characteristics from the optical constellation features. Toward this aim, we have trained different DNN models to predict the lightpath length, the number of hops, and the optical power of the lightpath as a function of the optical constellation features. All DNN models followed the same structure: 80 input neurons, i.e., 5 features  $\times$  16 constellations points; 3 hidden layers with 160, 80, and 20 neurons, respectively; and 1 single output neuron predicting the specific characteristic of that model. In



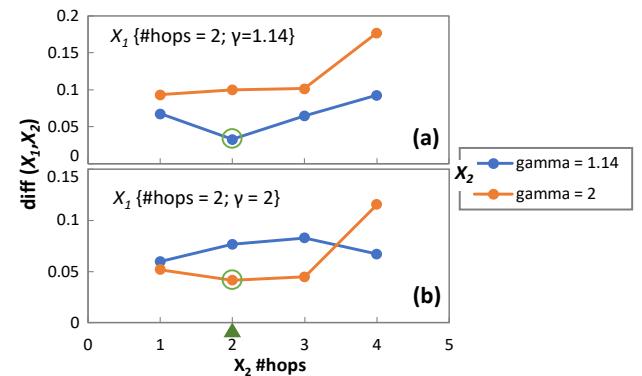
**Fig. 10.** Performance of lightpath metrics predictors.

addition, all models used (i) the hyperbolic tangent ( $\tanh$ ) as the activation function, (ii) root mean squared propagation as the optimization algorithm, (iii) a training stage with up to 5000 epochs and 500 epochs without improvement, and (iv) mean squared error as the loss function. Such a configuration resulted in up to 15 min of training time per model.

Figure 10 shows the performance of the several models considered in this work. Regarding the DNN model used to predict the total lightpath length [Fig. 10(a)], the average relative error is below 35% for all considered cases and below 10% for distances above 500 km. We tested the proposed models with the predicted constellation features only with the LI noise and considering both LI and NLI noise. Although differences can be observed, the NLI residual models do not introduce significant error for the predicted IQ optical constellation features. Concerning the DNN model used to predict the number of hops in the route of a lightpath [Fig. 10(b)], the model shows remarkable accuracy, providing average relative error below 4% for all the cases. Once again, both LI and LI + NLI residual features usually provide a similar performance. Finally, for the DNN model used to predict the optical power [Fig. 10(c)], the model produces average relative errors not exceeding 7% when considering only the LI noise and smaller than 2% when considering both LI and NLI noise.

Considering the quite good performance of the proposed DNN models used to characterize a lightpath given the optical constellation features, we conclude that a good enough characterization of a network is possible when crossing proprietary segments. To further illustrate this result, let us evaluate the performance of Algorithm 3 to find the most likely proprietary segment model. In Fig. 11, a reference 1300 km lightpath sample  $X_1$  crossing a proprietary segment with two hops of 300 and 200 km is considered. We observe that the minimum  $diff$  value is obtained exactly when the reference sample  $X_1$  is compared against one modeled sample  $X_2$  with similar configuration, i.e., two hops and coefficient  $\gamma = 1.14 (\text{W} \cdot \text{km})^{-1}$  [Fig. 11(a)] or  $\gamma = 2 (\text{W} \cdot \text{km})^{-1}$  [Fig. 11(b)].

To conclude, let us analyze the computation time of OCATA and compare it with SSFM. Table 5 shows the time needed to generate optical constellations (each with 2048 symbols) with both methods for lightpaths of different numbers of hops and link length. The experiments were performed on a computer with an Intel Core i5 CPU at 2.67 GHz with 8 GB RAM and running Windows 10 64-bit. The results show much better scalability of OCATA, which runs over 3 orders of magnitude faster than SSFM. In addition, the running



**Fig. 11.** Performance of the proprietary segment model algorithm.

**Table 5. Execution Time for Lightpaths of Different Characteristics**

Lightpath Characteristics			Execution Time [s]	
Link Length [km]	#hops	Lightpath Length [km]	SSFM	OCATA
100	1	100	4.5	$4.0 \cdot 10^{-3}$
100	2	200	6.1	$5.6 \cdot 10^{-3}$
100	3	300	8.6	$6.5 \cdot 10^{-3}$
100	4	400	10.9	$7.6 \cdot 10^{-3}$
400	1	400	10.3	$4.0 \cdot 10^{-3}$
400	2	800	18.1	$5.6 \cdot 10^{-3}$
400	3	1200	26.2	$6.5 \cdot 10^{-3}$
400	4	1600	34.7	$7.6 \cdot 10^{-3}$

time is only dependent on the number of hops and not on the distance, as in SSFM.

