Machine learning assisted abstraction of photonic integrated circuits in fully disaggregated transparent optical networks

Machine learning assisted abstraction of photonic integrated circuits in fully disaggregated transparent optical networks

Ihtesham Khan[1], Maryvonne Chalony[2], Enrico Ghillino[3], M Umar Masood[1], Jigesh Patel[3], Dwight Richards[4], Pablo Mena[3], Paolo Bardella[1], Andrea Carena[1], Vittorio Curri[1]

[1]*Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129, Torino, Italy*

[2] *Light Tec SARL, Pôle d'Activités Hyérois, 1128 Route de Toulon, 83400 Hyères, France*

[3] *Synopsys, Inc., 400 Executive Blvd Ste 101, Ossining, NY 10562, United States*

[4] *College of Staten Island, CUNY, 2800 Victory Blvd, Staten Island, NY 10314, United States*

e-mail: enrico.ghillino@synopsys.com

ABSTRACT Optical networks are fast evolving towards full disaggregation and softwarization down to layer-0: the data transport layer. Moreover, network elements are progressively exploiting photonic integrated circuits (PICs) to perform complex functions at the photonic level. Thanks to the advanced simulation tools, also the behavior of photonic integrated circuits can be abstracted and used within the SDN paradigm for network planning and management, permitting a full network disaggregation and softwarization down to below layer-0. To this aim, one of the main issues is the need for exact knowledge of the physical parameters of integrated circuits. In this work, we use machine learning techniques to deliver an augmented knowledge of the physical parameters of integrated circuits to be used for their full and accurate softwarization. We consider a performance prediction problems applied to a switching component. Overall results as well as data sets for machine-learning training are obtained by leveraging the integrated software environment of the Synopsys Photonic Design Suite.

Keywords: Machine learning; Photonic Integrated Circuits; Q-factor.

1. INTRODUCTION

A dramatic increase in the global IP traffic, urged by the introduction of 5G technology along with the expansion of bandwidth hungry applications such as 4K or Full High Definition (FHD) video and Virtual and Augmented Reality (VR and AR) contents, is forecast for the coming years [1]. To support this trend, current optical networks requires an upgrade toward a full orchestration including also the physical layer. This can only be achieved by the abstraction of all network elements and functionalities, starting from the component design up to the network management. Network elements are progressively exploiting photonic integrated circuits (PICs) to perform complex functions at the photonic level avoiding the bottleneck of opto-electronic conversion. As this trend raises the complexity of such devices [2], more sophisticated optimization and modeling tools are required. Thanks to advanced simulation tools, the behavior of PICs can be predicted with high accuracy and abstracted. This allows to consider it within the software-defined networking (SDN) paradigm for network planning and management. Using this approach, we achieve a full network disaggregation and softwarization down to below layer-0.

The design of PIC based on numerical simulation is a well established approach. Two main issues can affect the process: the exact knowledge of physical parameters and time consuming optimization that imposes limit to *run-time* applications. For example, an important approach in this area is the inverse design that allows to derive PIC parameters based on the required functionality [3], [4]. In recent years, Machine Learning (ML) approaches were also introduced to design a sub-wavelength focusing lens and an optical coupler [5]. Moreover, Deep Neural Networks (DNN) were utilized to predict the geometry of a nanophotonic structure for a desired response [6]. These recent studies indicate that the future of PICs will be massively dependent on automated intelligent design techniques [7].

Another important application of neural networks (NN) to the analysis of PICs is the prediction of the behaviour of a component, without the need to recur to long numerical simulations. In this case, for example, an NN substitutes the PIC model capturing the relationship between input control signals and output transfer functions. This a first step toward the abstraction of optical components to be used in SDN controller, but it would require a further processing of the NN output to evaluate the impact on the Quality of Transmission (QoT) of channels processed by the PICs. In this paper, we close the gap between the component design level and the evaluation of system performance applying machine learning techniques directly to predict the dependence of QoT on control signals of a PIC. Our approach can be used in system level design to understand the impact on performance of a given optical component, without the need to run lengthy numerical simulations. And in a longer term view it would be the required real-time abstraction at network level of an optical component.

