Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering
(38^{th}cycle)

Device-Level Modeling of Photonic Integrated Circuits for Neuromorphic Computing

Andrea Marchisio
******

Supervisor(s):
Prof. Andrea Carena, Supervisor
Prof. Paolo Bardella, Co-Supervisor
Prof. Vittorio Curri, Co-Supervisor

Doctoral Examination Committee:
Prof. Andrea Melloni, Referee, Politecnico di Milano
Dr. Antonio Fincato, Referee, ST Microelectronics

Politecnico di Torino
2026

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Andrea Marchisio
2026

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

I had no idea, as I abided in the unshaken belief that the time of cruel wonders was not yet over.

Stanisław Lem - *Solaris*

Acknowledgements

First, I would like to thank my supervisors Andrea Carena, Vittorio Curri, and Paolo Bardella, not only for their tireless technical support during the research activities of my PhD, but also for their constant help while I was navigating the bureaucracy of the Italian University.

A huge thank you to Enrico Ghillino from Synopsys Inc. for his three-year-long collaboration and insight, and to Sander Roosendaal, Joao Marques, Charalampos Mavidis, and all the good people from the Synopsys office in Enschede for hosting me for six months and helping me acquire new, invaluable skills.

My gratitude also goes to Prof. Patty Stabile and to Antonio Lechiara from the Technical University of Eindhoven for the incredible opportunity I had during my six-month research period there. Not only did I grow as a researcher with the photonic laboratory activity, but I also met wonderful people and had amazing experiences that I will cherish forever.

Finally, I would like to thank Prof. Francesco Da Ros from the Technical University of Denmark for kindly providing the experimental data used to validate our MZI model.

I would like to dedicate this thesis to my colleagues from the "Acquario" office and the "Marconi" (unofficial) lunchroom at Politecnico di Torino, especially Saverio, Leonardo, Lorenzo, and Lorenzo Tunesi, for partaking in my insanity.

To my parents Piercarlo and Maria Teresa, my sisters Silvia and Sara, and my grandmother Rita, for supporting me with selfless love in all my life choices (no matter how ill-advised).

To Valeria, for being who you are. Every day you make me a better person.

Abstract

In recent years, the world experienced the Artificial Intelligence (AI) revolution: since the launch of ChatGPT in 2022, AI has suddenly become an extremely popular topic, not only at the academic research level, but also for the broader general public, on one hand, sparking countless moral discussions about its ethical implications, on the other, attracting billions of dollars in investments from the major tech companies (e.g., Google, Meta, NVidia, etc.).

This AI surge has been powered both by the computational capabilities of modern computers and by the capillary availability of data. Indeed, AI, in particular Machine Learning (ML), is a "data-driven" application, meaning that it requires vast and diverse datasets in order to be effectively trained. However, the required dataset size has been doubling every 3.5 months and, at some point, traditional computers based on the Von Neumann architecture will no longer be able to effectively handle the input data.

In this context, a new paradigm could be required to maintain the projected AI trends. Photonic Integrated Circuits (PICs) appear to be an interesting candidate for the creation of the next-generation hardware accelerators for AI. Indeed, despite being a relatively new and less mature technology with respect to standard CMOS, PICs offer several beneficial advantages. For instance, they show very high operational speed (information is carried by light), low power consumption (most of the commonly used components are fully passive), and high parallelism (various techniques could be employed, such as wavelength, polarization, and mode division multiplexing). Moreover, photonic platforms can implement matrix multiplications very naturally, whereas traditional Von Neumann architectures perform them inefficiently due to the high number of memory accesses required.

To leverage these advantages, it is possible to design PICs for neuromorphic computing applications. Among the possible strategies, it is possible to create Photonic

Neural Networks (PNNs) made with a linear section performing the weighted sum of the inputs and a nonlinear section applying the nonlinearity to the output, thus mimicking the operation of a traditional artificial neuron.

In this context, the work of this thesis tackles the modeling of PICs for photonic neuromorphic computing applications, dealing with both the linear and nonlinear sections of the PNNs.

In particular, the linear section of the PNN is typically implemented with a mesh of interconnected thermally-controlled Mach-Zehnder Interferometers (MZIs): by controlling the propagation of light through the MZIs via thermal tuning, the weighted sum can be performed. However, without proper insulation, neighboring devices will influence each other with spurious thermal interactions (i.e. thermal crosstalk), which can strongly reduce the final accuracy of the trained PNN. For this reason, we have developed a comprehensive model for MZI meshes able to predict both the light propagation through the circuit and the thermal crosstalk between the devices. The model is validated with the measurements of a 3X3 MZI mesh with incomplete interconnections and it is shown to accurately predict the crosstalk in this reference PIC.

Furthermore, in this thesis, an all-optical Nonlinear Activation Function (NLAF) device able to implement nonlinearities for neuromorphic applications has been analyzed. This device is realized on the Indium phosphide Membrane on Silicon (IMOS) technological platform and it is made with an MZI cascaded with a ring-assisted MZI. A model for predicting the nonlinearities of the IMOS Microring Resonator (MRR) has been constructed and validated against experimental data and it has been used to create a simulative model for the complete NLAF device.

Publications

Publications in peer reviewed journals

[1] A. Marchisio, E. Ghillino, V. Curri, A. Carena, P. Bardella, "Particle swarm optimization-assisted approach for the extraction of VCSEL model parameters" in *Optics Letters*, vol. 49, no.1, pp. 125-128, Dec. 21, 2023, DOI: 10.1364/OL.506958

[2] A. Marchisio, F. Da Ros, V. Curri, A. Carena, P. Bardella, "Comprehensive model of MZI-based circuits for photonic computing applications", in *Nature Portfolio Communication Physics*, vol. 8, no. 277, Jul. 3, 2025, DOI: 10.1038/s42005-025-02176-0

Publications in international conference proceedings

[1] A. Marchisio, I. Khan, L. Tunesi, M.U. Masood, E. Ghillino, V. Curri, A. Carena, P. Bardella, "Automated model for characterization of VCSEL circuit-level parameters using machine learning", in *European Conference on Integrated Optics (ECIO)*, Enschede, The Netherlands, Apr. 19 - 21, 2023

[2] A. Marchisio, L. Tunesi, A. F. Forrest, M. A. Cataluna, M. Krakowski, P. Bardella, "Time-Domain Travelling-Wave Analysis of Semiconductor Optical Amplifiers based on Chirped Quantum Dot Materials", in *International Conference on Lasers and Electro-Optics (CLEO)*, San Jose, CA, USA, May 7 - 12, 2023, DOI: 10.1364/CLEO_AT.2023.JTh2A.63

[3] I. Khan, A. Marchisio, L. Tunesi, M. U. Masood, E. Ghillino, V. Curri, A. Carena, P. Bardella, "A Machine Learning-Based Model for Characterizing Stationary-and-Dynamic Behavior of VCSEL", in *International Conference*

on Lasers and Electro-Optics (CLEO), San Jose, CA, USA, May 7 - 12, 2023, DOI: 10.1364/CLEO_AT.2023.JW2A.141

[4] L. Tunesi, I. Khan, M. U. Masood, A. Marchisio, E. Ghillino, V. Curri, A. Carena, P. Bardella, "Machine Learning Aided Prediction of Fabrication Uncertainties in Integrated Multi-Ring Filters", in *International Conference on Lasers and Electro-Optics (CLEO)*, San Jose, CA, USA, May 7 - 12, 2023, DOI: 10.1364/CLEO_SI.2023.STh4H.2

[5] A. Marchisio, V. Curri, A. Carena, P. Bardella, "Deep Learning-Driven Extraction of Superluminescent Diodes Parameters", in *International Conference on Numerical Simulation of Optoelectronic Devices (NUSOD)*, Torino, Italy, Sep. 18 - 21, 2023, DOI: 10.1109/NUSOD59562.2023.10273512

[6] A. Marchisio, A. Cem, Y. Ding, V. Curri, A. Carena, F. Da Ros, P. Bardella, "Optimization of 3x3 neuromorphic photonic network for programmable Boolean operations," in *SPIE Photonic West, Physics and Simulation of Optoelectronic Devices XXXII*, San Francisco, CA, US, Jan. 27 - Feb. 1, 2024, DOI: 10.1117/12.3002953

[7] A. Marchisio, E. Ghillino, V. Curri, A. Carena, P. Bardella, "Particle swarm optimization hyperparameters tuning for physical-model fitting of VCSEL measurements", in *SPIE Photonic West, Vertical-Cavity Surface-Emitting Lasers XXVIII*, San Francisco, CA, US, Jan. 27 - Feb. 1, 2024, DOI: 10.1117/12.3002576

[8] M. U. Masood, I. Khan, L. Tunesi, B. Correia, A. Marchisio, E. Ghillino, P. Bardella, A. Carena, V. Curri, "Waveband selective switch: a network analysis for advanced optical transport networks in 6G and beyond technologies", in *SPIE Photonic West, Next-Generation Optical Communication: Components, Sub-Systems, and Systems XIII*, San Francisco, CA, US, Jan. 27 - Feb. 1, 2024, DOI: 10.1117/12.300099

[9] M. U. Masood, I. Khan, L. Tunesi, B. Correia, A. Marchisio, E. Ghillino, P. Bardella, A. Carena, V. Curri, "Impact of Waveband and Wavelength Switching in the Next-Generation Optical Networks", in *International Conference on Transparent Optical Networks (ICTON)*, Bari, Italy, Jul. 14 - 18, 2024, DOI: 10.1109/ICTON62926.2024.10647953

[10] A. Marchisio, A. Carena, V. Curri, P. Bardella, “PSO-assisted extraction of VCSEL parameters from LI and S21 measurements”, in *International Conference on Numerical Simulation of Optoelectronic Devices (NUSOD)*, New Delhi, Sep. 23 - 27, 2024, DOI: 10.1109/NUSOD62083.2024.10723706

[11] M.U. Masood, I. Khan, L. Tunesi, A. Marchisio, E. Ghillino, P. Bardella, A. Carena, V. Curri, “Networking Study of Waveband/Wavelength Switching for Multi-Granular and Multi-Band Optical Networks”, in *IEEE Global Communications Conference (GLOBCOM)*, Cape Town, South Africa, Dec. 8 - 12, 2024, DOI: 10.1109/globecom52923.2024.10901718

[12] A. Marchisio, L. Tunesi, E. Ghillino, V. Curri, A. Carena, P. Bardella, “Comprehensive thermal crosstalk model of meshed MZI topologies for neuromorphic computing”, in *SPIE Photonic West, AI and Optical Data Sciences VI*, San Francisco, CA, USA, Jan. 25 - 31, 2025, DOI: 10.1117/12.3043230

[13] T. Ma, A. Marchisio, F. Da Ros, V. Curri, P. Bardella, A. Carena, “Wideband operation of 3x3 Mach-Zehnder Interferometer mesh for programmable photonic computing”, in *International Conference on Transparent Optical Networks (ICTON)*, Barcelona, Spain, Jul. 6 - 10, 2025, DOI: 10.1109/ICTON67126.2025.11125174

[14] T. Ma, A. Marchisio, F. Da Ros, V. Curri, P. Bardella, A. Carena, “Investigation on digital twin model of 3x3 Mach-Zehnder Interferometer mesh”, in *International Conference on Numerical Simulation of Optoelectronic Devices (NUSOD)*, Łódź, Poland, Sep. 14 - 18, 2025, DOI: 10.1109/NUSOD64393.2025.11199402

[15] A. Lechiara, A. Marchisio, B. Shi, Y. Wang, A. Carena, Y. Jiao, R. Stabile, “Ultra-Compact Leaky ReLU Nonlinear Function on IMOS”, in *European Conference on Optical Communications (ECOC)*, Copenhagen, Denmark, Sep. 28 - Oct. 2, 2025, DOI: 10.1109/ECOC66593.2025.11263433

[16] A. Lechiara, A. Marchisio, B. Shi, Y. Wang, A. Carena, Y. Jiao, R. Stabile, “Reprogrammable All-Optical Ultra-Compact Nonlinear Activation Function for Neuromorphic Computing on Indium Phosphide Membrane on Silicon”, in *IEEE Photonics Benelux Annual Symposium*, Bruxelles, Belgium, Nov. 20 - 21, 2025

Contents

List of Figures

List of Tables

Acronyms

Chapter 1

Introduction

1.1 Context and motivation

Photonic Integrated Circuits (PICs) are microchips that manipulate and process light similarly to how traditional electronic integrated circuits manipulate electron currents. PICs can be realized on multiple technological platforms based on multiple semiconductor materials, such as Silicon-on-Insulator (SOI), Indium Phosphide (InP), Silicon Nitride (SiN), Lithium Niobate ($LiNbO_3$), but the general idea is to integrate both passive (waveguides, splitters, etc.) and active components (lasers, detectors, amplifiers, etc.) on the same chip, to perform complex operations in the optical domain.

PICs offer multiple advantages related to their inherent physical operation. Indeed, since PICs operate with optical signals traveling at the speed of light, they are characterized by large bandwidth and low latency. Moreover, light and photonic components can host multiple multiplexing techniques (i.e. wavelength, mode, polarization division multiplexing), which enable high parallelization of the operations performed by the circuit.

Another important advantage of PICs is that the majority of the commonly used integrated components are passive and PICs in general are not affected by resistive heating losses, meaning that these photonic circuits are characterized by good energy efficiency compared to traditional integrated circuits. Although PICs are a less mature technology with respect to Complementary Metal-Oxide-Semiconductor (CMOS), for which billions of transistors can be integrated on the same chip, in

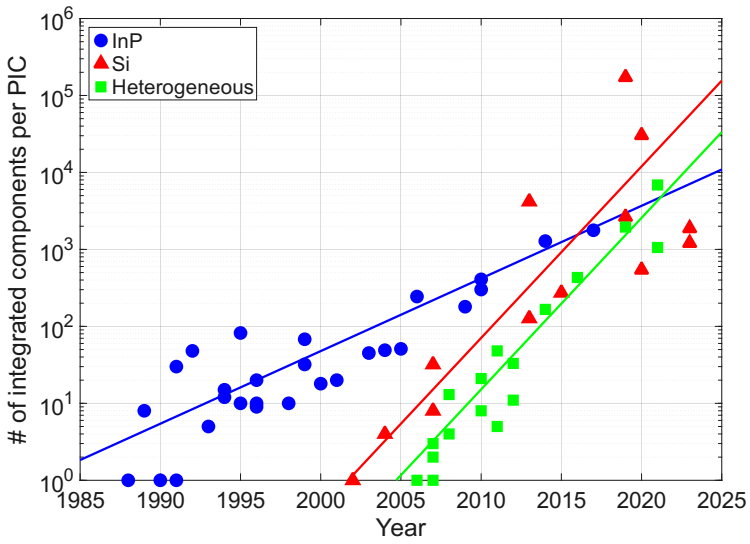

Figure 1.1 Graphical representations of the evolution over the years of the number of integrated components per PIC for different technological platforms: monolithic InP (blue), monolithic Si (red), and heterogeneous solutions (green). Data points from [1].

recent years, great advancements have been made for what concerns the integration density of photonic components. A representation of the number of integrated photonic components per PIC over the years for different technological platforms is reported in Fig. 1.1. Even denser integration could be enabled by the use of foundry-based CMOS processes for the fabrication of PICs, not only increasing the number of integrated components per PIC, but also leveraging the high volumes to reduce production costs.

Moreover, with PICs, it is possible to co-integrate photonic components with electro-optical devices on the same chip. This is important to expand the range of possible operations that can be performed with PICs, thus increasing the number of possible application fields.

Finally, each technological platform offers its unique advantages with respect to the others. For instance, with SOI, it is possible to better exploit the CMOS processes, while with InP, it is possible to integrate active sources more naturally, thanks to the InP direct bandgap.

Thanks to these advantages, PICs have garnered more and more success for many commercial applications and in many research fields. The most prominent application is undoubtedly related to telecommunications, since PICs are employed both for the creation of transceivers for long-haul communications and for short-haul

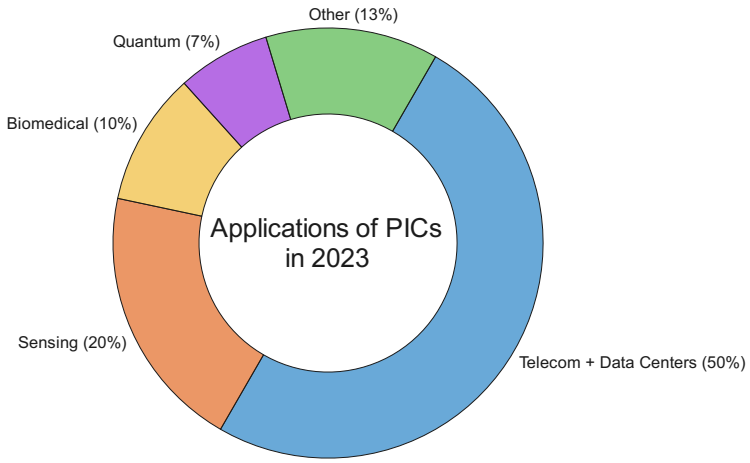

Figure 1.2 Estimation of the market share of different applications of PICs in 2023. Data from [2].

intra-data center fast interconnects. In addition to this, PICs are also a popular choice for the creation of on-chip switching and multiplexing systems. In total, telecom applications represent about 50% of the total market for PICs [2].

Another important field of application for PICs is represented by sensing, which can be subdivided into fiber, biomedical, and Light Detection and Ranging (LiDAR) sensing. For instance, PICs have been used to detect specific chemical compounds both for environmental monitoring (e.g., gas or water pollutant sensors) and for medical/biomedical diagnosis (e.g., glucose level monitoring or viral infection detection) [3]. Researchers have proposed the idea of creating photonic Laboratories-on-Chip (LoCs), capable of performing multiple sensing operations on a compact PIC [4]. Instead, LiDAR is an imaging technique that exploits light from a laser and its reflections to create a 3D rendition of the surrounding space. For instance, PICs have been employed to create LiDAR Systems-on-Chip (SoCs) for autonomous driving [3].

Then, PICs have started to be employed for Quantum Computing applications, because of their capability of implementing integrated single-photon sources, of manipulating quantum states, and performing on-chip metrology [5]. PICs have the potential to open a path towards better scaling for quantum technology.

Finally, in recent years, after the exponential growth of Artificial Intelligence (AI) applications, PICs have also been envisioned as hardware accelerators for AI,

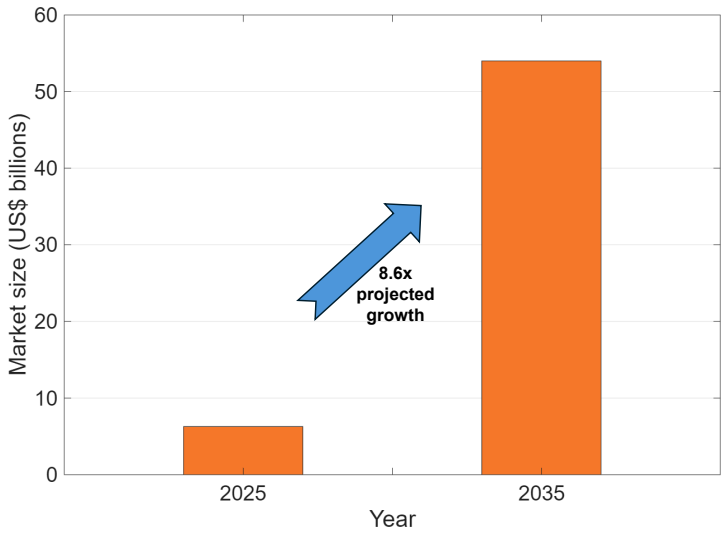

Figure 1.3 Projected market size growth for PICs between 2025 and 2035. Data from [6].

exploiting their inherent advantages for more efficient training procedures. These are called photonic neuromorphic computing applications.

Due to the increasing popularity of PICs for an ever-growing number of cutting-edge technological applications, market analysts predict the PIC market to exceed US$54 billion mark by 2035, starting from a US$6 billion market value in 2025, with data center applications at the forefront of this 8.6x growth [6].

Although the PIC market will be dominated by datacom and telecom applications in the coming years, photonic neuromorphic applications are also of particular interest, driven by the rapid growth of AI. To date, most Machine Learning (ML) applications have relied on traditional computers for the training of Neural Networks (NNs); however, this approach is not optimal, as training procedures require a large number of matrix multiplications. Indeed, traditional computers are based on the Von Neumann architecture, in which memory and processing units are spatially separated, resulting in significant computational overhead for operations such as matrix multiplications due to frequent memory accesses. As dataset sizes will inevitably continue to increase, this inefficiency may become a major bottleneck for the development of next-generation AI platforms.

For this reason, leveraging the advantages offered by PICs can pave the way to the creation of hardware accelerators for AI applications, ideally capable of performing high-accuracy training in the optical domain and, at the same time,

solving the computational inefficiency issue of traditional computers. This topic will be expanded in Chapter 2.