Finally, let us numerically evaluate the reduction of model size achieved by propagating a subset of constellation points. Quantitatively, a model (optical link or ROADM) with both LI and NLI DNNs propagating the selected four constellation points contains up to 960 nonzero coefficients (weights). Instead, when using the full set of 16 constellation points, the number of nonzero coefficients can rise up to 15,360. Hence, selecting 25% of the constellation points reduces the number of coefficients by 93.8%.

## 6. CONCLUDING REMARKS

A low complexity deep-learning-based digital twin for the optical time domain, named OCATA, has been presented. DNN models for IQ optical constellations include LI noise

resulting from optical amplification, optical filtering in the ROADM, and NLI noise due to nonlinear optical fiber propagation. OCATA is based on the concatenation of several pretrained DNN models used for specific network elements, i.e., ROADM and optical links. The proposed DNN models are fed by features characterizing the IQ optical constellations.

Two optical network scenarios have been analyzed for disaggregated and mixed disaggregated proprietary domains. For each network scenario, several lightpaths with different configurations in terms of span and link length, number of hops, optical power, and coefficient gamma, have been used to obtain datasets to train, test, and validate the proposed models. Exhaustive results show the accuracy of OCATA for optical constellation modeling, including constellation feature propagation and reconstruction in both LI and NLI optical transmission regimes. Additionally, DNN models for end-to-end lightpath modeling in such mixed disaggregated proprietary optical networks have been proposed. Such models can predict typical lightpath parameters, such as the number of hops, optical power, and coefficient gamma, from optical constellation features.

With these models, OCATA can be used as a tool for network automation and failure management, including anticipated degradation detection, identification, and localization.

**Funding.** Agencia Estatal de Investigación (PID2020-114135RB-I00 (IBON); Horizon 2020 Framework Programme (101016663 (B5G-OPEN)); H2020 Marie Skłodowska-Curie Actions (813144 (REAL-NET)); Institució Catalana de Recerca i Estudis Avançats.