In this work, taking advantage of the capability of the Synopsys multi-layer design environment [8],[9], we show an application of this ML based expansion of the abstraction paradigm below the transmission layer – layer-0 –, bringing the network abstraction to the component-design layer. We use ML techniques to deliver an augmented knowledge of the physical parameters of integrated circuits to be used for their full and accurate softwarization. This work is based on the component design presented in [10], for an optical Benes Switch design in Silicon Photonics using Analog Photonics (AP) Process Design Kit (PDK) component library elements [11].

2. Simulation Model Data Set Analysis

To demonstrate the potentiality of the Synopsys design tool and illustrate a possible workflow to develop optical components, we analyze the impact of an optical switch on the transmission performance of channels processed by it. For this purpose, a 4x4 multi-stage switch based on a Benes layout [12] is considered. Synopsys design and simulation environment: Optsim and OptoDesigner synergistically enable a vertical abstraction below layer-0, which allows to analyze the Benes structure, starting from the physical layout of the building blocks, shown in Fig.1a, up to the estimation of system level performance when employed to process PM-MQAM signal. System performance is measured based on BER, through an error counting approach, but we report them using a Q-factor. The Benes Switch can redirect any of the four wavelengths ($\lambda 1$, $\lambda 2$, $\lambda 3$, $\lambda 4$) at its inputs to any of the four outputs: this is achieved by varying the voltages of six internal signal, each one controlling a single ring resonator [12]. Considering simulation parameters as defined in [10], we have nominal voltages for cross and bar states of 0V and 8.4V, respectively. The 4x4 Benes Switch has a total of 2^6 combinations corresponding to switching states. In this study we consider that control signals may suffer a perturbation, in the range of $\pm$1V. Along with this perturbation, we consider two distinct working modes of Benes Switch from already mentioned 2^6 combinations. The first mode, which we refer to as combination A, corresponds to the output sequence $\lambda 1$, $\lambda 2$, $\lambda 3$, $\lambda 4$ at the corresponding Output-1, Output-2, Output-3 and Output-4 respectively, while the second mode, which we refer to as combination B, corresponds to the output sequence is $\lambda 1$, $\lambda 3$, $\lambda 2$, $\lambda 4$ at the corresponding Output-1, Output-2, Output-3 and Output-4 respectively.

To obtain a data set, the integrated software environment of the Synopsys Photonic Design Suite is used. We consider the two different switching states described above. Perturbing the nominal control voltage the transmitted channels incur in different filtering through the paths inside the 4x4 Benes Switch, varying the Q-factor. Each element in the data-set is composed as follows: for each of the six input voltages we have four output Q-factor values, one for each channel processed.

Building the data-set, we can also draw some basic considerations by computing the average of the Q-factor for each output port for all the realizations. In Fig. 1b we show results referring to the particular combination A, similar results have been obtained also for combination B. The average values of Q-factor (purple dots) is comprised between 2.68 and 2.75 dB, with standard deviation (purple error bar) of about 0.20 dB. Green line shows the minimum value for each output where 1.79 dB (red line) is the global minimum. The present Q-factor statistics quantify Q-factor predictions for a particular output. Considering a worst case scenario, without any further knowledge, so that the same Q-factor threshold must be enforced for all output ports with a magnitude lower than a global expected minimum (red line). In this approach, due to the fluctuations in Q-factor values it creates up to 1.25 dB of average margin in Q-factor. The major challenge in the present environment is to decrease the uncertainty in Q-factor prediction, in the absence of exact knowledge of driving conditions of Benes Switch.