1.2 Thesis outline

Given the general context of PICs and AI discussed briefly in Section 1.1, this work will be articulated as follows:

Chapter 2 will expand upon the discussion begun in Section 1.1 with a detailed state-of-the-art review for the prominent photonic neuromorphic computing solutions, starting from a general discussion about theoretical aspects of traditional Machine Learning, in order to understand how PICs could implement the main neuromorphic functionalities, potentially overcoming traditional AI bottlenecks, thanks to the inherent advantages of photonic technology. In particular, the main photonic neuromorphic solutions will be presented, namely Reservoir Computing, Spiking Neural Networks, and photonic Fully-Connected Neural Networks.

Chapter 3 will focus on the modeling of linear neurons for photonic Fully-Connected Neural Networks, implemented with meshes of thermally-controlled Mach-Zehnder Interferometers (MZIs). In particular, these thermally-tuned devices are affected by thermal crosstalk: if multiple neighboring devices are operated at the same time, spurious thermal interaction will effectively reduce the thermal tunability efficiency of these devices, in turn reducing the accuracy of the neural network in the context of its training. Therefore, in this chapter, we develop a comprehensive model that describes both the light propagation through the linear neuron and the spurious thermal interactions between the building blocks. The proposed model is validated by reproducing the measurements of a 3×3 MZI mesh with incomplete interconnections. As an example, the model is then employed to find suitable driving voltages to drive the nine MZIs of the reference device to implement user-defined Boolean functions on a large operational bandwidth.

Chapter 4 will provide an analysis of a device capable of implementing all-optical nonlinear activation functions for neuromorphic computing applications. This device is implemented on the Indium Phosphide Membrane on Silicon (IMOS) platform and its geometry comprises an MZI tunable coupler cascaded with a ring-assisted MZI. First, a model for the description of the nonlinear effects affecting an

IMOS-based Microring Resonator (MRR) will be developed and validated through direct comparison with laboratory measurements. Then, this model will be employed for the simulation of the complete nonlinear activation function device and the exploration of its solution space.

Appendix A contains an in-depth analysis of the Particle Swarm Optimization (PSO) algorithm, a technique that was employed across the various avenues of research presented in this thesis, especially whenever the fitting of experimental data was required. First, the rules of the basic algorithm will be explained, followed by the description of an advanced PSO variant, called Adaptive Particle Swarm Optimization (APSO). Finally, the algorithm is applied to an explanatory example problem, namely the extraction of physical model parameters from Vertical-Cavity Surface-Emitting Laser (VCSEL) power and small signal measurements.

Chapter 2

State-of-the-art of photonic neuromorphic computing

This chapter contains a comprehensive review of the state-of-the-art for Photonic Neural Networks, starting from the history and fundamental concepts at the base of Machine Learning, in order to explain the possible advantages that could stem from a change of paradigm with respect to the use of traditional computers, in particular with the use of photonic platforms.

2.1 Fundamentals of Artificial Neural Networks

2.1.1 Machine learning: a brief chronology

AI has recently become a prime subject of discussion in mainstream media and in the academic research discourse, especially since AI-powered tools such as OpenAI's ChatGPT sparked the interest of the general public with its debut in 2022 [7].

In addition to the recent surge in interest, AI has a long history as a computer science research field: in 1943, McCulloch and Pitts laid the foundational work for the definition of Artificial Neurons (ANs) and Artificial Neural Networks (ANNs) [8], while, in 1950, Turing first defined the concept of artificial intelligence along with his Turing test [9]. After these seminal works, in 1958, Rosenblatt first proposed the concept and mathematical model for the perceptron [10], one of the first practical

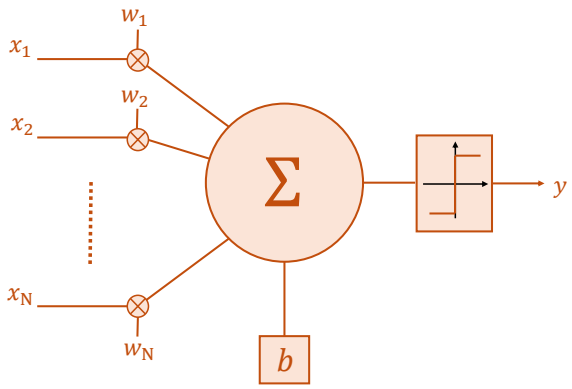

Figure 2.1 Schematic representation of a single-layer perceptron. x_i with $i = \{1, 2, \ldots, N\}$ are the input signals, w_i with $i = \{1, 2, \ldots, N\}$ are the multiplicative weights that modify the corresponding x_i, Σ represents a summation operation, b is the additive bias, and y is the output. After the summation of the weighted inputs and addition of the bias, a nonlinear activation is applied.

and functioning implementations of an AN and a key instrument for the future developments of ML. In 1967, Amari trained the first Multi-Layer Perceptron (MLP) via stochastic gradient descent [11], effectively creating one of the first working instances of Deep Neural Network (DNN). Further developments came decades later in 1986, when Rumelhart, Hinton, and Williams employed the backpropagation algorithm in the training procedure of an MLP [12], popularizing this technique. Indeed, in 1989, Le Cun *et al.* applied this backpropagation algorithm to the training of a Convolutional Neural Network (CNN) for the task of classifying handwritten digits [13]. From this point onward, a long list of improvements in the field has been published in the literature. For instance, in 2012, Krizhevsky, Sutskever, and Hinton successfully trained AlexNet [14], a deep CNN capable of performing complex image recognition tasks with record-breaking accuracy.

This was a brief overview of the main achievements that paved the way to the development of modern NNs, that lead to the pervasive diffusion of AI in everyday life with the explosion of Large Language Models (LLMs), which are capable of performing extremely complicated tasks such as human-like image and text generation. In the next subsection, we will start to analyze the perceptron, i.e. the simplest AN that can be implemented and cascaded to create ANNs.

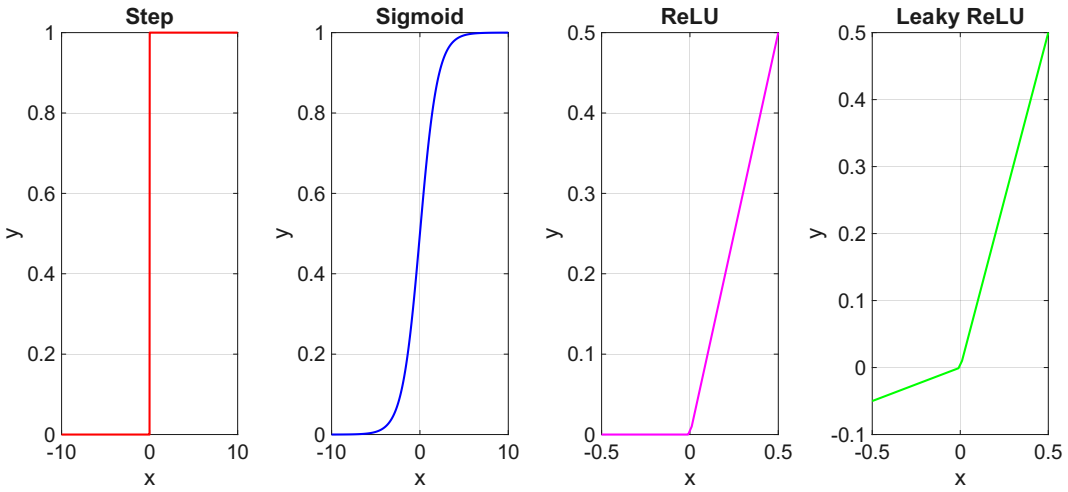

Figure 2.2 Different types of common activation functions: step (red), sigmoid (blue), ReLU (magenta), and Leaky ReLU (green).

2.1.2 Perceptron and Multi-layer perceptron

As mentioned before, the perceptron was proposed by Rosenblatt as a simple mathematical model to emulate the basic operation of a biological neuron. A schematic of the perceptron is contained in Fig. 2.1. A set of N inputs x_i is weighted by just as many multiplicative weights w_i. These are linearly combined and then summed to an additive term b, called "bias". Finally, the output of the perceptron y is obtained by applying a nonlinear activation function $f(\cdot)$, which will establish a different value for the output based on the result of the weighted sum of the inputs, thus determining whether the perceptron is activated or not. This can be expressed in mathematical terms as [15]:

$$y = f\left(\sum_{i=1}^{N} w_i x_i + b\right) \tag{2.1}$$

Depending on the target application, the activation function $f(\cdot)$ can be defined in different ways. The simplest example is a binary step function, where y is set to 1 if the weighted sum yields a result larger than a given threshold or 0 otherwise (Fig. 2.2(a)). The problem is that this function is non-differentiable, meaning that it cannot be employed effectively in a gradient descent algorithm, which is a mandatory part of the training process. Therefore, it is a better choice to employ the sigmoid

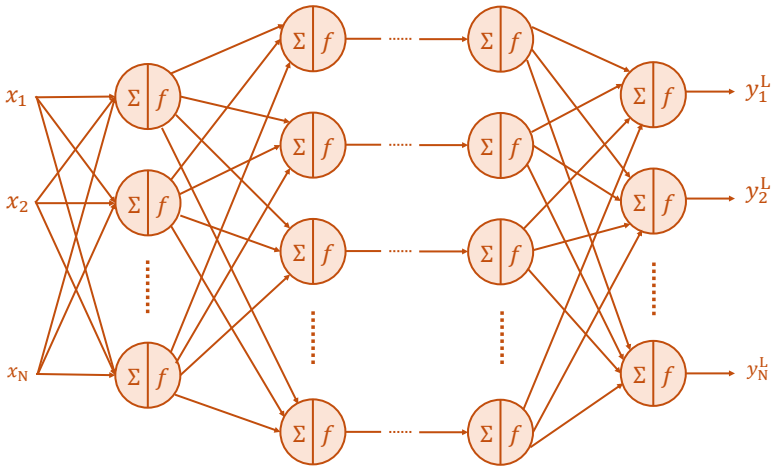

Figure 2.3 Schematic representation of an MLP/FFNN. Each neuron of the network is a perceptron with the structure of Fig. 2.1.

function, which assumes the values 0 and 1 like the step function, but with a smoother transition in between (Fig. 2.2(b)).

In order to train a perceptron, two fundamental tools are required: first, we need an optimizer (e.g., the gradient descent algorithm mentioned before) to iteratively change the value for weights and bias, with the goal of minimizing a given error metric until the wanted input-output relationship is provided by the perceptron; second, the error metric itself, usually called “loss function”, which measures the distance between the output of the perceptron and the desired result.

If the output y of a perceptron is used as input of other perceptrons, it is possible to create an MLP, also commonly called Feed Forward Neural Network (FFNN). This name comes from the fact that the information can only move from the input to the output, without any feedback loops, although there exist advanced NN implementations containing feedback loops, such as the Recurring Neural Networks (RNNs).

Each perceptron in the MLP/FFNN acts as an AN. By stacking N perceptrons in parallel, we have a “hidden layer" with width N. By cascading L hidden layers, we create an MLP/FFNN with depth L. This is one of the most common implementations of modern DNNs, with the structure schematically represented in Fig. 2.3. Each layer l receives the inputs from the layer $l-1$ and generates the inputs for the layer $l+1$, by applying the transformation of Eq. (2.1) through each AN of the layer.

In the case of MLP for Deep Learning (DL), the most commonly used nonlinear activation functions are the sigmoid (Fig. 2.2(b)), the Rectified Linear Unit (ReLU) (Fig. 2.2(c)), and the Leaky ReLU (Fig. 2.2(d)), even though there exist countless variations and exotic activations functions to target specific applications and problems [16].

2.1.3 Deep Neural Networks training techniques

In this subsection, practical details regarding the training of a DNN will be examined.

Loss function

The first important step for the training of a NN is the definition of a "loss" (or "cost") function $J(\theta)$, where θ are the trainable parameters of the network. Through the dependency of the NN response on the parameters θ, this function $J(\theta)$ gives us a measure of how far the predictions are from the desired output (e.g., correct classification of the input data).

There are many different mathematical implementations of the loss function, usually tailored to the application at hand. For instance, for binary classifications (i.e. a classification problem where the outputs can only be 0s and 1s), the go-to loss function is called Binary Cross-Entropy (BCE). Instead, one of the most common loss functions for regression and classification problems is called Mean Squared Error (MSE) and it consists in computing the average error for each element of the output vector with respect to the corresponding "true" value. For instance, for an MLP/FFNN with a single output neuron and with sample size N, we can define it as

$$J_{\text{MSE}}(y_i, \hat{y}_i) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.2}$$

where y_i is the i-th sample of the desired output and $\hat{y}_i$ is the i-th sample of the NN prediction. In the case of a NN as the one represented in Fig. 2.3, where there are multiple output neurons, this error is computed for each one of the ANs in the output layer.

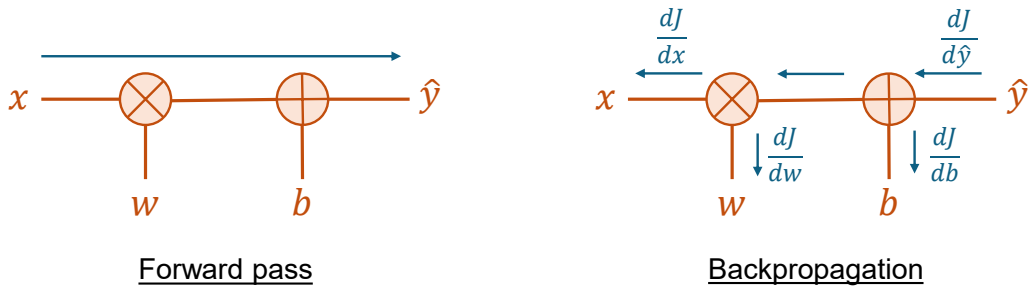

Figure 2.4 Forward pass and gradient backpropagation for a single-layer NN.

Gradient backpropagation

During the training of the network, there is first a "forward pass", where the inputs are processed with the current setting of the NN. The generated output predictions, during this first pass, are employed to compute the loss function $J(\theta)$, for instance with the formulation of Eq. (2.2).

At this point, we have to compute the gradient of $J(\theta)$ with respect to all the parameters θ of the NN (∇_θ). This can be done with what is usually called "gradient backpropagation" in ML jargon, which is essentially Euler's chain rule of derivatives. For simplicity, let us consider a single-layer NN that applies the transformation $\hat{y} = wx + b$ during the forward pass. It is possible to define the loss function $J(y, \hat{y})$ that depends on the given transformation $\hat{y}$. Starting from the output, we can compute the derivatives of J with respect to each parameter of $\hat{y}$:

$$\frac{dJ}{db} = \frac{dJ}{d\hat{y}}\frac{d\hat{y}}{db} = \frac{dJ}{d\hat{y}}\frac{d}{db}(wx+b) = \frac{dJ}{d\hat{y}} \tag{2.3}$$

$$\frac{dJ}{dw} = \frac{dJ}{d\hat{y}}\frac{d\hat{y}}{dw} = \frac{dJ}{d\hat{y}}\frac{d}{dw}(wx+b) = \frac{dJ}{d\hat{y}}x \tag{2.4}$$

$$\frac{dJ}{dx} = \frac{dJ}{d\hat{y}}\frac{d\hat{y}}{dx} = \frac{dJ}{d\hat{y}}\frac{d}{dx}(wx+b) = \frac{dJ}{d\hat{y}}w \tag{2.5}$$

This concept can be extended to include nonlinear activation functions, additional ANs, and additional layers, by chaining the various parameters of the applied transformations.

The backpropagation is extremely important because it allows us to compute the gradient of the loss function with respect to all trainable parameters of the network,

which is a fundamental ingredient for the optimization procedure of the weights of the NN while training.

Stochastic gradient descent algorithm

The Stochastic Gradient Descent (SGD) algorithm is one of the possible procedures that can be employed to optimize the trainable parameters of a NN, with the goal of minimizing the loss function (i.e. the prediction error at the output of the NN). Given the set of trainable parameters of the network θ and the gradient of the loss function J with respect to each of these parameters $\nabla_\theta J$ computed with the gradient backpropagation strategy, we can update these parameters θ with the following strategy:

$$\theta \leftarrow \theta - \eta \nabla_\theta J \tag{2.6}$$

where η is the learning rate.

One problem with gradient descent is that NNs are highly nonlinear, meaning that the optimization of J is a non-convex problem [15]: indeed, J could be characterized by the existence of local minima that could hinder its minimization and, consequently, the optimization procedure of the weights, yielding lower accuracy of the trained NN. For this reason, SGD employs random subsets of the complete dataset (called "mini-batches") in order to compute the gradient $\nabla_\theta J$, thus using an estimate instead of the actual gradient. Thanks to the random selection of the mini-batch, the value of the estimated gradient will have random fluctuations, which help the optimizer to exit a local minimum.

The state-of-the-art weight optimizer in DL is a variation of the SGD algorithm, called **ADAM** (Adaptive Moment estimation) [17], where both the gradient ("first moment") and the squared gradient ("second moment") are estimated and employed in order to compute per-parameter adaptive learning rates.

Weight initialization

The initialization of the trainable parameters is a very important practical aspect, since the wrong starting point choice can hinder the optimal convergence of the NN.

Not only does the initialization determine whether learning converges or not, but, when it does, the choice also influences the convergence speed and the final loss function value [15]. Clearly, it is important to carefully initialize the weights of the NN with an appropriate choice.

The most naive weight initialization choice is the **zero initialization**, where all weights are initialized to zero or to a small value. However, this is a poor choice because it introduces symmetry: all neurons in a layer receive identical gradients and therefore learn identical features, preventing the network from learning useful representations.

For this reason, the best practice for the initialization of the weights consists in avoiding creating symmetries inside the NN, because ANs initialized with the same parameters will perform the same transformation and therefore learn redundantly. For this reason, the most common idea is to employ randomly-generated initial weights, drawn from a Gaussian or uniform distribution. Despite the benefits of employing a random distribution for the initial weights, their scale is also a major issue. If the initial weights are large in magnitude, we are sure that the signals will be able to reach the deeper hidden layers, but we might experience the exploding gradient problem. On the other hand, employing smaller random weights could avoid exploding gradients, but it may result in slow learning due to the smaller values of the gradients (vanishing gradient problem).

The **Glorot** (or **Xavier**) initialization is a compromise between these two extremes [18], able to prevent both exploding and vanishing gradients. The weights of the i-th layer are initialized with a Gaussian distribution with mean $\mu = 0$ and variance [18]

$$\sigma = \sqrt{\frac{2}{n_i + n_{i+1}}} \tag{2.7}$$

where n_i is the number of neurons in the i-th layer and n_{i+1} is the number of neurons in the next one. Alternatively, the set of weights W could be generated with a uniform distribution with [18]

$$W \sim U\left[\pm\sqrt{\frac{6}{n_i + n_{i+1}}}\right] \tag{2.8}$$

The Glorot initialization works best with layers that have zero-centered activation functions (e.g., sigmoid or tanh). When we employ ReLU layers with Glorot-initialized weights, the activation functions collapse to zero, thus hindering learning. For this reason, the **He** (or **Kaiming**) initialization has been proposed for NN employing ReLU-like functions [19]. For the weights of the i-th layer, the He initialization uses a normal distribution with mean $\mu = 0$ and variance [19]

$$\sigma = \sqrt{\frac{2}{n_i}} \tag{2.9}$$

While weights are typically initialized with random values, biases and other trainable parameters of the NN are initialized with properly-chosen empirical values [15].

Choosing the nonlinear activation function

Another important research topic for ML is the optimization of the neuron activation functions. Here we will briefly discuss the most commonly deployed activation functions in modern AI.

Historically, the first activation function to be extensively used for ML was the **logistic sigmoid** function (Fig. 2.2(b)), which gained popularity due to its resemblance to the firing behavior of natural neurons. The sigmoid is defined as

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.10}$$

This function squashes the output values in the range $[0,1]$, saturating the value to 0 and 1 for very small or very large input values, respectively. This is the reason for one of the main issues of the sigmoid, namely the vanishing gradient problem: since this activation function has saturating behavior, there its gradient will assume very small values, inhibiting learning. Moreover, the fact that $\sigma(x)$ is always positive (non-zero-centered) implies that the gradient of the activation functions will have the same sign during backpropagation. As a result, weights are often updated in the same direction, leading to inefficient optimization and slower learning. This problem can be tackled by deploying zero-centered functions like the **hyperbolic tangent** (tanh) function, which has a shape similar to the sigmoid, but centered around the y-axis origin. An additional problem regarding the use of the logistic sigmoid function

is the presence of the exponential function in its definition. Indeed, it is a more expensive operation to compute, especially on older hardware.

Another popular choice is the use of the **ReLU** function (Fig. 2.2(c)), which is a piecewise function with $f(x) = 0$ if $x \leq 0$ and linear behavior if $x > 0$:

$$f(x) = \max(0, x) \tag{2.11}$$

This function is able to solve multiple issues presented by the logistic sigmoid: first, it does not saturate, thus mitigating the vanishing gradient issue; second, it is a quasi-linear function, which is much more computationally efficient with respect to the sigmoid; third, in practice, it allows for a much faster convergence with respect to the sigmoid and tanh functions. However, it is affected by the "dying ReLU" problem, which is due to the fact that the ReLU is always equal to zero for negative x values and so will be its gradient (i.e. the AN is not learning).