## REFERENCES

1. V. López and L. Velasco, *Elastic Optical Networks: Architectures, Technologies, and Control*, Optical Networks Book Series (Springer, 2016).
2. L. Velasco, A. Chiadò Piat, O. González, A. Lord, A. Napoli, P. Layec, D. Rafique, A. D'Errico, D. King, M. Ruiz, F. Cugini, and R. Casellas, "Monitoring and data analytics for optical networking: benefits, architectures, and use cases," *IEEE Netw. Mag.* **33**(6), 100–108 (2019).
3. H. Hallingby, S. Fletcher, V. Frascolla, A. Gavras, I. Mesogiti, and F. Parzysz, "5G ecosystems," white paper (5G-IA, 2021), <https://doi.org/10.5281/zenodo.5094340>.
4. D. Rafique and L. Velasco, "Machine learning for optical network automation: overview, architecture and applications [Invited Tutorial]," *J. Opt. Commun. Netw.* **10**, D126–D143 (2018).
5. L. Velasco, B. Shariati, F. Boitier, P. Layec, and M. Ruiz, "A learning life-cycle to speed-up autonomic optical transmission and networking adoption," *J. Opt. Commun. Netw.* **11**, 226–237 (2019).
6. L. Velasco, S. Barzegar, F. Tabatabaeimehr, and M. Ruiz, "Intent-based networking for optical networks [Invited Tutorial]," *J. Opt. Commun. Netw.* **14**, A11–A22 (2022).
7. P. Freire, V. Neskoruk, A. Napoli, B. Spinnler, N. Costa, G. Khanna, E. Riccardi, J. E. Prilepsky, and S. Turitsyn, "Complex-valued neural network design for mitigation of signal distortions in optical links," *J. Lightwave Technol.* **39**, 1696–1705 (2021).
8. R. Morais and J. Pedro, "Machine learning models for estimating quality of transmission in DWDM networks," *J. Opt. Commun. Netw.* **10**, D84–D99 (2018).
9. R. Proietti, X. Chen, K. Zhang, G. Liu, M. Shamsabardeh, A. Castro, L. Velasco, Z. Zhu, and S. J. B. Yoo, "Experimental demonstration of machine learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," *J. Opt. Commun. Netw.* **11**, A1–A10 (2019).
10. J. Shao, X. Liang, and S. Kumar, "Comparison of split-step Fourier schemes for simulating fiber optic communication systems," *IEEE Photon. J.* **6**, 7200515 (2014).
11. D. Wang, Z. Zhang, M. Zhang, M. Fu, J. Li, S. Cai, C. Zhang, and X. Chen, "The role of digital twin in optical communication: fault managements, hardware configuration, and transmission simulation," *IEEE Commun. Mag.* **59**(1), 133–139 (2021).
12. V. Curri, "GNPy model of the physical layer for open and disaggregated optical networking [Invited]," *J. Opt. Commun. Netw.* **14**, C92–C104 (2022).
13. P. Poggolini, G. Bosco, A. Carena, V. Curri, Y. Jiang, and F. Forghieri, "The GN-model of fiber non-linear propagation and its applications," *J. Lightwave. Technol.* **32**, 694–721 (2014).
14. S. Barzegar, M. Ruiz, A. Sgambelluri, F. Cugini, A. Napoli, and L. Velasco, "Soft-failure detection, localization, identification, and severity prediction by estimating QoT model input parameters," *IEEE Trans. Netw. Serv. Manag.* **18**, 2627–2640 (2021).
15. L. Gifre, J.-L. Izquierdo-Zaragoza, M. Ruiz, and L. Velasco, "Autonomic disaggregated multilayer networking," *J. Opt. Commun. Netw.* **10**, 482–492 (2018).
16. A. Amari, O. Dobre, R. Venkatesan, O. Kumar, P. Ciblat, and Y. Jauén, "A survey on fiber nonlinearity compensation for 400 Gb/s and beyond optical communication systems," *IEEE Commun. Surv. Tutorials* **19**, 3097–3113 (2017).
17. B. Shariati, M. Ruiz, J. Comellas, and L. Velasco, "Learning from the optical spectrum: failure detection and identification [Invited]," *J. Lightwave. Technol.* **37**, 433–440 (2019).
18. A. P. Vela, B. Shariati, M. Ruiz, F. Cugini, A. Castro, H. Lu, R. Proietti, J. Comellas, P. Castoldi, S. J. B. Yoo, and L. Velasco, "Soft failure localization during commissioning testing and lightpath operation," *J. Opt. Commun. Netw.* **10**, A27–A36 (2018).
19. M. Ruiz, D. Sequeira, and L. Velasco, "Deep learning-based real-time analysis of lightpath constellations [Invited]," *J. Opt. Commun. Netw.* **14**, C70–C81 (2022).
20. N. Bouguila and W. Fao, *Mixture Models and Applications* (Springer, 2020).
21. J. Proakis and M. Salehi, *Digital Communications*, 5th ed. (McGraw-Hill Education, 2007).
22. T. Rahman, A. Napoli, D. Rafique, B. Spinnler, M. Kuschnerov, I. Lobato, B. Clouet, M. Bohn, C. Okonkwo, and H. Waardt, "On the mitigation of optical filtering penalties originating from ROADM cascade," *IEEE Photon. Technol. Lett.* **26**, 154–157 (2014).
23. M. Ruiz, L. Velasco, and D. Sequeira, "Optical Constellation Analysis (OCATA)," CORA Repòitori de Dades de Recerca: Version 2, 2022, <https://doi.org/10.34810/data146>.