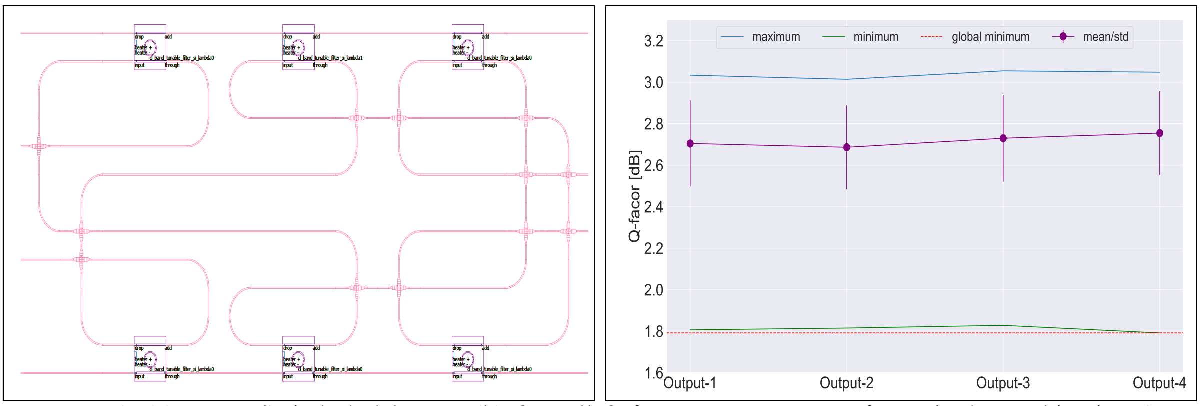

Figure 1: (a) Benes Switch 4x4 layout. (b) Overall Q-factor measurements for a single combination A.

3. MACHINE LEARNING

The proposed study presents the use of data-driven ML module to deliver an augmented knowledge of the physical parameters of photonic integrated circuit, considering both the inverse design and the performance prediction problem shown in Fig. 2 b. This special kind of abstraction of integrated circuit, particularly Benes Switch in the present study is used for its full and accurate softwarization. In this particular scenario, ML based learning method is used to reduce the uncertainty in the Q-factor description of each output port of Benes Switch. Like all other ML based learning methods, the proposed model training and prediction processes require the definition of the features and labels, which indicate the system inputs and outputs, respectively. The manipulated features include the voltage measurements that are delivered to the six input configuration ports, while the exploit label is Q-factor of the particular output port shown in Fig. 2a. The proposed ML module cognates the features and labels of the realization of each combination using the functionality of ML, specifically Deep Neural Network (DNN) [13]. The proposed DNN is developed by using higher level application program interfaces (APIs) of TensorFlow© platform that consists of 2 hidden layers along with 10 neurons for each hidden layer, having *ReLU* as activation function that allows translation of the given input features into the prediction of label of our point of interest with less complexity [14]. The proposed ML model is evaluated by *Mean Square Error (MSE)* as a loss function given by equation (1).

$$\mathrm{MSE} = \frac{\sum_{i=0}^{n} (\mathrm{Qfactor}_i^{\mathrm{p}} - \mathrm{Qfactor}_i^{\mathrm{a}})^2}{n}, \tag{1}$$

where $\mathrm{Qfactor}_i^{\mathrm{a}}$ and $\mathrm{Qfactor}_i^{\mathrm{p}}$ are the actual and predicted values of Q-factor of particular output port for the ith realization, respectively, and n is the total number of Q-factor realizations to be tested. The model is configured for training, validation and testing by the conventional rule 70/15/15 having training-steps of 10000, in order to give the model sufficient intelligence. The training set for each combination in the present scenario consists of 600 realizations, while the test set consists of 100 realization. In addition to this, the model is empowered by default *Adaptive Gradient Algorithm (ADAGRAD)* keras optimizer with default *learning rate* = 0.01 and L_1*-regularization* = 0.001 [15]. The weight and bias values are updated according to *ADAGRAD* optimization.