This disadvantage of the ReLU function can be easily solved with the use of a variation, called **Leaky ReLU** (Fig. 2.2(d)). In this case, instead of always having 0 for negative input values, the function is linear with a small slope:

$$f(x) = \max(0.1x, x) \tag{2.12}$$

In this way, we are able to harness all the advantages of the ReLU function, but without experiencing the dying ReLU issue. For this reason, the Leaky ReLU function is one of the most deployed activations in modern ML.

Overfitting prevention techniques

When training a NN, we want the network not only to achieve good accuracy while training (ability to reproduce the training dataset), but also to maintain good performance when dealing with unseen data (part of the testing dataset). If the NN has very accurate predictions on the training set, but performs poorly on the testing set, we might be in the presence of "overfitting". In this case, the network effectively memorizes the training set, but it lacks generalization capabilities. Fig. 2.5 depicts graphically how overfitting can manifest when fitting a set of data points.

Overfitting can be caused by multiple factors, including:

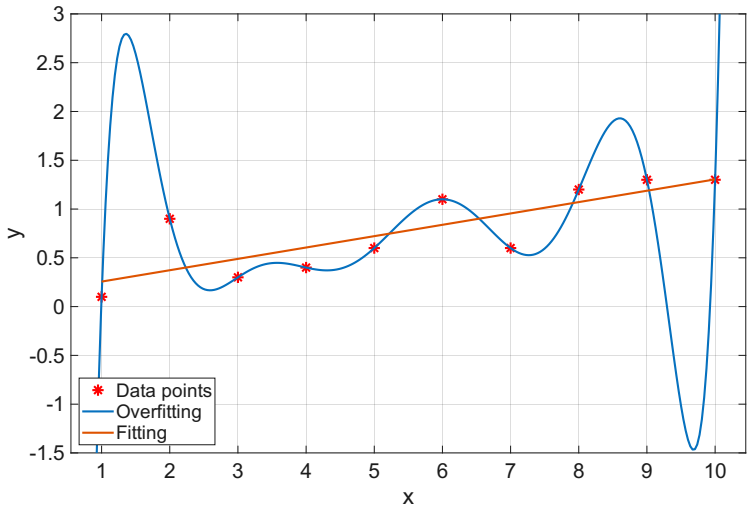

Figure 2.5 Example of fitting (orange line) vs. overfitting (blue line) of some data (red asterisks).

1. **the dataset is too small**: if the training dataset is not large enough, the data may not be representative enough of the real data distribution, thus the NN will not be able to learn the general patterns;

2. **the dataset is too noisy**: if the training dataset is affected by many outliers or errors, the NN may be trying to learn these, which are not representative of the real data distribution;

3. **the NN is too complex**: if the network is too deep and/or too wide (too many layers and neurons per layer, respectively), it will be prone to overfitting the dataset, including noise and outliers;

4. **the NN is trained for too long**: if the network is trained for too many iterations (or "epochs"), it will be able to fit increasingly subtle details of the data distribution, including noise and outliers;

For this reason, a popular avenue of research of ML is "overfitting prevention" and many techniques have been proposed to increase the accuracy of NNs. A very simple, yet effective idea is **early stopping**: since the model starts overfitting as the training continues for too many epochs, we can stop this process at the optimal time, namely when the loss on a separate validation dataset stops improving or starts to deteriorate during training.

Another approach consists in using "regularization", which is defined as any modification of the loss function done in order to improve the generalization capabilities of the network [15]. In particular, **L regularization** techniques are commonly employed in modern ML training procedures and they exploit the fact that small weights make the network more robust to noise. The first class of L regularizations is the **L1**, which encourages some of the weights of the model to be null ("sparse weights") by modifying the loss function with the sum of the absolute values of the weights:

$$J_{\mathrm{L1}} = J + \sum_i |w_i| \tag{2.13}$$

Then, there exists the **L2** regularization, which, instead, promotes the use of weights with small values by summing the sum of the squares of the weights themselves:

$$J_{\mathrm{L2}} = J + \sum_i w_i^2 \tag{2.14}$$

Finally, it is possible to combine the two techniques with the **Elastic net** regularization:

$$J_{\mathrm{EN}} = J + \lambda_1 \sum_i |w_i| + \lambda_2 \sum_i w_i^2 \tag{2.15}$$

where λ_1 and λ_2 are two coefficients that can be fine-tuned to give more weight either to the L1 or L2 terms.

Then, it is possible to employ the **Dropout** technique. During the training of the NN, at each epoch, a subset of the neurons of the network is randomly deactivated. This is shown to be useful because it forces the neurons to be independent from the others, without relying on them, in order to achieve the best possible output. In this way, the ANs are capable of learning more general patterns, because they have to deal with randomly deactivated pathways, acting similarly to noise. Moreover, since during training some of the neurons are not employed, this means that we are not using the NN at full capacity, effectively mitigating the risk of overfitting due to the complexity of the model. Dropout is applied only during training, whereas it is used at full capacity during testing or inference.

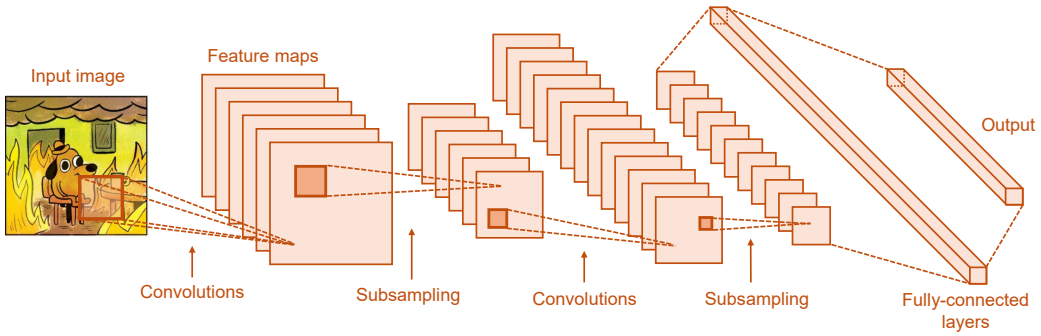

Figure 2.6 Schematic of a typical CNN.

Finally, the use of **mini-batching** can (indirectly) help with overfitting. Indeed, the stochasticity introduced by the random selection of the samples that make up the mini-batches, which change at every epoch, introduces statistical noise in the training procedure, preventing the model from memorizing the training data and enabling better generalization.

2.1.4 Convolutional Neural Networks

Up to now, we have only considered FFNNs, where the layers are said to be "fully connected", meaning that each neuron of the current layer has a weighted connection with all the neurons from the previous layer. However, in recent years, more complex ANN architectures have been garnering popularity for specialized applications. The prime example are Convolutional Neural Networks, which are very effective in computer vision applications (e.g., image and video processing) or, in general, whenever the NN has to process a grid-like structure (images, audio, time-series, etc.) [15]. The schematic structure of a typical CNN is reported in Fig. 2.6.

The fundamental component of a CNN is the **convolutional layer**, which performs a convolution between the input data and a weight filter (called "kernel"), outputting a "feature map":

$$h_i^{l+1} = W_i * h^l + b \tag{2.16}$$

where h_i^{l+1} is the output feature map for the i-th kernel, W_i is the i-th kernel, $*$ is the convolution operator, h^l is the input, and b is the bias. Typically, the input is not one-dimensional, so the CNN will have multiple channels working in parallel

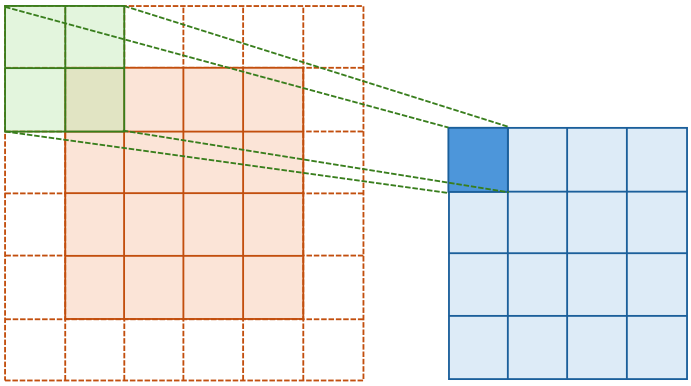

Figure 2.7 Representation of a convolution between a 4×4 input grid (orange) and a 2×2 kernel (green), resulting in a 4×4 output feature map (blue). The input grid has padding to obtain the correct dimensions for the output map.

(e.g., one for each color of an RGB picture). Basically, the kernel will scan the input grid and create a map of the response of the input to the chosen kernel. A representation of this scanning operation is reported in Fig. 2.7, where we have a convolution between a 4×4 input (orange) and a 2×2 kernel (green).

One of the most important properties of CNNs is **weight reuse**: since the weight kernel scans the input on all its spatial dimensions, its weights will be reused in multiple different spatial locations. Compared to fully connected FFNNs, for which each neuron has its own weight, in the case of CNNs, we have a lower number of parameters to be trained, resulting in a much better training efficiency.

From Fig. 2.6, it is also possible to appreciate that not all layers perform convolutions, but some of them are dedicated to subsampling the image. The idea is to summarize the features of the image layer by layer, until reaching a single scalar value that could be employed for classification. This is accomplished with the use of **pooling** layers: a subset of the image is considered and a given mathematical operation is applied to get a single value for the subsampled image, then we move by one pixel until the entire original grid has been subsampled. The most common pooling operations are **max pooling** (only the maximum value of the subset ends up in the subsampled image) or **average pooling** (the average of the subset ends up in the subsampled image). Another common pooling technique is **striding**, where the pooling filter is shifted by more than one pixel at a time (by a quantity of pixels called “stride”).

Finally, many CNNs include some fully connected layers at the end of the architecture, where the classification will be performed. In these layers, we can no longer exploit the weight reuse property, but, due to the subsampling performed before, the fully connected layers can be smaller than in a traditional FFNN architecture.

2.2 Photonic neuromorphic implementations

As explained in the previous section, AI and ML are data-driven applications, which makes them extremely computationally-intensive: in order to effectively train a NN, the network must process a vast amount of data samples, even in the order of billions, depending on the application. For instance, an LLM like OpenAI's GPT3 was trained on a dataset of 570 GB (45 TB of raw dataset before filtering) [20]. For this reason, the computational power required to train state-of-the-art AI has been doubling every 3.5 months [21], as shown graphically in Fig. 2.8. However, conventional computers are expected to become a bottleneck for further advances in NN training for two main reasons: first, progress in high-density integration has slowed with respect to Moore's law [22] since transistors are approaching their physical size limits, for which quantum tunneling effects hinder their operation [23]; second, the Von Neumann architecture of electronic computers, where memory and processing unit are physically separated [24], is inefficient in handling Multiply-and-Accumulate (MAC) operations, which are central to NN training [25] and for which many memory accesses are needed.

This is why it is of paramount importance to develop alternative hardware accelerators able to implement MAC operations efficiently. These could facilitate the training on even larger datasets, thus enabling next-generation AI models. For this purpose, both electrical and photonic hardware accelerators have been investigated. The first approach includes the exploitation of readily available resources (for instance, exploiting the large number of cores in a Graphical Processing Unit (GPU) [27]) or the development of new specialized hardware platforms that perform NN computations more efficiently (for instance, Field-Programmable Gate Arrays (FPGAs) [28] or Application-Specific Integrated Circuits (ASICs) [29]).

On the other hand, it is also possible to create photonic-based circuits for ML computations (often referred to as "photonic neuromorphic circuits"). Despite the disadvantage of being a less mature technology with respect to traditional CMOS-

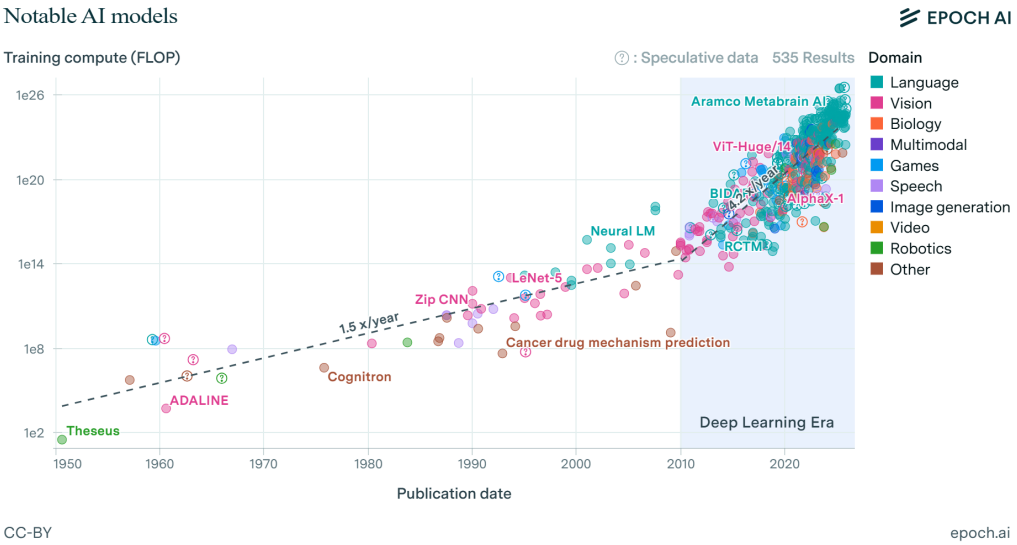

Figure 2.8 Graphical representation of the evolution of the training compute, i.e. the total computational power required to train a neural network, measured in Floating-Point Operations (FLOPs), including the most notable AI models [26]. The dashed black line represents a linear regression of the FLOPs over the years, showing that the number of FLOPs has been increasing 4.2 times a year in the DL era of AI.

based analog integrated electronics, photonic neuromorphic circuits could be beneficial for a series of inherent advantages of PICs, such as increased parallelism when exploiting Wavelength Division Multiplexing (WDM), Polarization Division Multiplexing (PDM), and Mode Division Multiplexing (MDM), reduced power consumption, ultra-high operating speeds, and compatibility with the silicon industry [30–33]. These advantages could potentially give an edge to photonic-based solutions with respect to analog electronic accelerators.

Among the photonic hardware accelerators for neuromorphic computing, we have multiple possible implementations based on widely different working principles and with varying degrees of biological brain inspiration. Among these, we have Reservoir Computing (RC), Spiking Neural Network (SNN), and Fully Connected Neural Network (FCNN).

In the next subsections, the details regarding these possible solutions will be discussed.

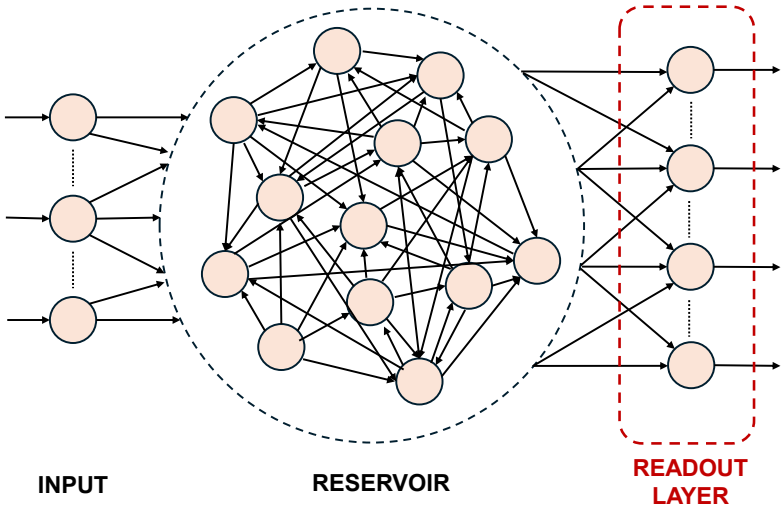

Figure 2.9 Schematic representation of a Reservoir Computing NN. The input layer and reservoir have fixed weights, while the readout output layer is trainable to meet application-specific requirements.

2.2.1 Photonic Reservoir Computing

RC is a ML paradigm that is well suited for learning time-dependent features [34, 35] and that is garnering increasing attention in the context of photonic neuromorphic computing thanks to its compatibility with optics [36]. Rather than employing multiple independently-trainable neurons, RC uses a set of layers with fixed random weights (i.e. the so-called "reservoir"), while only a limited number of output layers (called "readout layers") are trained [30, 36], usually with a linear regression. A graphical representation of the structure of an RC NN is reported in Fig. 2.9. For this reason, RC is beneficial from a practical standpoint because it reduces the requirements for tunable elements implementing the trainable ANs and, consequently, power consumption, training time, and training procedure complexity [37]. Another advantage of this approach is that the same reservoir structure can be used for multiple tasks, by retraining the application-specific readout layers [30], with a similar approach to transfer learning in traditional ML [38].

To achieve good performance, the effect of the initial conditions on the reservoir state must vanish over time, ensuring that the state is driven mainly by the input sequence rather than the initial state of the network [35]. An RC NN that satisfies this condition is said to possess the Echo State Property (ESP) [39]. Basically, the

RC NN must have both a sufficiently rich nonlinear transformation of the input and a suitable degree of fading memory, so that past inputs are stored for a limited time, while older information progressively dissipates [40]. When ESP holds, the readout layer can reliably extract task-relevant features from the evolving reservoir states using only a linear combination of them. As a consequence, the design of photonic reservoirs typically aims at operating the system in a regime where the best nonlinearity-memory trade-off is achieved, avoiding regions that are too linear (insufficient separability) or too chaotic (poor trainability) [41].

Within the context of photonic RC, different implementation solutions are possible. The first kind of possible implementations consists in the use of reservoirs made with discrete optical nodes [42]. These discrete nodes can be implemented, among the other instances, with Micro Ring Resonator (MRR), fully-passive photonic elements [43], or Vertical-Cavity Surface-Emitting Laser (VCSEL) arrays [44]. In these architectures, the interconnections between nodes (e.g., couplers, waveguide lattices, interferometric meshes) form a fixed network that naturally introduces mixing and recurrence between the optical signals. Depending on the specific platform, the nonlinearity can be provided by the optical component itself (e.g., laser saturation, Kerr effects) or by the photodetection stage, where the quadratic response of the photodiode supplies the nonlinear activation [42]. These reservoirs with discrete nodes are particularly appealing in the context PICs because they can take advantage of optical parallelism [35], with all nodes evolving simultaneously, potentially allowing very high throughput operation.

Another compelling idea consists in using Time Division Multiplexing (TDM), where a single node is implemented, but multiple "virtual nodes" are accessed with the use of delay lines [37]. With this strategy, the implementation complexity is significantly reduced, but, due to the presence of the delay lines for TDM, the processing speed is slowed down as well [37]. This approach has been demonstrated in both fiber-based [45] and on-chip implementations [37], with the advantage of reducing the footprint of the NN. Despite the constraints related to the trade-off between processing speed and processing capabilities [46], delay-based reservoirs have achieved compelling results in prediction, classification, and signal-processing tasks [42], confirming their practical relevance.

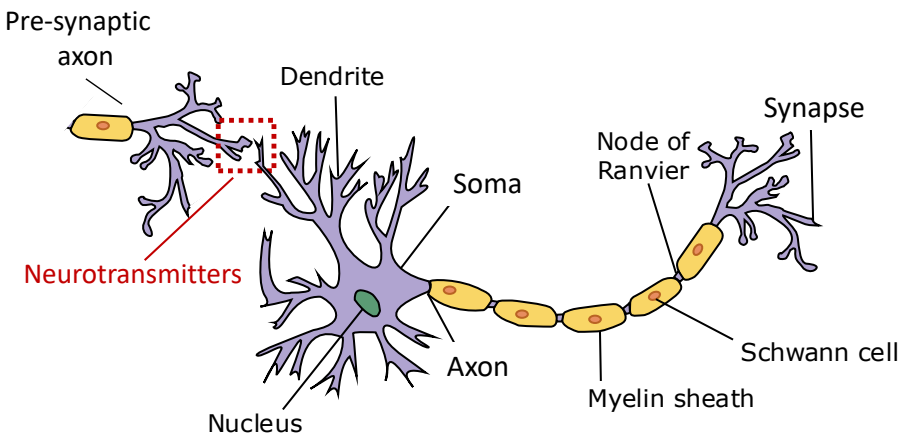

Figure 2.10 Schematic of a biological neuron. The "dendrites" represent the input of the neuron, where the signals from other neurons are collected; the "soma" is the main body of the cell, where the input signals are integrated and where it is established if the neuron will fire; the "axon" is the channel where the signals of the neuron are conveyed to other neurons of the network. It is usually covered by "myelin sheaths" interrupted by "nodes of Ranvier" for faster propagation of the action potential; the "synapses" are the interfaces with the dendrites of the neighboring neurons, where the electrical and chemical signals are exchanged via some "neurotransmitters". Image adapted from [47].

2.2.2 Photonic Spiking neural networks

The main idea behind SNN consists in developing NNs with ANs that mimic the biological behavior of natural neurons, bridging the gap between neurosciences and ML [48]. This could be particularly beneficial to exploit the innate efficiency of biological brains, thus reducing power consumption for training of NNs in traditional DL. The basic mechanism at the foundation of SNNs is the encoding of information on discrete signals (called "spikes") that work in the continuous time domain [49, 48], making SNNs well suited for handling temporal code.