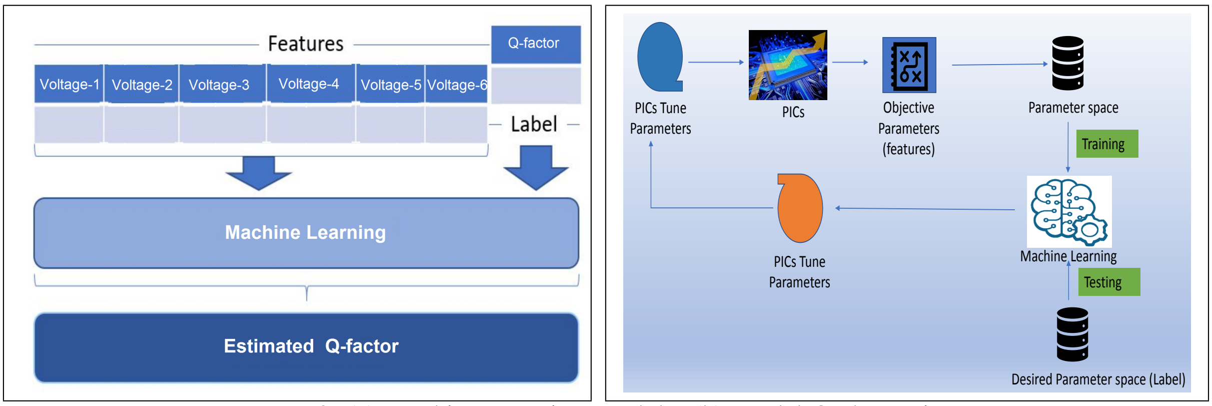

Figure 2: (a) Machine Learning Module; (b) Model Orchestration;

4. Results

In this section, we exploit the performance of the proposed ML module in-order to decrease the uncertainty in Q-factor margin for each output port of Benes Switch . The metric to quantify this ability of ML engine is defined by ΔQ-factor, where ΔQ-factor = Q-fator^{predicted} - Q-fator^{actual}.

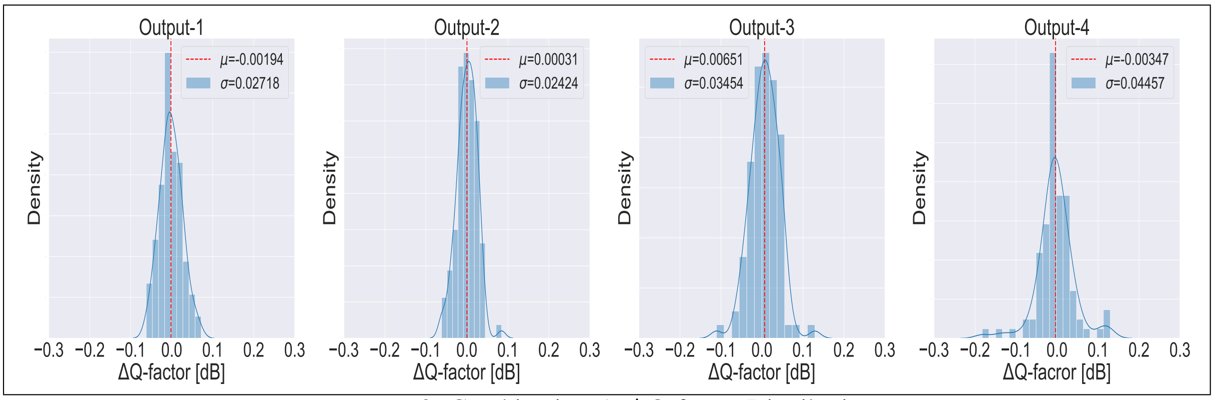

Figure 3: Combination A ΔQ-factor Distribution

The reliability of the module is verified by testing it on two distinct combination A and B. The distribution of the ΔQ-factor along with the mean (μ) and standard deviation (σ) statistics against each output port of the Benes Switch for both the combination A and B are shown in Fig. 3 and Fig. 4.

Analyzing the statistics of μ and σ, to be more confident and conservative we consider the margin twice the average σ value of ΔQ-factor of all output ports. Demonstrating the results related to combination A the uncertainty in the Q-factor margin is reduced to 0.06 dB, while for combination B it is reduced to 0.05 dB. The remarkable decrease in the margin, shows that ML played a promising role to yield the a high degree of information related to physical parameters of Benes Switch. This information can not only be utilized to accurately characterize the Benes Switch but also to provide its complete software abstraction.