In order to understand the operation of SNNs, it is important to first understand the basic working principles of biological neurons. A graphical representation of a natural neuron is contained in Fig. 2.10. A brain cell is formed by a main portion called "soma", where the nucleus is contained, where the input signals are integrated, and where it is established whether the neuron will fire or not. The signals reach the soma through short neural terminations called "dendrites", acting as inputs of the neuron. The "axon" is a long neural fiber that acts as a channel to transport the action potential towards other neurons. Most axon types are covered with the

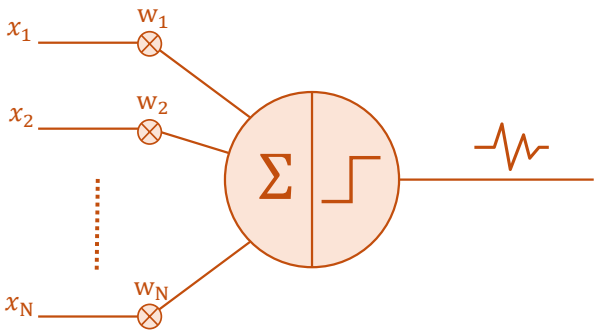

Figure 2.11 Schematic of an LIF spiking neuron. The inputs are weighted and integrated until a given threshold is reached: at this point, a spike is emitted.

"myelin sheath", an insulating lipidic material that allows faster propagation of the action potential along the axon itself. Finally, the "synapse" is the interface between the neuron and the next cells, allowing the exchange of potential either electrically (fast, but no signal amplification) or chemically (regulated by "neurotransmitter" compounds, much slower, but with signal amplification) [48].

Concerning the operation of the neuron, instead of using the most biologically-accurate Hodgkin-Huxley model, it is possible to accurately describe its behavior with a Leaky Integrate and Fire (LIF) model. The soma integrates the weighted signals coming from the dendrites over time, increasing the membrane voltage until a given threshold is reached: at that point, a spike of action voltage is emitted along the axon [50]. After this spike, the membrane voltage remains at a reset value for a given time (called "refractory time"), which effectively limits the operational speed of the neuron [48]. The model is said to be "leaky" because the membrane voltage tends to decrease because of ionic leakage if not pumped. This working principle can be implemented to create an artificial spiking neuron (a schematic is reported in Fig. 2.11).

Of course, in order to have effective operation of an SNN, we have to encode information in the spikes provided at the input of the network itself. For this purpose, two main strategies can be adopted [48]: on one hand, we have the "rate encoding", where the information is encoded with the average number of spikes n over a given time window T, with which the firing rate can be computed as $r = n/T$ [48]. For instance, if we want to encode an image, we can match the color value of each pixel with the corresponding firing rate. On the other hand, with the "temporal encoding", the information is not encoded with the number of spikes, but with their timing,

thus obtaining much sparser spike firing rates, even though this method is more susceptible to temporal jitter [48]. In this case, in order to encode an image, the pixel value is normalized and the spike at time t will correspond to the pixel with a normalized value equal to t. In general, the encoding is based on the occurrence of events in time [50].

It is possible to implement photonic hardware capable of generating spikes for the creation of photonic SNNs, attempting to combine the natural efficiency of biological brains and the practical advantages of photonic devices. Both non-integrated and integrated solutions have been proposed [23]. For instance, in Reference [51], the use of a discrete VCSEL combined with a Saturable Absorber (SA) is used to emulate the LIF neuron. The idea of creating an artificial LIF neuron is reiterated in Reference [24] with Waveguides (WGs) and MRRs, integrated with Phase Change Material (PCM) cells. Signals at different wavelengths are weighted by different PCM-loaded WGs before being summed by an MRR-based WDM multiplexer onto a single WG, which inputs the combined spikes into a PCM neuronal cell crossing an MRR. If the combined spikes have enough power to switch the PCM to the amorphous state, this will detune the MRR resonance so that a probe pulse is transmitted as an output spike past the MRR. One last example is [52], where the nonlinear optical effects of a single MRR are exploited to create a reliable spike generation paradigm.

2.2.3 Photonic Fully connected neural networks

The idea behind photonic FCNNs is the implementation of an AN similar to the ones used in traditional ML. For this reason, as in an AN, there will be a portion of the neuron dedicated to the weighted linear combination of the inputs and a nonlinear activation function that nonlinearly modifies the output based on the power levels of the linear neuron (similarly to the threshold behavior of the spiking neurons described in Subsection 2.2.2). As mentioned before, like traditional ML and DL NNs, this strategy is less biologically plausible with respect to RC NNs or SNNs, thus lacking their natural efficiency, but it offers the advantage of well-established ML techniques, in conjunction with the inherent advantages of photonic technology that partially solve the inefficiency of traditional AI.

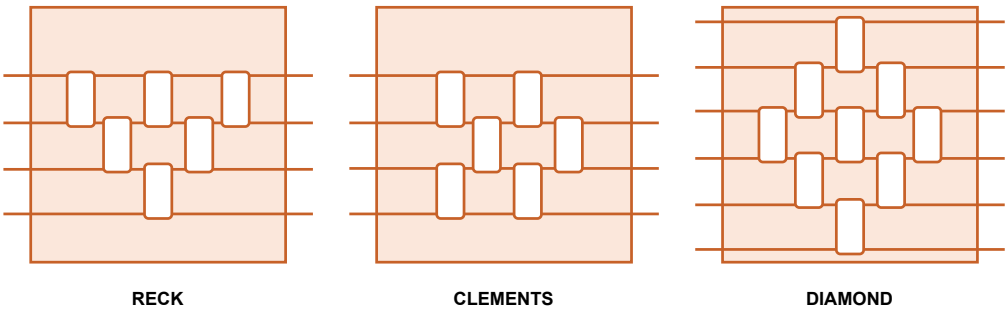

Figure 2.12 Graphical representation of the three main linear neuron implementations (Reck [53], Clements [54], and Diamond [55]). Each one of the white boxes represents a programmable switching node (e.g., MZI).

Linear neuron implementations

In order to implement the linear part of the neuron, it is possible to employ the canonical architectures employed for mesh-based linear programmable photonic processors that have been shown to enable the implementation of any unitary transformation [56]. These architectures usually rely on meshes of tunable elements, primarily Mach-Zehnder Interferometers (MZIs). The most used implementations are the triangular Reck [53], the square Clements [54], and Diamond [55], which is basically a mirrored Reck architecture. A schematic for each one of these architectures is reported in Fig. 2.12, where each box represents a basic processing unit (e.g., MZI). The **Reck** architecture with N inputs and N outputs is characterized by a triangular layout with $N(N-1)/2$ processing cells. It is characterized by an "optical depth" (i.e. the maximum number of processing cells to be crossed to connect any input with any output) of $2N-3$ [31]. It has the disadvantage of having unbalanced optical paths, so different numbers of devices will be crossed depending on the input-output pair.

The **Clements** architecture, instead, relies upon a rectangular layout with the same number of devices as Reck. The Clements implementation is instead characterized by balanced optical paths and by a shorter optical depth of N, resulting in lower Insertion Loss (IL) [31]. Due to this feature, compared to Reck, the Clements architecture also results in being more robust with respect to fabrication imperfections and noise, but it has the disadvantage of being harder to calibrate and monitor [57].

Finally, it is possible to implement the **Diamond** mesh [55] which is characterized by $(N-1)^2$ devices. Despite having the disadvantage of having a larger footprint

Table 2.1 Summary of the main features of the three standard linear photonic processor architectures, highlighting their advantages and disadvantages.

Mesh	# MZIs	Depth	Pros	Cons
Reck	$\frac{N(N-1)}{2}$	$2N-3$	Smaller footprint Simple calibration and monitoring	Unbalanced optical paths High IL Susceptible to noise and fabrication Limited scalability
Clements	$\frac{N(N-1)}{2}$	N	Smaller footprint Robust to noise and fabrication Balanced optical paths Low IL	Non-trivial monitoring Limited scalability
Diamond	$(N-1)^2$	$2N-3$	Robust to noise and fabrication Simple calibration and monitoring	Larger footprint Unbalanced optical paths High IL Limited scalability

and the same optical depth as the Reck architecture due to the unbalanced optical paths, it offers robustness to noise and defects like Clements and simple monitoring like the triangular Reck architecture. All three architectures are affected by limited scalability due to the fact that the number of tunable elements scales as N^2 [31], thus making the control of larger meshes extremely difficult from the practical point of view. Other architectures (e.g., Bokun architecture [57]) have been proposed to overcome the trade-off between Reck's and Clements' advantages and disadvantages, although the problem of scalability cannot yet be solved with the current maturity of photonic technology. Table 2.1 summarizes the main advantages and disadvantages of the three architectures discussed.

Since in traditional NNs we have to adjust the weights and biases to achieve learning, in the case of FCNNs, we want to have the possibility of performing the same operation with the photonic hardware. For this reason, programmable devices such as thermally-tuned MZIs are a popular choice, since it is possible to control the propagation of the field by modifying the relative phase difference between the two arms of the MZI itself (e.g., by means of thermal phase shifters, but also exploiting mechanical or electro-optical effects [56]).

In order to program the phase shifts of the MZI mesh to achieve learning for a given task, different strategies can be employed [56]: each phase can be programmed separately, element-wise; a matrix decomposition could be used as proposed by [54] in order to find all the phases for all the MZIs in one step; an optimization routine could be executed to find the phases for the Thermo-Optic Phase Shifter (TOPS). Moreover, we can make the distinction between "online" (*in situ*) and "offline" (*ex*

situ) training. The first consists in finding the proper weights of the mesh directly on the hardware (e.g., with an optimizer or a backpropagation algorithm), while the latter consists in doing the same operation on a traditional computer, applying the found weights on the photonic platform *a posteriori*. The advantages and disadvantages of these two strategies will be discussed in detail in the next chapter, in Section 3.6.

The final element needed to achieve learning like in traditional ML is the implementation of a Nonlinear Activation Function (NLAF).

The challenge of all-optical nonlinear activation functions

The task of implementing a NLAF on a PIC is non-trivial: most photonic components in Si PIC (WGs, splitters, MZIs, etc.) are characterized by naturally linear operation with respect to input power, if the power levels are low enough so that third-order nonlinearities are not triggered. For this reason, for the purpose of neuromorphic computing applications, many studies either apply the NLAF on a standard computer [30] or exploit the quadratic nonlinearity introduced at the output layer by intensity-based detection with a Photodiode (PD) [30, 58]. However, to enrich the nonlinear response of the Photonic Neural Network (PNN), it would be beneficial to have a NLAF after each layer of linear neurons, not only at the output.

For this purpose, multiple solutions for integrated NLAF rely on Electro-Optic (EO) conversions, especially if programmability of the response is required [59, 60]. For instance, in [60], the NLAF is implemented by routing a small portion of the output power of the linear neuron to a PD via a Direction Coupler (DC). The photogenerated voltage is combined with a static external voltage and then used to pilot an MZI to apply the nonlinearity. The problem with these solutions is that they require additional power consumption for the EO conversion and co-integration with CMOS technology, which increases the circuit complexity [61].

The alternative is the development of all-optical NLAFs, thus avoiding power inefficient EO conversions and potentially harnessing all the advantages of photonic platforms. Many different implementations of all-optical NLAFs have been proposed since the inception of PNNs. There exist multiple solutions in free space [72] and ones with discrete optical components [73], for instance, exploiting the nonlinear properties of materials such as $LiNbO_3$, which provide strong nonlinear Pockels effect [74]; however, in this work, the focus will be placed on all-optical integrated

Table 2.2 Comparison of the footprint and optical threshold of various NLAF implementations from literature.

Ref.	Device	Footprint (μm^2)	Optical threshold (dBm)
[62]	Ge/Si MRR	1.29	-1.31
[24]	PCM cell in MRR	100	3.62
[63]	Graphene layer on WG	400	10.0
[64]	Ge/Si PD	240	0.41
[65]	Graphene/Si heterojunction in MRR	80.0	-3.00
[66]	$MoTe_2$ integrated in WG	50.0	-23.1
[67]	Ge/Si asymmetric coupler	70.0	7.08
[60]	Si EO MZI	100000	7.00
[61]	Si MZI + RAMZI	7500	5.00
[68]	Si photonic crystal Fano laser	100	-3.00
[69]	Si add-drop MRR	100	-10.96
[70]	Thin-film $LiNbO_3$	4250.0	-24.0
[71]	IMOS MZI + RAMZI	54400	-2.5

implementations, which offer all the practical benefits of PICs, together with the advantage of integrating the NLAF after each layer of integrated photonic ANs, for a richer response.

For instance, in Reference [61], an integrated programmable all-optical NLAF on Silicon-on-Insulator (SOI) is proposed. The structure is made of an MZI cascaded with a Ring-Assisted Mach-Zehnder Interferometer (RAMZI), which is a traditional MZI with an MRR nested on the upper arm of the interferometer. The two MZIs are thermally controlled via metallic TOPSs and provide the programmability of the response by controlling the amount of light that enters the MRR, which provides the nonlinear response. In particular, in SOI platforms, the main nonlinear effect is Free Carrier Dispersion (FCD) and this implementation is capable of providing different nonlinear P_{in}-P_{out} responses, depending on the splitting ratio of the MZIs. Reference [61] shows the capability of the device to implement Sigmoid, Clamped ReLU, Softplus (akin to a Leaky ReLU function), and Radial Basis Function (RBF) functions with different combinations of driving voltages of the TOPSs.

In Reference [71], a similar structure is implemented on the Indium Phosphide Membrane on Silicon (IMOS) platform (Fig. 2.13), showing the capability of the device to implement a Softplus or Leaky ReLU-like response without any power consumption, if the input laser wavelength is close to the MRR resonance. Additional nonlinear shapes can be obtained either by biasing the ultra-compact TOPS present on the MZI and on the MRR or by acting on the detuning of the input laser light

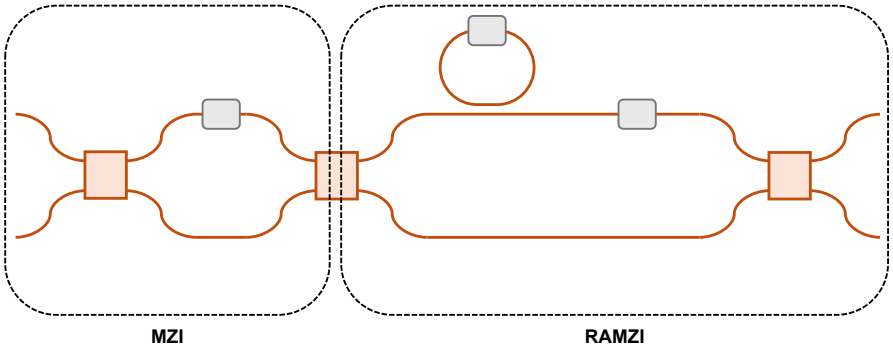

Figure 2.13 Schematic of the NLAF structure of [71], realized with cascaded MZI and RAMZI. The orange rectangles represent the MMIs of the MZI, while the gray ones represent the ultra-compact TOPSs.

with respect to the MRR resonance, for an additional degree of freedom with low power consumption. Contrary to the SOI-based solution of [61], in this case, thanks to the nonlinear properties of InP, other effects such as Optical Kerr Effect (OKE) and Self-Heating (SH) effects are more easily harnessed.

In Table 2.2 we report a comparison of some of the NLAF implementations that can be found in the literature, analyzing their footprint and their optical threshold.

2.3 Non-neuromorphic programmable photonic platforms

Instead of designing photonic ASICs such as the ones that we analyzed up to now, a very appealing idea consists in the development of multipurpose programmable PICs that could be employed similarly to electronic FPGAs. This idea has the potential of harnessing the advantages of electronic FPGAs over ASICs (e.g., reconfigurability, reduced engineering costs, faster deployment, etc.), making it a very useful instrument for fast prototyping at the research level.

One of the most prominent examples is the Silicon photonic multipurpose processor core proposed in Reference [75], based on a hexagonal waveguide mesh, with basic processing units implemented with thermally-tuned MZIs (see Fig. 2.14). This photonic processor was patented and commercialized under the name iPronics SmartLight.

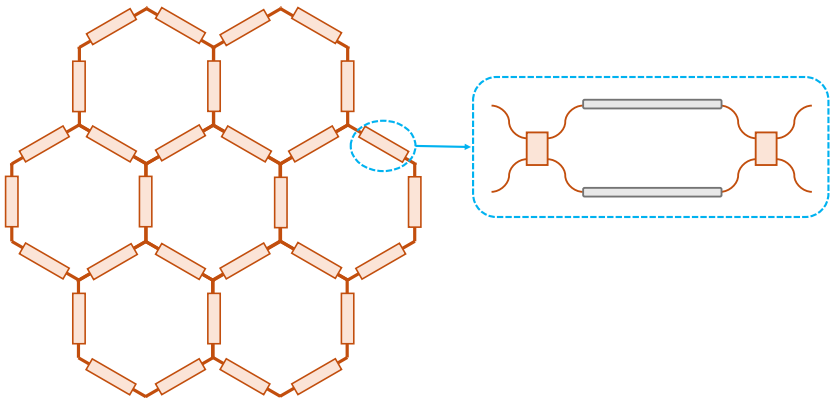

Figure 2.14 Schematic of the hexagonal mesh proposed in [75]. Each orange rectangle inside the hexagons represents a basic programming unit, created with a thermally-tuned MZI (represented in the inset).

In [75], the structure represented in Fig. 2.14 is shown to be capable of implementing multiple different configurations reproducing commonly used photonic components, including Finite Inpulse Response (FIR) filters, Optical Ring Resonators (ORRs), and Coupled Resonator Optical Waveguides (CROWs), but also linear matrix transformations, such as Pauli matrices and a C-NOT gate.

In general, this kind of photonic processor core, included in the iPronics platform, has garnered popularity for initial fast prototyping and/or proof of concept at the academic research level [76], because it allows the demonstration of the proposed components or functionalities, without requiring the development of expensive *ad-hoc* PICs. It is possible to envision the use of such FPGA-like devices in the context of neuromorphic computing with a strategy similar to the photonic FCNNs described in Subsection 2.2.3, due to their capability of implementing arbitrary matrix transformations [75].

2.4 Summary

In this chapter, we conducted a review of the state-of-the-art of photonic solutions for neuromorphic computing. First, we started by analyzing the basics of traditional ML, starting from the simple MLP and we finished with the description of CNNs, a complex architecture that is most commonly used for image recognition tasks. Training techniques have been thoroughly discussed as well.

After this general discussion to understand the bottlenecks of traditional ML, we moved on with the review of the main photonic neuromorphic implementations, characterized by varying degrees of biological inspiration. In particular, we have focused on reservoir computing, spiking neural networks, and fully-connected PNNs, explaining their basic operating principles and some of their practical implementations. In this context, we have also dealt with the problem of implementing all-optical nonlinear activation functions that could be used for photonic neuromorphic computing.

Finally, photonic FPGAs have been briefly mentioned due to their capability of implementing arbitrary matrix transformations, which could enable their use in the context of photonic neuromorphic computing.

In the next chapter, we will describe the circuit-level model that we have developed for the description of MZI-based linear neurons akin to those described in Subsection 2.2.3, affected by the thermal crosstalk arising from the use of multiple thermally-controlled MZIs at the same time.

Chapter 3

Circuital model for MZI-based linear neurons

As mentioned in Subsection 2.2.3, one of the possible solutions for the implementation of a photonic AN consists in creating an FCNN made up of a linear circuit performing the weighted linear combination of the input signals and a nonlinear activation function. In this chapter, we will analyze the physical behavior of a linear neuron implemented with interconnected thermally-controlled MZIs, developing a model capturing both the light propagation through the structure and the spurious thermal interactions between neighboring devices. These interactions, also called "thermal crosstalk", represent a critical effect in thermally controlled linear neurons, because they can strongly hinder the accuracy of the PNN during an "online" training procedure (the PNN weights are adjusted directly on-chip) [77, 78] and they are one of the main limitations to the integration of devices in large numbers [79]. Therefore, it is important to develop a physical understanding of thermal crosstalk to be able to counteract it and achieve high accuracy while training the PNN. The presented model is validated by direct comparison with the measurements of a linear mesh of 9 interconnected thermally-controlled MZIs, developed at the Photonic Department of the Denmark Technical University (DTU).

In this chapter, first, the reference circuit for the validation of the model and its technological platform will be introduced; then, the light propagation model for the single MZI and other circuital elements will be introduced; next, the thermal crosstalk model will be explained, followed by the results obtained during the validation of

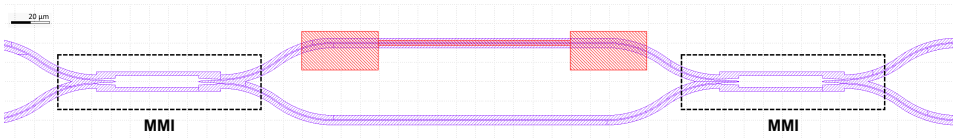

Figure 3.1 Mask of a single MZI from the reference circuit. The input and output 2×2 MMIs are indicated by the black dashed rectangles. The Ti strip used for thermal tuning, together with its electrical pads (shown in red), covers the central section of the upper waveguide. The grid size is $10\,\mu m \times 10\,\mu m$ and the total area of the depicted region is approximately $500\,\mu m \times 65\,\mu m$.

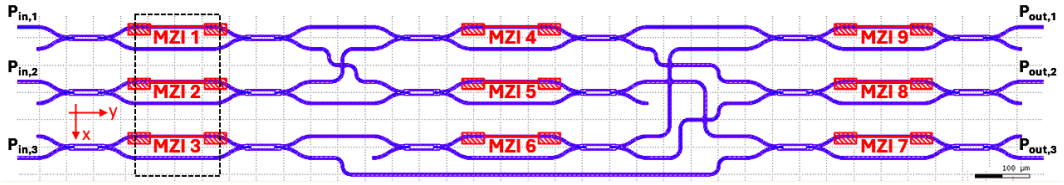

Figure 3.2 Mask of the reference circuit, composed of nine interconnected MZIs, each with the same design as the one shown in Fig. 3.1. The waveguide structures are depicted in blue, and the Ti heaters in red. The grid size is $50\,\mu m \times 50\,\mu m$ and the total area of the displayed region is approximately $1900\,\mu m \times 320\,\mu m$. The black dashed rectangle marks the area where the thermal analysis is performed.

the model; finally, the validated model is employed as a "digital twin" of the real device, showcasing the possibility of employing the simple 3×3 MZI mesh as a programmable photonic Boolean programmable gate, able to implement user-defined logic functions on a large wavelength range.

The content of this chapter is based on the published works of References [80–83].

3.1 Reference circuit and technology

The reference circuit, whose mask is reported in Fig. 3.2, is a 3×3 mesh of MZIs with incomplete interconnections, operated with three input ports and three output ports. The fundamental building block of this PIC is the 2-by-2 MZI reported in Fig. 3.1, which includes two Multi-Mode Interferometers (MMIs) in input and output, connected by two $\sim 267\,\mu m$ long waveguides. These MZIs can be switched from cross to bar state by thermally modifying the effective refractive index of their waveguides by operating the TOPS located on the upper arm. The TOPS is a $100\,\mu m$

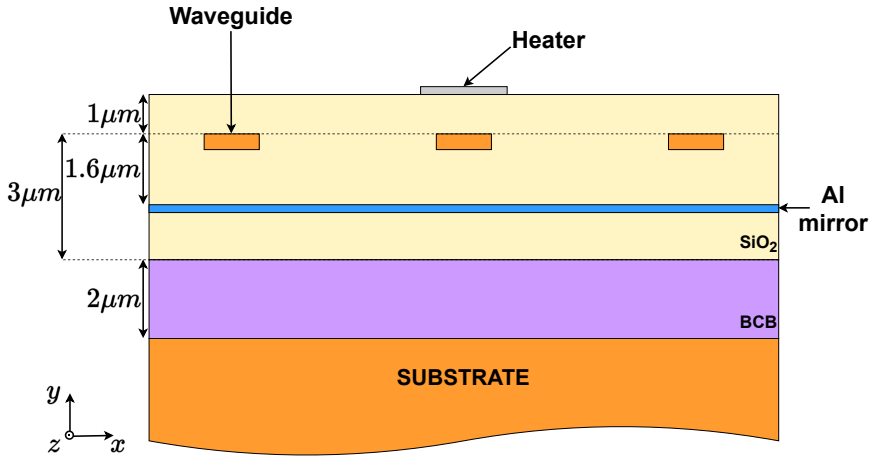

Figure 3.3 Schematic of the vertical cross-section of a portion of the PIC. Each color is associated with a different material: Si in orange, Benzocyclobutene (BCB) in lilac, SiO_2 in yellow, Al in blue, and Ti in gray.

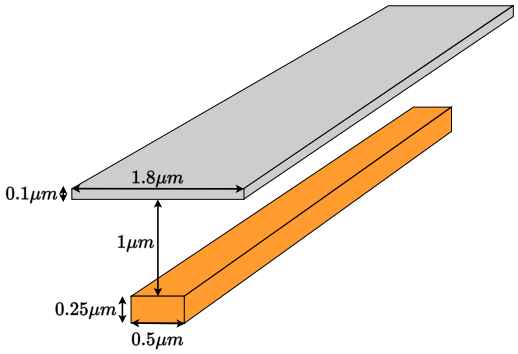

Figure 3.4 Geometry of the Si waveguides (orange) and Ti TOPS (grey). The latter is separated from the waveguides underneath via a 1 µm thick layer of SiO_2.

Titanium strip that can be biased with an applied voltage to exploit the Joule heating effect.

This reference circuit is part of a larger PIC developed at the Denmark Technical University as a 7×7 reconfigurable optical switch in C-band [84]. This PIC is fabricated on an SOI wafer with a buried Aluminum mirror, realized via flip-chip bonding [84]. A schematic of the technological platform on which this PIC is implemented is reported in Fig. 3.3 and the detailed fabrication procedure can be found in [84].

Table 3.1 Design lengths of the interconnections between pairs of MZIs.

Connection	Length (µm)	Connection	Length (µm)
$1 \rightarrow 4$	235	$4 \rightarrow 9$	305
$1 \rightarrow 5$	248	$4 \rightarrow 8$	319
$2 \rightarrow 4$	263	$5 \rightarrow 7$	338
$2 \rightarrow 5$	235	$6 \rightarrow 8$	358
$3 \rightarrow 6$	235	$6 \rightarrow 9$	418
$3 \rightarrow 7$	950	-	-

Figure 3.4 contains a schematic of the waveguide-microheater system, present in each MZI for thermal control. The Si waveguides and Ti heaters have $0.5\,\mu m \times 0.25\,\mu m$ and $1.8\,\mu m \times 0.1\,\mu m$ cross-sections, respectively. Each Ti heater is $100\,\mu m$ long and has two low-resistance electrical pads with $40\,\mu m \times 20\,\mu m$ area, in order to bias the Ti strip, localizing the heating in correspondence of the waveguide. These Ti microheaters show, on average, rise and fall times of $66\,\mu s$ and $27\,\mu s$, corresponding to relatively slow switching speeds [84]. Although only steady-state simulations will be performed for this analysis, it has to be taken into account that the microheater switching speed will limit the training (or programming) speed of the mesh in an experimental setting. Both Fig. 3.3 and Fig. 3.4 show that between waveguides and heaters there is a thin 1 µm layer of SiO_2: this has the purpose of reducing the metal absorption losses caused by the proximity between the Si structures and the metal plates.

Note that all the devices of the reference circuit are designed to operate in cross state at 0 V and to switch to a complete bar state when 2 V are applied to the TOPS. However, it was experimentally observed that these MZIs have slightly different initial phase shifts at 0 V and require different applied voltages to achieve a complete transition to bar state. This is caused by fabrication variation and defects [85], which affect the MZI working points randomly. In the most extreme cases, we have that MZIs 6 and 8 exhibit the opposite behavior (bar state without bias and cross state when 2 V was applied).

As shown in Fig. 3.2, the nine MZIs of the circuit are interconnected by means of a series of bent waveguides and optical crossings. For an accurate description of the phase change and the losses accumulated between consecutive MZIs, the length of each optical connection must be considered (listed in Table 3.1). Other circuit

elements (e.g., waveguide crossings) are taken into account by means of their IL, estimated in Section 3.3.

3.2 Single MZI model

In this section, a physical model for the stand-alone MZIs and their optical circuit-level interaction will be developed. This will be done employing the geometry and estimated losses of the reference circuit described in Section 3.1, but this methodology is valid for any MZI-based meshed topology, if its low-level components are properly taken into account.

In order to describe light propagation through a mesh of MZIs, we first require a reliable model for the physical behavior of the single device. For this reason, we develop a general transmission matrix-based description of each constitutive element of an MZI (two MMIs and the interferometric section), employing the results of Coupled Mode Theory (CMT) [86]:

$$T_{\mathrm{MMI_{in}}} = \begin{bmatrix} \alpha_{\mathrm{MMI_{in}}}\sqrt{\gamma_{\mathrm{in}1}} & j\alpha_{\mathrm{MMI_{in}}}\sqrt{1-\gamma_{\mathrm{in}2}} \\ j\alpha_{\mathrm{MMI_{in}}}\sqrt{1-\gamma_{\mathrm{in}1}} & \alpha_{\mathrm{MMI_{in}}}\sqrt{\gamma_{\mathrm{in}2}} \end{bmatrix} \tag{3.1}$$

$$T_{\mathrm{MMI_{out}}} = \begin{bmatrix} \alpha_{\mathrm{MMI_{out}}}\sqrt{\gamma_{\mathrm{out}1}} & j\alpha_{\mathrm{MMI_{out}}}\sqrt{1-\gamma_{\mathrm{out}2}} \\ j\alpha_{\mathrm{MMI_{out}}}\sqrt{1-\gamma_{\mathrm{out}1}} & \alpha_{\mathrm{MMI_{out}}}\sqrt{\gamma_{\mathrm{out}2}} \end{bmatrix} \tag{3.2}$$

$$T_{\mathrm{prop}} = \begin{bmatrix} \xi_{\mathrm{m}}\varepsilon_{+}e^{j(\frac{2\pi}{\lambda}n_{\mathrm{eff},1}(T)L_{\mathrm{h}}+\delta\varphi)} & 0 \\ 0 & \alpha_{\mathrm{b}}^{2}e^{-\alpha_{\mathrm{prop}}L}\varepsilon_{-}e^{j(\frac{2\pi}{\lambda}n_{\mathrm{eff},2}(T)L_{\mathrm{h}}-\delta\varphi)} \end{bmatrix} \tag{3.3}$$

Eq.(3.1) and (3.2) represent the transmission matrices for input and output MMIs, where $\alpha_{\mathrm{MMI_{in}}}$ and $\alpha_{\mathrm{MMI_{out}}}$ are insertion losses for input and output MMIs, respectively, $\gamma_{\mathrm{in}i}$ and $\gamma_{\mathrm{out}i}$ are the corresponding splitting ratios, defined as the ratio between the output power at port i and the total power at the two outputs of the MMI. Since input and output MMIs have equivalent designs, we assume that $\gamma_{\mathrm{in}i} = \gamma_{\mathrm{out}i} \triangleq \gamma_i$. For a shorter notation, we define four new quantities $\Gamma_{11} = \sqrt{\gamma_1}$, $\Gamma_{12} = \sqrt{1-\gamma_2}$, $\Gamma_{21} = \sqrt{1-\gamma_1}$, and $\Gamma_{22} = \sqrt{\gamma_2}$.

Instead, Eq. (3.3) describes light propagation through the two waveguides connecting the input and output MMIs and representing the interferometric section. This also accounts for the temperature dependence of the effective refractive index that will be fundamental to describe the effect of the Joule heating introduced by the Ti TOPSs. In this equation, we define $\xi_{\mathrm{m}} = \alpha_{\mathrm{b}}^2 \alpha_{\mathrm{m}} e^{-\alpha_{\mathrm{prop}} L}$, where α_{m} is a metal absorption factor, α_{b} is a bending radiation factor (which appear as a squared quantity since each arm includes two bends, as it can be appreciated from Fig. 3.2), α_{prop} are the propagation losses through the waveguide, L is the total length of the waveguide, L_{h} is the heater length, and λ is the signal wavelength. The terms $\varepsilon_{\pm} = e^{j(\frac{2\pi}{\lambda} n_{\mathrm{eff}_0}(L-L_{\mathrm{h}}) \pm \delta\varphi)}$, with n_{eff_0} effective refractive index at room temperature $T = T_0 = 293\,\mathrm{K}$ and $\delta\varphi$ empirical phase offset due to fabrication defects, introduce the optical phase accumulated in the portion of the arms that is not covered by the electrode.

As illustrated in Section 3.1, the Ti strips are physically separated from the Si waveguides with a SiO_2 buffer layer to prevent metal absorption losses. Nevertheless, Eq. (3.3) takes into account these losses through the α_{m} term in order to preserve the generality of the model. However, in order to simulate the reference device, it is assumed that $\alpha_{\mathrm{m}} = 1$.

As mentioned previously, the effective refractive index $n_{\mathrm{eff}}(T)$ of Si has a strong temperature dependence [87], which can be exploited to change the state of the MZIs via thermal tuning. Note that, despite the fact that the TOPS is located only on the upper arm of the interferometer (Fig. 3.1), the temperature dependence is present for the effective refractive index of both MZI waveguides. Indeed, due to the lack of insulation trenches between the two arms and their limited distance [84], the TOPS placed on top of the upper arm also affects the lower arm of the MZI, albeit with lower heating power. This effectively reduces the heating efficiency of the Ti heater, thus increasing the applied voltage required to obtain the desired phase shift. Besides this spurious self-heating affecting the single MZI, when a TOPS is biased, it will also have an effect on neighboring devices, thus reducing the overall heating efficiency of the PIC as a whole. These spurious thermal effects, in particular thermal crosstalk between neighboring MZIs, will be analyzed in Section 3.4.

Finally, with the inclusion of the phase offset $\delta\varphi$, Eq. (3.3) also allows us to take into account possible fabrication variations that cause the supposedly-identical MZI to operate in different working points, even without any applied voltage on their

microheater. This term, together with the unbalanced splitting ratios of the MMIs, is sufficient to model the left-over optical transmission on the opposite port that is observed when the MZIs should be in complete bar or cross state [80]. This is a simple, yet effective solution to model this effect, without requiring sophisticated models, such as the one presented in Reference [88], where optical crosstalk is statistically modeled as a random variable with a Gaussian distribution.

The total MZI transfer matrix can be computed by cascading the circuit elements constituting the MZI itself, with the multiplication of the matrices of Equations (3.1), (3.2), and (3.3) [89]:

$$T = T_{\mathrm{MMI_{out}}} T_{\mathrm{prop}} T_{\mathrm{MMI_{in}}} \tag{3.4}$$

The transmission matrix T of Eq. (3.4) can be employed in the computation of the two output field components by multiplying it by the two input field components.

$$\begin{bmatrix} E_1^{\mathrm{out}} \\ E_2^{\mathrm{out}} \end{bmatrix} = T \begin{bmatrix} E_1^{\mathrm{in}} \\ E_2^{\mathrm{in}} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} E_1^{\mathrm{in}} \\ E_2^{\mathrm{in}} \end{bmatrix} \tag{3.5}$$

With straightforward computations, it is possible to obtain an expanded formula for the field at the two outputs that can be efficiently evaluated numerically in MATLAB®.

$$E_1^{\mathrm{out}} = \xi_1 \Gamma_{11}^2 p_1 E_1^{\mathrm{in}} - \xi_2 \Gamma_{12} \Gamma_{21} p_2 E_1^{\mathrm{in}} + j\xi_1 \Gamma_{11} \Gamma_{12} p_1 E_2^{\mathrm{in}} + j\xi_2 \Gamma_{12} \Gamma_{22} p_2 E_2^{\mathrm{in}} \tag{3.6}$$

$$E_2^{\mathrm{out}} = j\xi_1 \Gamma_{21} \Gamma_{11} p_1 E_1^{\mathrm{in}} + j\xi_2 \Gamma_{22} \Gamma_{21} p_2 E_1^{\mathrm{in}} - \xi_1 \Gamma_{21} \Gamma_{11} p_1 E_2^{\mathrm{in}} + \xi_2 \Gamma_{22}^2 p_2 E_2^{\mathrm{in}} \tag{3.7}$$

where, for brevity's sake, the following quantities have been introduced:

$$\xi_1 = \alpha_{\mathrm{MMI_{in}}} \alpha_{\mathrm{MMI_{out}}} \alpha_{\mathrm{m}} \alpha_{\mathrm{b}}^2 e^{-\alpha_{\mathrm{prop}} L} \tag{3.8}$$

$$\xi_2 = \alpha_{\mathrm{MMI_{in}}} \alpha_{\mathrm{MMI_{out}}} \alpha_{\mathrm{b}}^2 e^{-\alpha_{\mathrm{prop}} L} \tag{3.9}$$

$$p_1 = e^{j(\frac{2\pi}{\lambda} n_{\mathrm{eff}}(T(V_{\mathrm{in}})) L_{\mathrm{h}} + \delta\varphi)} e^{j(\frac{2\pi}{\lambda} n_{\mathrm{eff}_0}(L - L_{\mathrm{h}}) + \delta\varphi)} \tag{3.10}$$

$$p_2 = e^{j(\frac{2\pi}{\lambda} n_{\mathrm{eff}_0} L - \delta\varphi)} \tag{3.11}$$

In an ideal situation, with the absence of metal absorption losses ($\alpha_{\mathrm{m}} = 1$) and with balanced MMIs splitting ratios ($\Gamma_{ij} = \sqrt{0.5}$, $i, j = 1, 2$), we would obtain that $\xi_1 = \xi_2$. In this case, by applying the prosthaphaeresis formulae, it would be possible to find a closed-form solution for Eq. (3.6) and (3.7) proportional to the cosine of $\Delta\phi$, namely the phase difference between the two MZI arms. Indeed, assuming the simplest case with $\alpha_{\mathrm{m}} = 1$, $\alpha_{\mathrm{b}} = 1$, $\alpha_{\mathrm{MMI_{in}}} = \alpha_{\mathrm{MMI_{out}}} = 1$, $\Gamma_{ij} = \sqrt{0.5}$, $i, j = 1, 2$, and assuming that the heaters affect the whole waveguides ($L_{\mathrm{h}} = L$), we obtain the output field in the following form:

$$E_1^{\mathrm{out}} = 2j\frac{1}{2}e^{-\alpha_{\mathrm{prop}}L}\left[E_1^{\mathrm{in}}\sin\left(\frac{\Delta\phi}{2}\right) + E_2^{\mathrm{in}}\cos\left(\frac{\Delta\phi}{2}\right)\right]e^{j\frac{1}{2}\frac{2\pi}{\lambda}(n_{\mathrm{eff},1}+n_{\mathrm{eff},2})L} \tag{3.12}$$

$$E_2^{\mathrm{out}} = 2j\frac{1}{2}e^{-\alpha_{\mathrm{prop}}L}\left[E_1^{\mathrm{in}}\cos\left(\frac{\Delta\phi}{2}\right) - E_2^{\mathrm{in}}\sin\left(\frac{\Delta\phi}{2}\right)\right]e^{j\frac{1}{2}\frac{2\pi}{\lambda}(n_{\mathrm{eff},1}+n_{\mathrm{eff},2})L} \tag{3.13}$$

where $\Delta\phi = 2\pi/\lambda(n_{\mathrm{eff},1} - n_{\mathrm{eff},2})L$ represents the aforementioned phase difference between the two MZI arms.

However, in this example, due to the unbalanced MMI splitting ratios, $\xi_1 \neq \xi_2$. Therefore, it will not be possible to obtain the closed-form analytical solution, but Eq. (3.6) and (3.7) can be solved numerically, yielding the same sinusoidal behavior.

As an example, Fig. 3.5 shows the results retrieved when the model (neglecting any loss and thermal crosstalk effects) is solved injecting 1 mW in input port 1 and applying an increasing voltage on the microheater (from 0 V to 2 V). The output power displays the behavior expected from the theory of MZIs: when no voltage is applied to the heater, the device is in cross state, meaning that the majority of the output power exits from port 2 if the light is injected in port 1; then, progressively increasing the applied voltage to 2 V, the MZI gradually switches to bar state, for which the majority of the power is routed to output port 1 if the light is injected in port 1. From Fig. 3.5, it is possible to appreciate that the P_{out}-V_{in} curve behaves as $\cos^2(V_{\mathrm{in}}^2)$. This result is compatible with the theoretical dependence of temperature difference on the applied voltage due to the Joule heating effect [90].

The method employed to estimate the physical parameters that appear in the model (e.g., losses, coupling ratios, etc.) will be described in the next section.

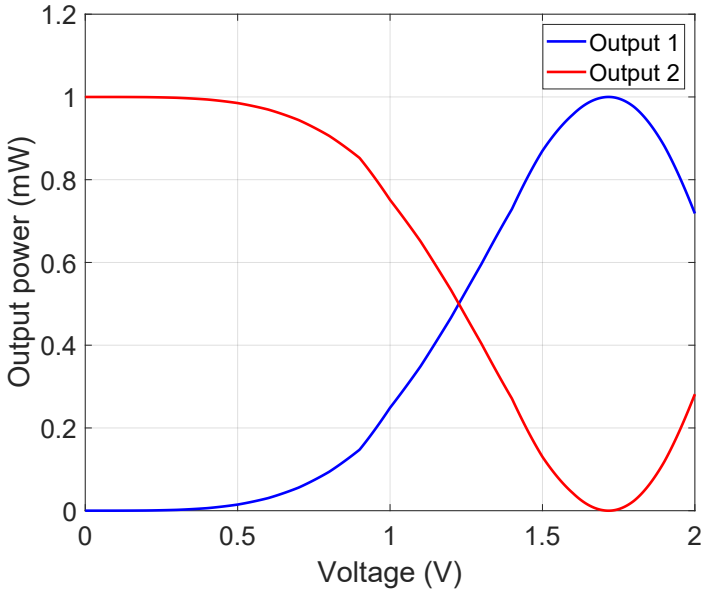

Figure 3.5 Example of output power of an MZI simulated with the presented model, when changing the voltage applied to the Ti TOPS from 0 V to 2 V and injecting 1 mW in input port 1. It is clear that the device operates in cross state when no voltage is applied (the majority of the power exits from the output port 2 when injecting light in input port 1); when the voltage is gradually increased, the MZI progressively switches to the bar state (the majority of the power exits from the output port 1 when injecting light in input port 1).