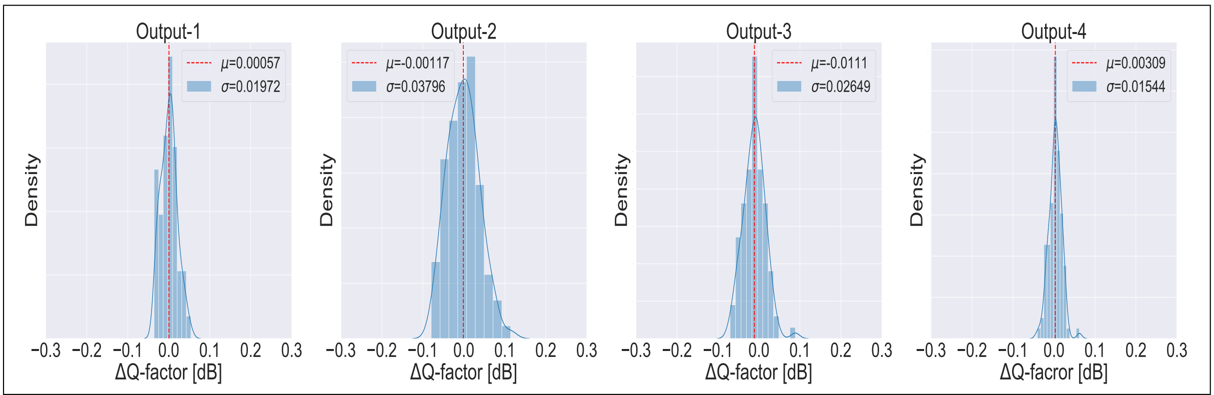

Figure 4: Combination B ΔQ-factor Distribution

5. Conclusion

In summary, we propose and exploit the ability of ML technique for abstraction of photonic integrated circuit. Analyzing a Benes Switch structure, we demonstrated the capability of predicting impact on system performance of component driving signals.

The proposed ML approach is used to provide an augmented knowledge of the impact of component parameters of the analyzed component to deploy its complete and authentic softwarization.

Exploiting the results of ML approach, the uncertainty in the description of Q-factor is dramatically decreased to negligible values: 0.06 dB for combination A and 0.05 dB for combination B.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," Tech. rep., Cisco (2017).
[2] N. Radhakrishnan et.al., "Large-scale photonic integrated circuits for long-haul transmission and switching," JON **6**, 102–111 (2007).
[3] A. Piggott et al., "Inverse design and demonstration of a compact and broadband on-chip wavelength demultiplexer," Nature Photonics **9**, 374–377 (2015).
[4] L. Su et al., "Inverse design and demonstration of a compact on-chip narrowband three-channel wavelength demultiplexer," ACS Photonics **5**, 301–305 (2018).
[5] T. Mirbek et.al, "Ultracompact photonic structure design for strong light," JLT **36**, 2812–2819 (2018).
[6] L. Dianjing et.al, "Training deep neural networks for the inverse design of nanophotonic structures," ACS Photonics **5**, 1365–1369 (2018).
[7] M. Sean et.al, "Inverse design in nanophotonics," Nature Photonics **12**, 659–670 (2018).
[8] `https://www.synopsys.com/photonic-solutions/pic-design-suite.html`.
[9] `https://www.synopsys.com/photonic-solutions.html`.
[10] E. Ghillino et.al, "Assessing the impact of design options for an optical switch in network routing impairments," in *2019 21st ICTON,* (IEEE, 2019), pp. 1–4.
[11] T. Erman et.al, "Apsuny process design kit (pdkv3.0): O, c and l band silicon photonics component libraries on 300mm wafers," in *2019 OFC,* (IEEE, 2019), pp. 1–3.
[12] L. Qi et.al, "Single microring-based 2x2 silicon photonic crossbar switches," IEEE Phot. Tech. Letters, **27**, 1981–1984 (2015).
[13] C. M. Bishop, *Pattern recognition and machine learning* (springer, 2006).
[14] A. Kızrak, "Comparison of activation functions for deep neural networks," (2020).
[15] D. John et.al, "Adaptive subgradient methods for online learning and stochastic optimization," JMLR **12**, 2121–2159 (2011).