3.3 Modeling of secondary photonic components

In order to enable a more accurate description of the device, multiple wavelength-dependent loss terms are considered in the presented model. Although measurements of these terms are not readily available, the constitutive components of the circuit (multi-mode interferometers, straight and bent waveguides, crossings) are reproduced in RSoft™ Computer-Aided Design (CAD) [91] and simulated in order to obtain realistic values for the wavelength-dependent losses. To ensure accurate results for all the structures considered (in particular, the waveguide crossings and MMIs), the Finite-Difference Time-Domain (FDTD) method is employed.

With these FDTD simulations, the following quantities have been evaluated. For the MMI (Fig. 3.6(a)), the insertion loss α_{MMI} and the splitting ratios γ_1 and γ_2 have been computed with the following formulae:

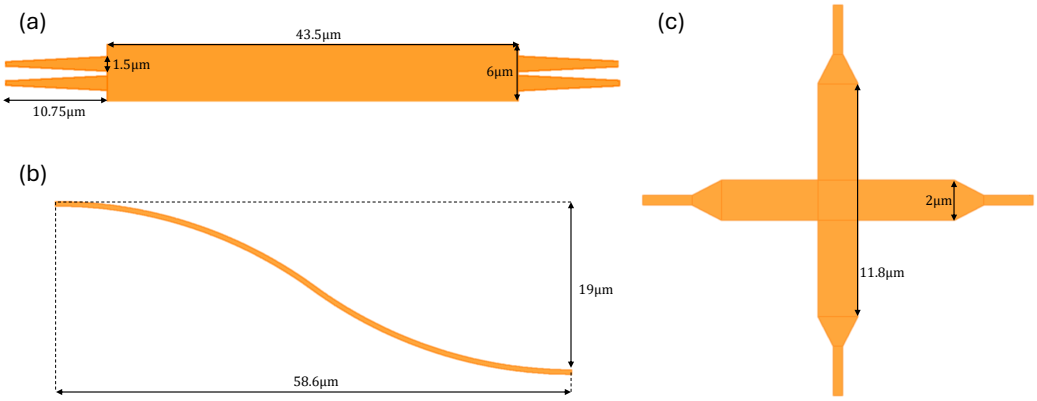

Figure 3.6 Silicon circuital elements simulated in RSoft™ CAD. The basic waveguide cross-section used in the simulations is $0.5\,\mu m \times 0.25\,\mu m$, the same as that of the fabricated waveguide, reported in Fig. 3.4. (a) Simulated MMI structure, used to evaluate its insertion loss and splitting ratios. (b) Simulated S-bent waveguide, used to evaluate the bending radiation losses. (c) Simulated waveguide crossing structure, used to evaluate its insertion loss.

$$\alpha_{\text{MMI}} = \frac{P_{11}^{\text{out}} + P_{21}^{\text{out}}}{P_1^{\text{in}}} = \frac{P_{12}^{\text{out}} + P_{22}^{\text{out}}}{P_2^{\text{in}}} \tag{3.14}$$

$$\gamma_1 = \frac{P_{11}^{\text{out}}}{P_{11}^{\text{out}} + P_{21}^{\text{out}}} \tag{3.15}$$

$$\gamma_2 = \frac{P_{22}^{\text{out}}}{P_{22}^{\text{out}} + P_{12}^{\text{out}}} \tag{3.16}$$

Then, for the bent waveguide of Fig. 3.6(b), assuming negligible propagation losses, the radiating losses are computed as:

$$\alpha_{\text{b}} = \frac{P_{\text{out}}}{P_{\text{in}}} \tag{3.17}$$

Finally, the insertion loss term for the waveguide crossing of Fig. 3.6(c) can be estimated for each combination of input and output ports, assuming negligible optical crosstalk between the possible optical paths [92]:

$$\alpha_{\text{cross}} = \frac{P_j^{\text{out}}}{P_j^{\text{out}}} \tag{3.18}$$

Table 3.2 Values of the MZI and crossing parameters obtained with RSoft™ CAD simulations. All reported parameters are dimensionless and in linear scale.

Parameter	$\lambda = 1545$ **nm**	$\lambda = 1550$ **nm**	$\lambda = 1555$ **nm**
Bending radiation term α_b	0.9901	0.9892	0.9868
MMI insertion loss α_{MMI}	0.8789	0.8947	0.9099
MMI splitting ratio 1 γ_1	0.5030	0.5046	0.5058
MMI splitting ratio 2 γ_2	0.5030	0.5047	0.5058
WG crossing insertion loss α_{cross}	0.8708	0.8844	0.8974

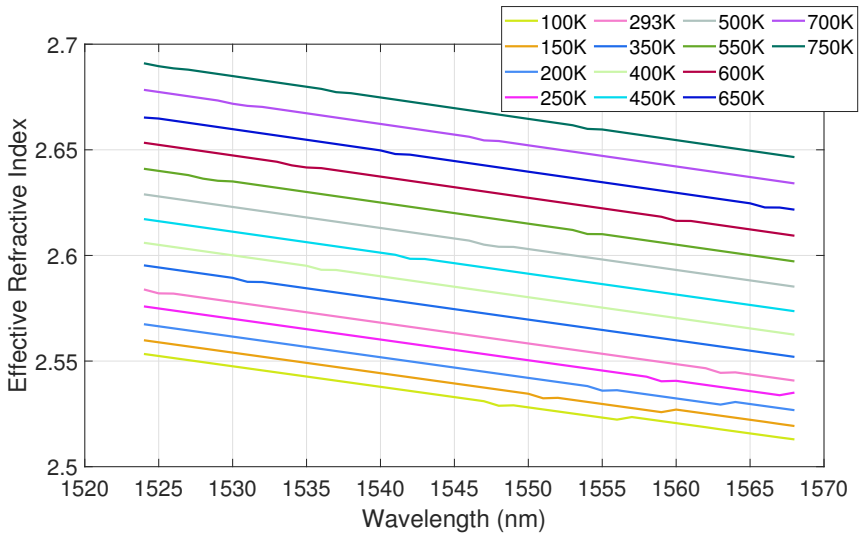

Figure 3.7 Wavelength dependent behavior of the effective refractive index n_{eff}, for different temperatures.

These simulations have been performed on the wavelength range from 1524 nm to 1568 nm, which includes the entire C-band. The simulations yield power spectra that are used to evaluate the quantities listed above. These are reported in Table 3.2 for three different sample wavelengths, namely 1545 nm, 1550 nm (the nominal design wavelength of the chip), and 1555 nm. The simulation results show that the MZI parameters depend weakly on the wavelength over the considered range, implying a wide-band of operation for the device. Note that, in these simulations, we neglected all nonlinear effects (e.g., Two Photon Absorption (TPA)), assuming that we are always working with sufficiently low input power levels [93].

A single straight Si waveguide with the same cross-section as that of the waveguides of the fabricated device (Fig. 3.4) was simulated using RSoft™ in order to compute the effective refractive index n_{eff}, estimating its dependence on both wavelength and temperature. This temperature dependence will be important when

modeling the thermal effects in Section 3.4. Since RSoft™ does not offer explicit thermal simulations, the temperature dependence of n_{eff} is estimated by setting different refractive index values n at different temperatures [94] inside the simulator and calculating n_{eff} with the mode computation tool of the software. The simulated wavelength-dependent n_{eff} curves obtained at different temperatures are reported in Fig. 3.7.

3.4 Thermal crosstalk model

In this section, we will describe in detail how the thermal effects have been modeled to describe both device-level and circuit-level interactions.

The main thermal effect to be included, of course, consists in the thermal control of the MZIs by means of Ti TOPSs. A voltage is applied to the Ti strip that will heat up as a result of the Joule heating effect and, consequently, will increase the temperature in the waveguide underneath. This induces a change of the n_{eff} of the waveguide and a subsequent phase difference between the two arms of the interferometer. Following a well-known approach [90], we can express the dependence of the effective refractive index with respect to temperature T by introducing a first-order Taylor expansion:

$$n_{\text{eff}}(T) = n_{\text{eff}}(T_0) + \left.\frac{\mathrm{d}n_{\text{eff}}}{\mathrm{d}T}\right|_{T_0} (T - T_0) \tag{3.19}$$

The derivative of n_{eff} with respect to the temperature T is computed starting from the effective refractive index curves of Fig. 3.7 and it amounts to $1.9832 \times 10^{-4}\ \mathrm{K}^{-1}$ at $\lambda = 1550\,\mathrm{nm}$, which is in line with the values reported in the literature for waveguides with similar cross-sections [95].

The n_{eff} shift is related to the temperature difference $T - T_0$, which, in turn, is caused by the Joule heating induced by the input voltage V_{in} applied to the Ti microheater. In order to be able to relate this temperature shift to the applied voltage that caused it, we chose to use targeted COMSOL Multiphysics® simulations of a simplified system. This approach allows us to integrate this effect into a larger simulation framework and achieve an accurate representation of thermal effects, without relying on complex analytical or numerical models.

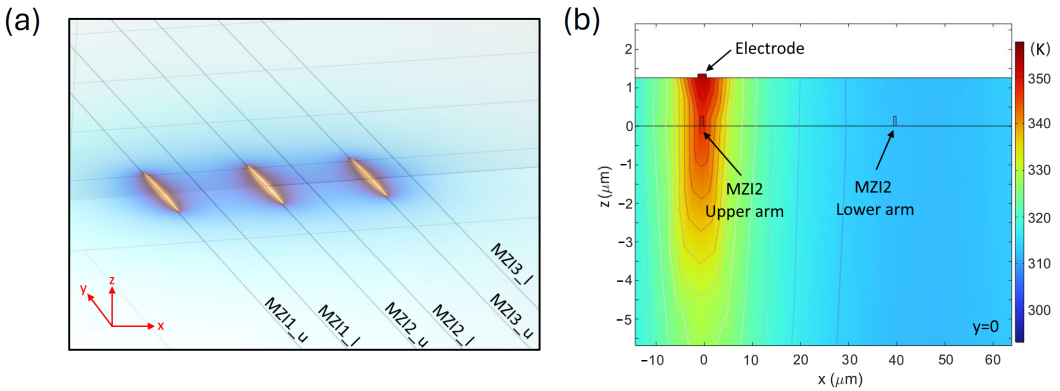

Figure 3.8 (a) 3D view of the simulated domain and corresponding heat map obtained when a voltage of 2 V is applied to the three electrodes. The straight lines indicate the upper and lower arms of MZIs 1, 2, and 3. (b) COMSOL heat map in the transverse plane intersecting the electrodes ($y = 0$), when 2 V are applied to the microheater of MZI 2. The geometry is not shown to scale for better graphical clarity.

In COMSOL, we simulated a 3D structure consisting of six parallel Si waveguides placed on a $1\,\text{mm} \times 1\,\text{mm} \times 200\,\text{mm}$ SiO_2 substrate, covered by a SiO_2 cladding layer. On top of this cladding, three Ti strips representing the microheaters are positioned. Because of the circuit geometry, we assume that thermal effects are significant only for devices aligned along the same y-position (black dashed rectangle in Fig. 3.2), while they are negligible in other regions, since the distance between the interferometer arms is much smaller than the separation between different MZIs along y. Thermal crosstalk can be estimated by calculating the overall spatial temperature variation produced by three vertically stacked MZIs.

In the COMSOL simulations, we included the Joule heating effect in the Ti electrodes, convection between solid materials for the transfer of heat in the rest of the geometry, and a linear resistivity model for the metal strips. The Boundary Conditions (BCs) of the simulations are as follows: the top surface exchanges heat with the surrounding air through convection (Robin BCs [96]); the lateral surfaces are assumed to be adiabatic, without any external heat exchange (homogeneous Neumann BCs [96, 97]); the SiO_2 substrate is kept at ambient temperature with an ideal Peltier cell (Dirichlet BCs [96, 97]). Figure 3.8(a) shows a screenshot of the simulated system in COMSOL Multiphysics®. Fig. 3.8(b) also illustrates the layer structure in the transverse plane.

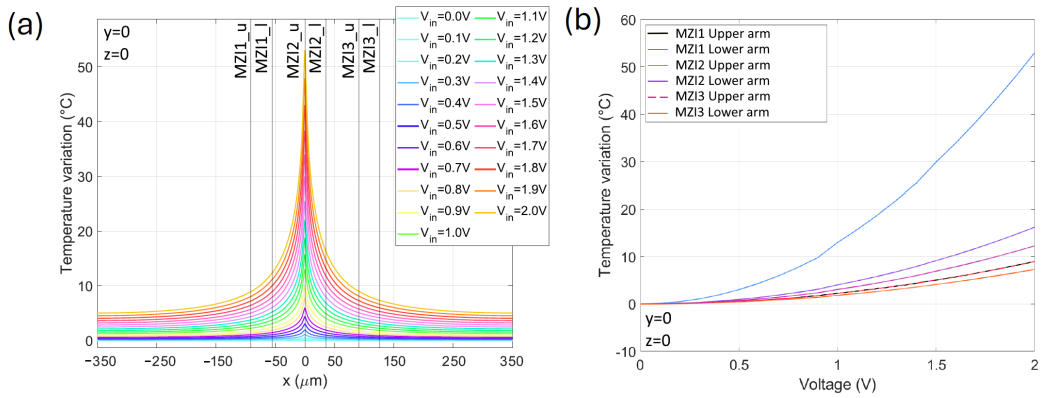

Figure 3.9 (a) Spatial distribution of the temperature variation when different voltages $V_{\text{in},2}$ are applied to the microheater of MZI 2. The vertical black lines indicate the positions of the six waveguides. (b) Temperature in the waveguides as a function of the voltage applied to the microheater of MZI 2; the voltages of MZIs 1 and MZI 3 are approximately the same.

With these COMSOL simulations, we can estimate the spatial temperature variation caused and experienced by the single MZI when applying a V_{in}. Fig. 3.9(a) represents the temperature variation from room temperature (20 °C) when a voltage $V_{\text{in},2}$ is applied to the central heater while the other ones are grounded. For instance, let us consider $V_{\text{in},2} = 2\,\text{V}$: in this case, we observe that the temperature in the waveguide directly below that heater increases by $\sim 52\,°\text{C}$, but also that the temperature variation is $\sim 16\,°\text{C}$ in the lower arm of the same MZI (Fig. 3.9(a) and 3.9(b)). This implies that the Ti heater is characterized by a reduced thermal tuning efficiency, as the optical path variation is proportional to the difference in temperature between the two arms ($\Delta T_{\text{MZI},2} = 36\,°\text{C}$). Even more importantly, there is also a significant temperature variation between the waveguides of MZI 1 ($\Delta T_{\text{MZI},1} = 5\,°\text{C}$) and MZI 3 ($\Delta T_{\text{MZI},3} = 2\,°\text{C}$), despite both having grounded heaters: this is thermal crosstalk. Despite the possibility of mitigating it with a larger separation between the waveguides of the same MZI or between different MZIs, or with insulation trenches, these solutions would imply either lower integration density or increased fabrication complexity. For this reason, the presented model is beneficial to predict spurious interactions and account for them during the training of a PNN.

With these COMSOL simulations, we can now compute the temperature shift experienced by each waveguide of the system when the MZIs are turned on separately or when multiple MZIs are switched at the same time, which is fundamental to describe a realistic use of the device affected by thermal crosstalk. Instead of simulating each possible combination of input voltages that can be applied to the mi-

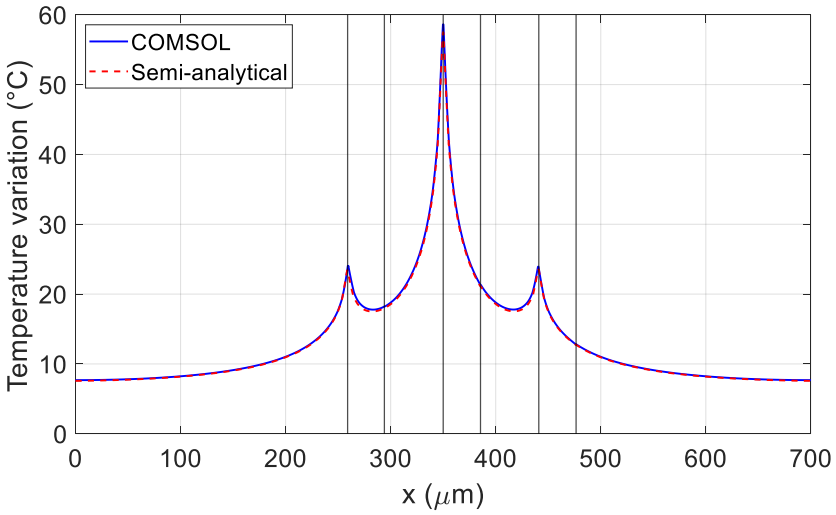

Figure 3.10 Spatial distribution of temperature variation for $V_{\text{in},1} = 1\,\text{V}$, $V_{\text{in},2} = 2\,\text{V}$, $V_{\text{in},3} = 1\,\text{V}$ applied to the three microheaters, simulated in COMSOL Multiphysics® (solid blue line) and reconstructed with our method (dashed red line).

croheaters of the three stacked MZIs, which would be time consuming and resource intensive, we can employ the results of the thermal simulation of the single heater (Fig. 3.9(a)): for each MZI $k = 1, 2, 3$ of the stack, we select the curve at the correct $V_{\text{in},k}$ and shift it in the x direction (for the lateral MZIs). The three contributions are summed to approximate the complete spatial temperature distribution. The same approach is used for the two other groups of MZIs present in the device, in order to compute the thermal response of the complete circuit. This strategy was applied to the reference circuit of Fig. 3.2, but it can be extended in order to simulate circuits with a larger number of stacked MZIs or even circuits where the y direction thermal interactions are non-negligible.

Fig. 3.10 depicts the reconstructed spatial distribution of the temperature variation with $V_{\text{in},1} = 1\,\text{V}$, $V_{\text{in},2} = 2\,\text{V}$, and $V_{\text{in},3} = 1\,\text{V}$ applied to three heaters, compared to the actual COMSOL simulation of the system represented in Fig. 3.8(a): it is evident that the results obtained with our procedure accurately reproduce the COMSOL thermal simulations. Moreover, to showcase the flexibility of this approach, we simulated two additional MZIs in the COMSOL system of Fig. 3.8(a) and compared its results to the thermal curve for the 5-MZI system reconstructed with our method. This was done applying $V_{\text{in},1} = 1\,\text{V}$, $V_{\text{in},2} = 1.5\,\text{V}$, $V_{\text{in},3} = 2\,\text{V}$, $V_{\text{in},4} = 1.5\,\text{V}$, and $V_{\text{in},5} = 1\,\text{V}$ to the heaters and the results are reported in Fig. 3.11. The comparison

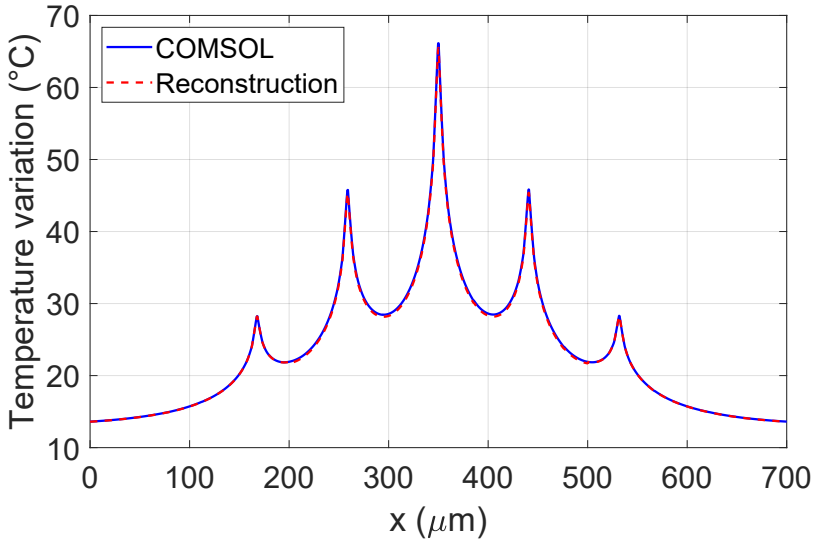

Figure 3.11 Example of thermal distribution of a 5-MZIs system with $V_{\text{in},1} = 1\,\text{V}$, $V_{\text{in},2} = 1.5\,\text{V}$, $V_{\text{in},3} = 2\,\text{V}$, $V_{\text{in},4} = 1.5\,\text{V}$, $V_{\text{in},5} = 1\,\text{V}$ applied simultaneously to the heaters, simulated in COMSOL Multiphysics (solid blue line) and reconstructed using our method (dashed red line).

shows that the methodology does not lose accuracy in predicting the thermal spatial distribution with a larger number of MZIs; therefore, the presented model can be employed to analyze more complex architectures.

With this instrument, we can sample temperature change ΔT in the positions corresponding to each waveguide and use these in Eq.(3.19) to compute the effective refractive index n_{eff} variation induced by the voltages applied to each heater. Note that this approximation of the sum of three contributions holds because the heat sources are far enough from the box borders in the x direction, otherwise the adiabatic BCs would not be true and would affect the result.

This way, we are able to create a model that can rapidly compute the response of a meshed MZI-based topology, including multiple effects, thus maintaining the physical accuracy, but without any time-consuming *ad-hoc* multi-physics simulations. Moreover, by computing the overall thermal distribution of the stack of MZIs, we are able to capture the thermal interplay between the various active devices in a single step, without requiring complicated analytical models.

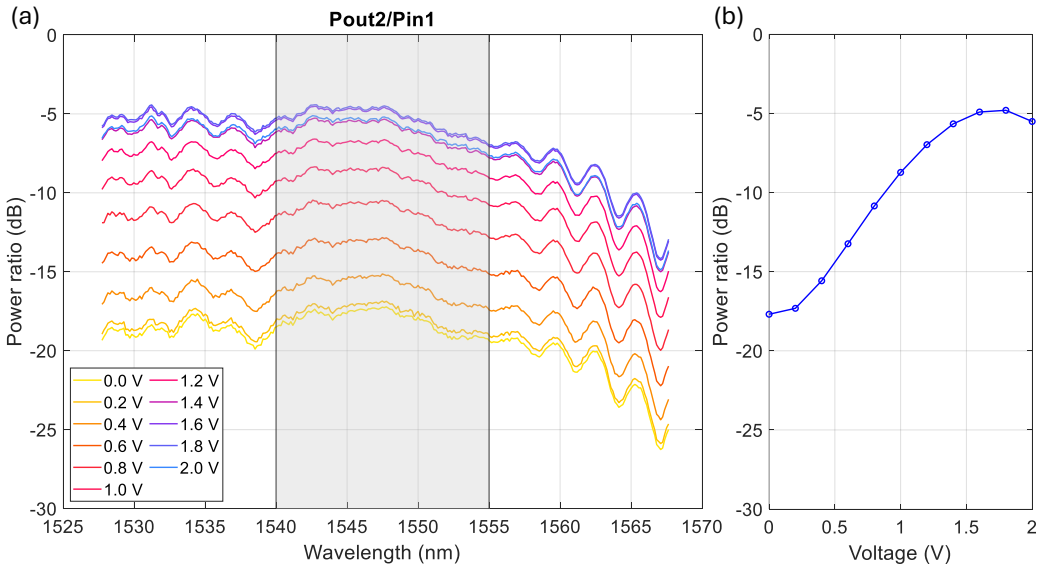

Figure 3.12 (a) Measured spectra obtained by applying increasing input voltages V_{in} to MZI 1 for the $P_{\text{out},2}/P_{\text{in},1}$ ratio. The gray window represents the range of wavelengths that were averaged to create the dataset. (b) Resulting $P_{\text{out},2}/P_{\text{in},1}$ power ratio curve with respect to V_{in} associated to MZI 1, obtained by averaging the spectra in the gray box.

With the complete model developed in Sections 3.2, 3.3, and 3.4, we are now able to simulate the behavior of the circuit of Fig. 3.2, in order to validate the model itself.

3.5 Experimental validation results

In this section, we will validate the model by comparing the simulated results with measurements of the actual device. For this purpose, we employ a set of measured output-input power ratios that are measured as follows: for each pair of input-output ports, a broadband signal is injected into one of the input ports and each MZI is gradually tuned, by spanning its input voltage from 0 V to 2 V in steps of 0.1 V, while all the other MZIs are grounded. Note that all the reference measurements used in this section have been performed with the chip mounted on a thermally controlled stage [85].

Fig. 3.12(a) reports an example of raw measured data [77, 85]: Amplified Spontaneous Emission (ASE) is injected into input port 1 and measured with an Optical

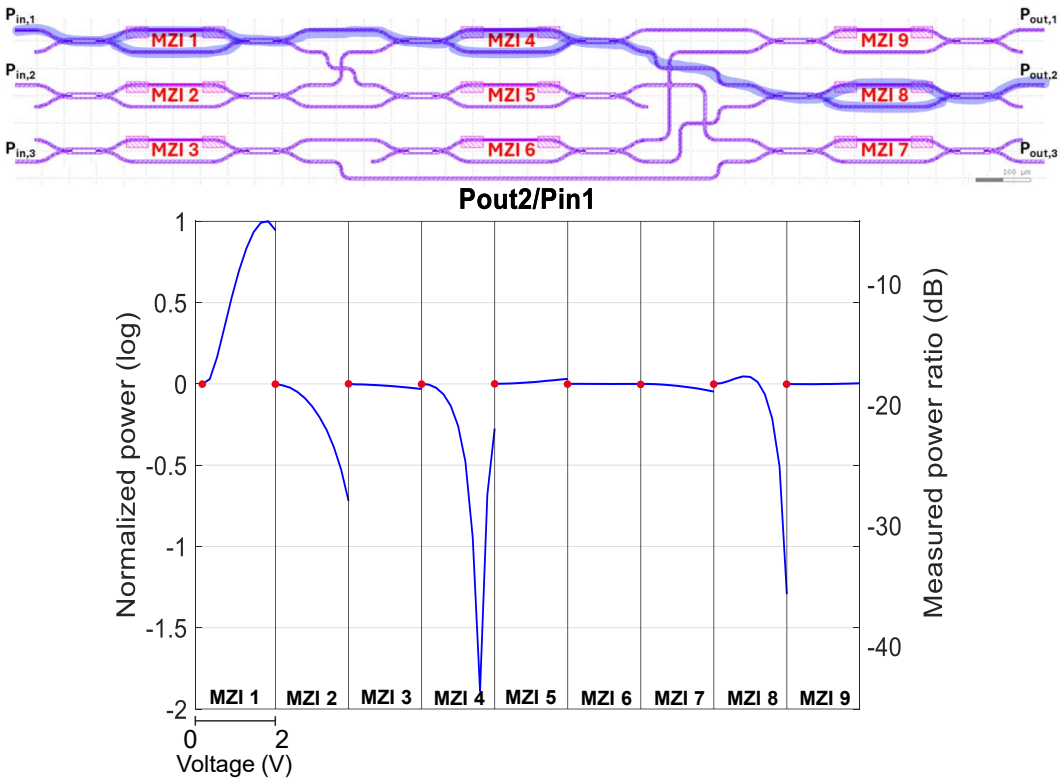

Figure 3.13 Example of experimental $P_{\text{out}}/P_{\text{in}}$ curve (input port 1, output port 2) and the related optical path through the circuit. The left y axis represents the raw measured data (in dB); the right y axis represents the power ratio scaled with Eq. (3.20).

Spectrum Analyzer (OSA) at output port 2 [85], for various values of $V_{\text{in},1}$ applied to MZI 1. The flat measured responses confirm the wideband properties of the device, which will also be investigated in Section 3.6. In order to have a set of reference curves that is more robust to noise and to simplify the subsequent analysis, the spectra were averaged over the 1540 nm-1555 nm range (gray box in Fig. 3.12(a)), leading to the power ratio curve shown in Fig. 3.12(b). Moreover, this process is repeated by applying an input voltage to each MZI in the sequence, thus obtaining eight additional averaged curves similar to the one of Fig. 3.12(b). These can be concatenated to create a single trace, as the one reported in Fig. 3.13.

The concatenated curve P_{dB} of Fig. 3.13 has been normalized using Eq.(3.20), which aligns the maximum value P_{max} (equal to −4.8 dB) to 1, while it converts the value P_{ref} measured with all null driving voltages (equal to −17.7 dB) to 0:

$$P_{\text{scaled}} = \frac{P_{\text{dB}} - P_{\text{ref}}}{P_{\text{max}} - P_{\text{ref}}} \tag{3.20}$$

This scaling operation allows for easier comparison with our simulation results, as experimental measurements may include additional optical losses (e.g., measurement setup losses) that are not accounted for in our model and are not readily quantifiable.

This kind of measured data is compelling because it provides concrete evidence of the impact of thermal crosstalk on the circuit's response. As shown in Fig. 3.13, we can observe three primary contributions to the $P_{\text{out}}/P_{\text{in}}$ curve, due to the three MZIs located along the optical path from input 1 to output 2, specifically MZIs 1, 4, and 8. Initially, all curves start from the same $P_{\text{out}}/P_{\text{in}}$ value, which corresponds to the case when all MZIs are grounded, as indicated by the red marker in Fig. 3.13. In this scenario, MZI 1 is in cross state, so $P_{\text{in},1}$ is routed primarily to MZI 5, with a minor residual directed to MZI 4 due to its non-ideal behavior, as discussed earlier. Since MZIs 5 and 7 are also in cross state, most of $P_{\text{in},1}$ ultimately goes to output 3. However, when a voltage is applied to the heater of MZI 1, it switches to bar state, directing $P_{\text{in},1}$ to output 2. Similarly, power is diverted from output 2 as MZI 4 transitions from cross to bar state and MZI 8 shifts from bar to cross state (as indicated in Section 3.1, MZIs 6 and 8 are naturally in bar state when grounded [85]). Nevertheless, experimental observations reveal that, despite not being on the direct light path between input 1 and output 2, MZI 2 influences the system: the heater of MZI 2 affects the waveguides of MZI 1, causing it to further assume cross state, which diverts power from output 2. This demonstrates an example of thermal crosstalk. Given its non-negligible effect when employing a single MZI with a single input port, it is clear that the impact will be even more pronounced when the circuit operates at full capacity.

In order to improve the match with the experiments by accounting for process variations, the phase correction terms $\delta\varphi$ introduced in Eq. (3.3) are now adjusted for each MZI. This can be done with an optimization procedure, for instance, using the Particle Swarm Optimization (PSO) method [98, 99]. All the details regarding the use of PSO in the context of parameter extraction are discussed in Appendix A.

The fitting parameters obtained with this procedure are reported in Table 3.3. Note that, for MZI 6 and MZI 8, we obtained values close to $\pm 180^\circ$, consistent with experimental evidence that these two devices are in the bar state when $V_{\text{in}} = 0\,\text{V}$, showing opposite behavior with respect to the other MZIs [85]. These fitting parameters are then used in our model to reproduce the $P_{\text{out}}/P_{\text{in}}$ measurements.

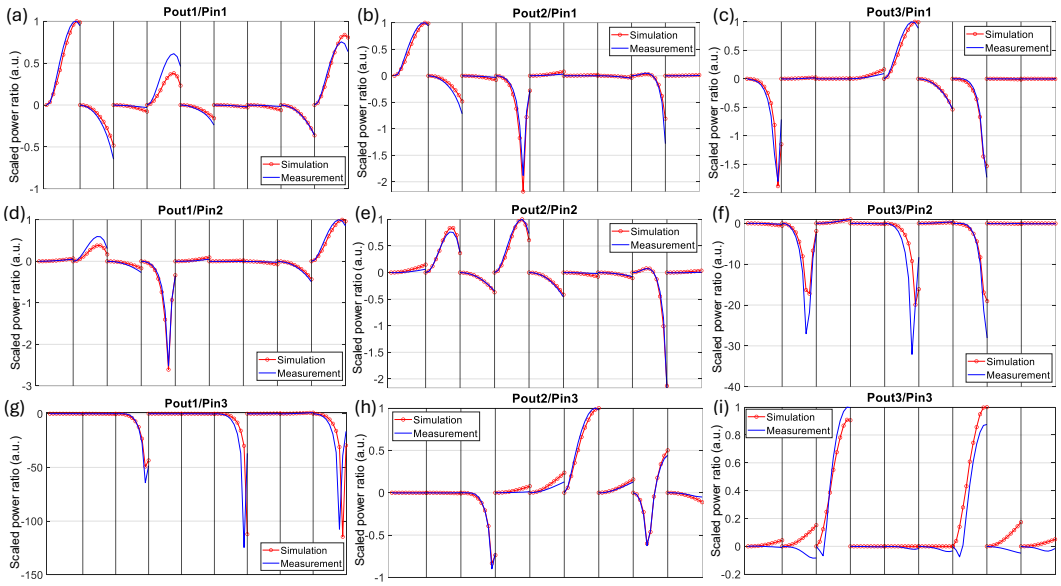

Figure 3.14 (a)-(i) Comparison of the results obtained by simulating the model with the fitted $\delta\varphi$ terms (red circles) and the measured curves (blue lines), for each combination of input/output ports (specified in the title of each panel). Each curve related to a single MZI is delimited by two consecutive black lines, indicating a 0 V-2 V span of applied input voltages. For all panels, the curves are concatenated and normalized as in Fig. 3.13.

In Fig. 3.14, we show the power ratios for all possible combinations of input and output ports. The solid blue lines represent the measured data [78], while the circled red lines indicate the simulated results employing the phase corrections listed in Table 3.3. The curves are normalized using Eq. (3.20), with the corresponding experimental values of $P_{\max}$ and P_{ref} for each input–output port combination. From the comparison, it is clear that our model, with the optimized phase correction terms, closely matches the experimental results: the overall behavior is well reproduced, showing that the model correctly captures the thermal crosstalk between neighboring MZIs. This is useful for compensating or accounting for these effects in specific applications (e.g., offline training of a PNN based on thermally-tuned MZIs).

Small discrepancies are still present between the simulated and experimental data. For instance, in Fig. 3.14(a), for MZI 4, the simulation produces a lower peak power. This and similar deviations can be attributed to additional effects in the real device or to measurement uncertainties, particularly near the minima of the transfer function. In Fig. 3.14(i), the simulated trends of MZI 2 and MZI 8 do not match the experimental results. However, it should be noted that this is the only case in which

Table 3.3 Phase correction terms $\delta\varphi$ extracted with PSO to fit the experimental curves.

MZI1	MZI2	MZI3	MZI4	MZI5	MZI6	MZI7	MZI8	MZI9
0.6299	1.5139	0.5423	1.3986	0.5242	-2.9750	0.7501	2.6017	0.4769

the experimental peak-to-floor difference is about $\sim 30\,\mathrm{dB}$. Moreover, the lowest value of the $P_{\mathrm{out},3}/P_{\mathrm{in},3}$ curve is $-73\,\mathrm{dB}$, which may be limited by the noise floor of the OSA used for the measurements.

By validating the model and adjusting the $\delta\varphi$ terms to reproduce the behavior of each separate MZI of the circuit, we have effectively created a "digital twin" of the PIC of Fig. 3.2. This methodology, based on the description of the single building blocks and completed by the inclusion of circuit-level thermal interactions (e.g., parasitic thermal crosstalk), can be easily generalized to describe more complex architectures, even ones that are not based on thermally-tuned MZIs, provided the proper description of the basic devices. This possibility is supported by the evidence that our methodology to describe thermal crosstalk can be extended to a larger number of stacked MZIs.

In the next section, the fitted and validated model will be employed as a "digital twin" of the device, in order to explore the possibility of using the device itself as a photonic programmable Boolean gate, able to implement user-defined logic functions on the three output ports separately.

3.6 Case study: Boolean programmability

In this section, we will explore the possibility of employing the model described in Sections 3.2 and 3.4 and validated in Section 3.5 as a "digital twin" of the real device of Fig. 3.2, namely the use of simulations to establish the driving voltages needed to obtain a given behavior from the PIC under test. The target application is the use of this simple 3×3 MZI mesh as a photonic programmable Boolean gate, able to implement user-defined logic functions on the three output ports separately. Basically, the fitted model will be employed to retrieve the voltages that enable the implementation of the wanted logic functions on the real device.

In the context of photonic neuromorphic computing, this procedure is called "offline training" and consists in performing the backpropagation on a traditional

computer and applying the retrieved weights (i.e. the voltages to be applied to the MZI microheaters) on the chip *a posteriori* [100]. However, the offline training procedure presents a series of disadvantages: due to the presence of unforeseen fabrication variations that modify the operating conditions of supposedly identical devices, the offline training accuracy can be drastically reduced [101]. While this accuracy reduction could be mitigated with error correction procedures, these are usually extremely challenging and time-consuming [100].

This limitation can be overcome with the use of "online training" procedures, where the backpropagation is executed directly on the photonic chip, thus automatically accounting for those defects and fabrication variations [100–102] that are the main source of training accuracy loss. These procedures can even be "physics agnostic" for better flexibility [102].

Our physical circuit-level model can be employed in conjunction with an offline training procedure, but it offers multiple advantages with respect to both online and traditional offline procedures. First, our model allows us to rapidly perform a large number of simulations, enabling in-depth exploration of the solution space. Indeed, employing a modern workstation, we are able to simulate approximately up to 1×10^4 different configurations of input voltages per second per core, thus enabling the rapid generation of large datasets for ML applications or the possibility of performing effective optimization procedures. This is an advantage with respect to *in situ* procedures, due to the inherent time-consuming nature of many laboratory measurements. For instance, approximately 40 hours were needed to create a dataset of 5000 measurements on the 3×3 reference circuit [85].

In addition to allowing us to generate larger datasets in a fraction of the time required experimentally, the model can also be used during the PIC design phase, enabling optimization of specific features (e.g., fabrication robustness, thermal crosstalk effects, etc.), without the need for expensive and time-consuming iterative prototyping.

Secondly, as mentioned previously, one of the main drawbacks of traditional offline training methods is the susceptibility to fabrication variations that make the real device deviate from the behavior expected from theory. However, this is taken care of by our methodology by using the phase correction terms $\delta\varphi$. With these, the behavior of each building block is better predicted by the model, thus automatically mitigating the accuracy loss effect.

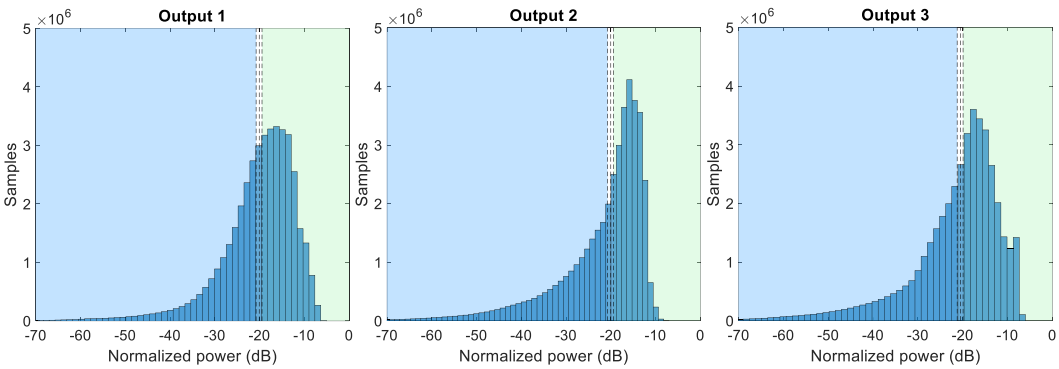

Figure 3.15 Probability density functions of the output power datasets for the three output ports of the reference device. The blue region represents the power values converted to logic 0s, while the green region represents the power values converted to logic 1s. The vertical black dashed lines represent the 0 and 1 thresholds and the median power of the dataset. Note that the power on the x axis has been represented in dB for optimal graphical clarity.

Within the framework of this physics-informed offline methodology, we employ the fitted model in order to retrieve combinations of driving voltages for the 9 MZIs that allow us to implement arbitrary Boolean functions at the three output ports separately.

Before tackling the extraction of these driving voltages, we first have to discuss the problem of converting the inherently analogical optical signals into digital signals (0s and 1s, for false and true, respectively). For this purpose, despite the existence of advanced Analog-to-Digital Conversion (ADC) techniques for optical signals [103], we employ an intensity-based approach by setting thresholds to distinguish logic 0s (low power output) and logic 1s (high power values).

In order to establish the 0 and 1 thresholds, we first generate 6 million randomized combinations of the 9 driving voltages and we then simulate them with our model, for each possible permutation of high and low values of input signals on the three input ports (2^3 in total). With these simulations, we are able to generate a dataset of 24 million values of power for each one of the three output ports. Thanks to the computational efficiency of the presented model, this dataset has been generated in less than 6 minutes on an Intel® i9 12$^{\text{th}}$ generation workstation. Then, for each dimension $k = \{1,2,3\}$ of the dataset, where k is the output port number, we can define the medians $P_{\text{m},k}$ of the k-th port and use it for the ADC with the following rule:

$$S_{\mathrm{d},k} = \begin{cases} 0, & P_{\mathrm{a},k} \leq \kappa_0 \cdot P_{\mathrm{m},k} \\ 1, & P_{\mathrm{a},k} \geq \kappa_1 \cdot P_{\mathrm{m},k} \end{cases} \tag{3.21}$$

where $S_{\mathrm{d},k}$ is the logical value resulting from the ADC of the power $P_{\mathrm{a},k}$ measured at the k-th port. $\kappa_0 = 0.85$ and $\kappa_1 = 1.15$ are the weights employed to distinguish the lower and upper thresholds starting from the medians of the datasets. This distinction between low and high thresholds has been introduced to mitigate the effect of measurement uncertainties that could affect the ADC in an experimental setting. A representation of the Probability Density Function (pdf) of the power distribution of each output port is reported in Fig. 3.15.

Once the logic thresholds are defined, we can test the performance of the reference circuit as a programmable logic gate using the proposed model. After defining the desired logic functions (which may include don't-care terms X), we need to determine the appropriate set of nine V_{in} voltages that enables the reproduction of the correct truth tables. This task can be accomplished either with a trained ML agent [104, 105] or through an optimization routine [99, 106]. We choose the latter approach and, to accelerate the computation, we first search our dataset of 6×10^6 entries for the combination of voltages that best approximates the desired truth table.

Table 3.4 lists the logic functions that have been tested. In several cases, the solution is not unique, meaning that different sets of voltages can implement the same target functions. Furthermore, the same group of functions can appear under multiple output permutations. For example, in case 2, the logic `and` can be obtained on port 1 and the logic `or` on port 2, or vice versa. The second-to-last column of Table 3.4 indicates if a combination of V_{in} values that can generate the desired functions was found, either exactly as listed or with permuted output ports. It should be noted that, for physical reasons, not all analyzed cases can be implemented successfully. For instance, it is not possible to invert all three inputs simultaneously on the three outputs (case 9): when all input signals are 0 (i.e. no power injected in the input ports), no power can be measured at any output.

Nevertheless, the simple 3×3 device can implement multiple basic logic functions (`and`, `or`, `xor`, `nand`, `nor`, sum of products, and product of sums), invert the signals at ports 1 and 2 when a logic 1 is applied to port 3, program half and full adders (between ports 1 and 2, with a carry-in on port 3), and compute the two's complement

Table 3.4 List of logic functions tested on the reference device. The second-to-last column indicates whether each function can be successfully implemented by the device, while the last column reports the maximum input voltage tolerance that still ensures correct operation. Cases marked with an asterisk require a constant input signal of $P_3 = 1$ to correctly perform the desired function. X denotes don't-care; SoP refers to sum of products and PoS to product of sums.

#	Description	$f_{out,1}$	$f_{out,2}$	$f_{out,3}$	OK?	tol.
1	No operation	P_1	P_2	P_3	✓	5%
2	`or/and`	$P_1+P_2+P_3$	$P_1P_2P_3$	X	✓	1%
3	`and/xor/or`	P_1P_2	$P_1 \oplus P_2$	P_1+P_2	✓	5%
4	`and/and/xor`	P_1P_3	P_2P_3	$P_1 \oplus P_3$	✓	1%
5	`or`	P_1+P_2	P_1+P_3	P_2+P_3	✓	5%
6*	`nand/nor`	$\overline{P_1P_2}$	$\overline{P_1+P_2}$	X	✓	2%
7	SoP/PoS	$(P_1P_2)+P_3$	$(P_1+P_2)P_3$	X	✓	2%
8*	2bit `not`	$\overline{P_1}$	$\overline{P_2}$	X	✓	5%
9	3bit `not`	$\overline{P_1}$	$\overline{P_2}$	$\overline{P_3}$	✗	–
10	Half adder	$P_1 \oplus P_2$	P_1P_2	X	✓	2%
11	Full adder	$P_1 \oplus P_2 \oplus P_3$	$P_2P_3+P_1(P_2 \oplus P_3)$	X	✓	2%
12	Half sub.	$P_1 \oplus P_2$	$P_1\overline{P_2}$	X	✓	5%
13	Full sub.	$P_1\overline{P_3}+P_2\overline{P_3}+P_1P_2$	$P_1 \oplus P_2 \oplus P_3$	X	✗	–
14	Half add/sub	$P_1 \oplus P_2$	$(P_1 \oplus P_3)P_2$	X	✗	–
15	2bit 2s compl.	$P_2 \oplus P_1 \oplus P_2$	$P_1 \oplus P_2$	X	✓	2%
16	3bit 2s compl.	P_1	$P_1 \oplus P_2$	$(P_1+P_2) \oplus P_3$	✓	2%
17	Select. `and/or`	$P_3(P_1P_2)+\overline{P_3}(P_1+P_2)$	X	X	✓	5%
18	Select. `nor/or`	$P_3(\overline{P_1+P_2})+\overline{P_3}(P_1+P_2)$	X	X	✓	5%

of 2-bit and 3-bit input numbers. It is also possible to configure the device so that it performs the logic `and` of the input signals at ports 1 and 2 when the signal at port 3 is true, or the logic `or` otherwise, using the optical signal at port 3 to select the operation. These results highlight the flexibility of the device.

From the experimental point of view, it might happen that, while applying the driving voltages to the device, these could be affected by random fluctuations of the source. For this reason, we want to assess the robustness of our results with respect to voltage uncertainties, in order to estimate the reliability of the device itself under realistic operation conditions. For this purpose, we perform a series of Monte Carlo simulations running 10^6 simulations with random perturbations applied to the nine nominal voltages previously determined for each successful case listed in Table 3.4. The perturbations are uniformly distributed within $\pm 5\%$ of V_{nom} and, for each perturbed run, we verify whether the same truth table is obtained. For the cases where $\pm 5\%$ deviations do not always produce the correct output, we also test $\pm 2\%$ and, if needed, $\pm 1\%$ of V_{nom}, still within the range supported by standard

Table 3.5 Results of the wideband analysis. The rightmost column contains the largest operational band for each one of the working arbitrary functions of Table 3.4.

Case	Description	Working wavelength range
1	No operation	1524 nm – 1568 nm
2	or/and	1524 nm – 1568 nm
3	and/xor/or	1524 nm – 1568 nm
4	and/and/xor	1550 nm
5	or	1524 nm – 1568 nm
6	nand/nor	1524 nm – 1568 nm
7	SoP/PoS	1524 nm – 1568 nm
8	2 bit not	1524 nm – 1568 nm
10	Half adder	1524 nm – 1568 nm
11	Full adder	1524 nm – 1568 nm
12	Half subtractor	1558 nm – 1561 nm
15	2bit 2's complement	1524 nm – 1568 nm
16	3bit 2's complement	1524 nm – 1568 nm
17	Selectable and/or	1524 nm – 1568 nm
18	Selectable nor/or	1524 nm – 1568 nm

electronic equipment. The rightmost column of Table 3.4 reports the maximum voltage tolerance that yields the correct truth table across all 10^6 runs, despite the applied perturbations. As shown, a tolerance range is found for all functional logic operations, suggesting that the device is not only capable of implementing arbitrary logic functions, but that it is also stable with respect to noise in the electrical control signals, up to 5% of the nominal driving voltage values in the most robust cases.

Up to now, the Boolean programmability analysis has been carried out considering signals at 1550 nm, i.e. the design wavelength of the reference circuit of Fig. 3.2. However, in light of the wideband properties demonstrated via the RSoft™ simulations of Section 3.3 and by the flat spectral responses such as the one reported in Fig. 3.12, it is possible to predict the ability of the device to perform these Boolean operations on a larger range of wavelengths, for instance, in a WDM scenario in which multiple logic operations are performed in parallel at different wavelengths.

Starting from results of Table 3.4, the same driving voltage combinations found to implement the Boolean functions are now tested on the wavelength range between 1524 nm and 1568 nm (covering the entire C-band) with 1 nm spacing, to investigate the capability of the device to yield the desired truth table on a wider wavelength

interval. In order to have a more general approach, the same thresholds for 0s and 1s that have been computed at 1550 nm (Fig. 3.15) will be used for this wideband analysis. A given wavelength is considered functioning if all three output ports yield the wanted truth table.

Table 3.5 reports the results of the wideband analysis for the working arbitrary functions of Table 3.4. As it is possible to appreciate from the rightmost column, most of the considered Boolean functions are actually exhibiting the wanted behavior on the entire range between 1524 nm and 1568 nm. The only two exceptions are cases 4 and 12: case 4 (`and/and/xor`) provides the correct truth table only in single, separate wavelengths, for instance, the nominal 1550 nm; while for case 12 (half subtractor) the largest continuous interval for which the results are correct is 3 nm (from 1558 nm to 1561 nm). In both non-ideal cases, the expected behavior is also observed at isolated wavelengths, although they lie outside the largest continuous working band.

It is important to specify that, both in cases 4 and 12, the operational bandwidth is limited by output 3, while ports 1 and 2 yield the wanted truth table on the entire investigated wavelength range between 1524 nm and 1568 nm. Since there is no reason to expect a degradation in the accuracy of our model for port 3, we attribute this behavior to the geometry of the device itself. As shown in Fig. 3.2, output port 3 is controlled by MZI 7, which receives one of its inputs directly from the first column of MZIs. This limited processing power due to the incomplete optical connections may explain the reduced robustness of output 3 with respect to shifts of the operational wavelength from the nominal one.

3.7 Summary

In this chapter, a comprehensive circuit-level model for MZI-based linear neurons has been developed and validated. The reference device, a 3×3 mesh of thermally-tuned MZIs fabricated on an SOI platform, was modeled to include optical propagation, losses, and thermal crosstalk effects.

The optical behavior was described using a transmission-matrix formalism, while wavelength-dependent parameters for MMIs, bent waveguides, and crossings were obtained from RSoft™ simulations. Thermal effects were modeled with COMSOL

Multiphysics® with targeted simulations and inserted in a semi-analytical model for circuit-level thermal interactions.

After fitting phase-correction parameters via PSO, the model accurately reproduced the experimental measurements, confirming its predictive capabilities. The validated model was then used as a digital twin to demonstrate Boolean programmability of the device, enabling the implementation of various logic functions and arithmetic operations. Monte Carlo and wideband analyses confirmed the robustness and broadband operation of the circuit across the C-band.

After the study of linear neurons carried out in this chapter, in the next one, the focus will be shifted to the implementation, modeling, and simulation of MRR-assisted NLAFs for neuromorphic computing applications.

Chapter 4

All-optical Nonlinear Activation Functions

After describing the behavior of a linear photonic neuron implemented with meshes of interconnected thermally-controlled MZIs and after developing a model for them and their spurious thermal interactions as we did in the previous one, in this chapter, we will analyze an implementation of all-optical NLAF, able to provide a nonlinear and programmable response thanks to its geometry made of an MZI cascaded with a RAMZI and thanks to the technological platform on which it is implemented. Indeed, as any other photonic resonant structure [107], MRRs offer enhanced nonlinear effects that can be exploited for the implementation of an all-optical NLAF. Moreover, the IMOS platform on which this device is realized offers stronger confinement and stronger nonlinear effects with respect to bulk InP and Si, respectively [108, 109].

In this chapter, the IMOS platform and its technological details will be introduced; next, the reference circuit containing the devices under test and the measurement setup that has been employed in the laboratory will be presented; then, the theoretical aspects of the MRR nonlinear effects will be explained in order to understand the mechanisms behind the experimental evidence, followed by the fitting of the measured data with a modified MRR nonlinear model that will enable the subsequent simulative analysis of the NLAF device; finally, the analysis and simulation of the complete NLAF structure will be performed.

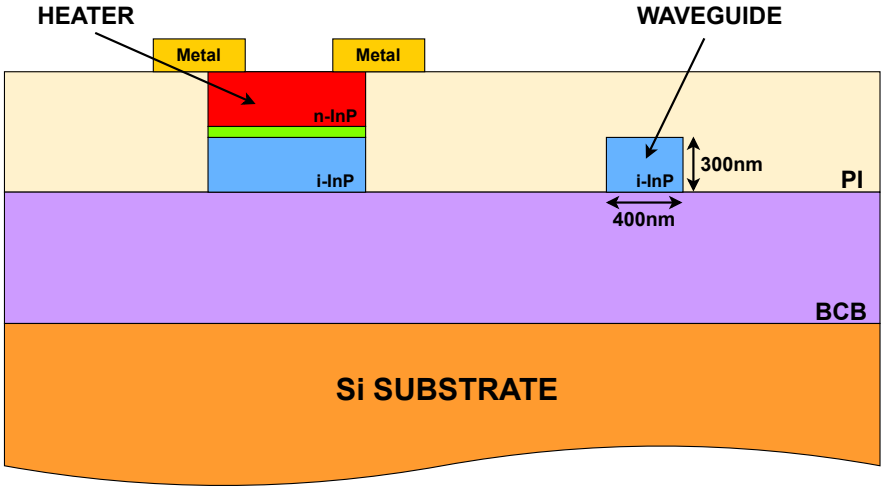

Figure 4.1 Schematic of the vertical structure of IMOS-based waveguides and ultra-compact microheaters. Each color is associated with a different material: Si in orange, BCB in lilac, intrinsic InP in blue, n-doped InP in red, PI in yellow. Green represents a layer of quantum wells [114].

The results presented in this chapter have been obtained in collaboration with the Technology University of Eindhoven and have been published in the following works [71, 110].

4.1 The IMOS platform

The reference chip that has been measured is realized on the IMOS platform, developed at the Technology University of Eindhoven. InP PICs have been gaining increasing commercial popularity, thanks to their favorable optical gain, electron velocity, and electro-optic coefficients [111], as well as the possibility of co-integrating both passive and active devices [108], thanks to the direct bandgap of InP [112]. Even though InP-based platforms are able to overcome one of the main limitations of the SOI platform (i.e. the direct integration of lasers on chip, due to the indirect bandgap of Si), bulk InP is characterized by low optical confinement, which implies the requirement of larger WG cross-sections and bending radii [113], thus hindering the achievable integration densities.

With IMOS, a thin InP membrane is bonded with a CMOS wafer via a BCB polymer layer [109]. The chips are planarized with a Polyimide (PI) layer, upon

Table 4.1 Comparison of the values of the physical parameters from the literature for InP waveguides on IMOS and Si waveguides on SOI (at 1550 nm).

Parameter	Symbol	Unit	IMOS [108]	SOI [108]
OKE coefficient	n_2	m^2/W	27×10^{-18}	$3 \times 10^{-18} - 6 \times 10^{-18}$
TPA coefficient	β_2	mW^{-1}	$14.2 \times 10^{-11} - 33 \times 10^{-11}$	$0.6 \times 10^{-11} - 1 \times 10^{-11}$
FCA coefficient	σ_{FCA}	m^2	$4 \times 10^{-21} - 7 \times 10^{-21}$	$1.24 \times 10^{-21} - 1.45 \times 10^{-21}$
FCD coefficient	σ_{r1}	m^3	–	8.8×10^{-28} [116]
Carrier lifetime	τ_{car}	s	$< 0.5 \times 10^{-9}$	$3 \times 10^{-9} - 4 \times 10^{-9}$
TO coefficient	dn/dT	K^{-1}	2.01×10^{-4} [117]	1.80×10^{-4} [118]

which the metal tracks are deposited [114]. The advantages offered by the IMOS platform are manyfold. First, BCB and PI offer good thermal insulation that is beneficial to achieve increased manufacturing quality [108, 114]. Then, thanks to the presence of the low-index BCB buffer layer, a submicrometric InP waveguide layer experiences high vertical index contrast, which enables the creation of thin 400 nm × 300 nm waveguides, with strong optical confinement [109]. Furthermore, IMOS offers good thermal insulation thanks to the insulation properties of BCB [109].

Thanks to these advantages offered by the IMOS fabrication process, it is possible to achieve devices with dimensions comparable with those of traditional SOI-based PICs, but with the InP material properties and its active-passive co-integration capabilities. An example of such miniaturized devices is the IMOS-based microheaters, which are implemented with an MMI crossing structure with a layer of n-doped InP that is biased to heat the intrinsic InP waveguide underneath [114]. These structures have an ultra-compact footprint of 97 μm^2 [114] and have the potential of reducing the area occupation of thermally-tuned devices of up to 60% [115].

Fig. 4.1 reports a schematic of the IMOS platform, with the cross-sections of a waveguide and of a microheater (simplified).

Due to its intrinsic features, the IMOS technological platform is characterized by strong light-matter interactions that make it a suitable candidate for the study of nonlinear optical effects, which can be further enhanced with resonant structures such as MRRs [107]. Among these, the most prominent ones are the Optical Kerr Effect (OKE), an optical intensity-induced nonlinear refractive index change [119], TPA, where the energy of two photons is absorbed to promote an electron from Valence Band (VB) to Conduction Band (CB) [120], thus causing free carrier-related nonlinear effects, such as Free Carrier Absorption (FCA) and Free Carrier Dispersion (FCD). FCA consists in the absorption by a free carrier in VB or CB of

a photon, thus causing intraband transitions and increased optical loss (nonlinear), with a dependence on the carrier density [121]. FCD, on the contrary, is a plasma effect that modifies the permittivity (and thus the refractive index) if the free carrier density is modified [122].

Moreover, an additional source of nonlinear change of the refractive index is linked to Self-Heating phenomena, caused by multiple physical effects, for instance FCA [123] and Shockley-Read-Hall (SRH) processes [124], as well as the linear absorption. These effects will be discussed in further detail in Section 4.3.

When dealing with an MRR, OKE and SH effects cause an increase of the refractive index, thus inducing a red-shift of the MRR resonance; FCD, instead, causes a reduction of the refractive index, inducing a blue-shift of the resonance wavelength. Table 4.1 reports the main physical parameters and coefficients (both linear and nonlinear) that are of interest for this analysis, comparing the InP values with their Si counterparts.

In the next section, the reference circuit and the laboratory setup that was employed for its measurements will be briefly described.

4.2 Reference circuit and measurement setup

Fig. 4.2 contains a picture of the mask of the chip under test fabricated on the IMOS platform. The total surface of the chip is $48\,\text{mm}^2$ ($8\,\text{mm}\times6\,\text{mm}$), but the actual usable area where components are located is $24\,\text{mm}^2$ ($6\,\text{mm}\times4\,\text{mm}$). Multiple components are contained in this chip, including both test structures (e.g., straight waveguides, MRR, racetrack MRR, MZIs with both traditional and ultra-compact TOPS, etc.) that allow users to estimate various metrics (e.g., optical losses) and actual devices (MZI+RAMZI NLAF [61], a complete photonic neuron with the Reck architecture [53], one with the Clements architecture [54], etc.). Among these devices, we have single complete neurons (linear summation section implemented with thermally-tuned MZIs followed by nonlinear activations), implemented with both traditional metallic strip TOPS and with ultra-compact ones, which allow for a smaller footprint that enables the creation of a double neuron processing unit. The devices can be accessed from the top surface of the chip via Grating Couplers (GCs) at the input and output of each component.

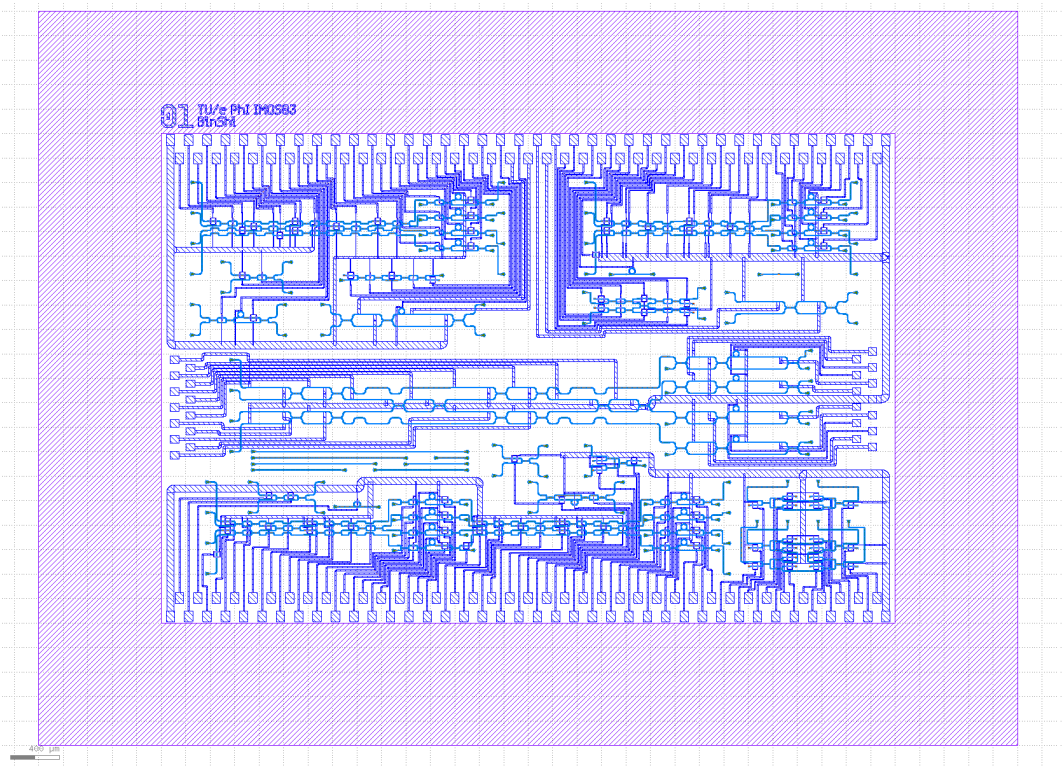

Figure 4.2 Mask of the chip under test. Its total size is 8 mm×6 mm, but the actual effective area where devices are fabricated is 6 mm×4 mm. The metal lines and pads are represented in violet, while InP waveguides are represented in blue.

During the collaboration with the Technology University of Eindhoven, the measurements focused on the test structures in order to obtain fundamental metrics of the chip (optical propagation losses and GC losses from the straight waveguides, heating power efficiency from the MZI, etc.) and to ensure the quality of fabrication, comparing the extracted metrics with the IMOS manual ones. After this, the NLAF device was measured. Future work will also involve the characterization and measurement of the complete neurons present on the chip.

A standard photonic laboratory setup was employed throughout the measurements, with the chip mounted on a water-cooled chip holder for thermal stabilization. Its schematic is reported in Fig. 4.3. The light emitted from a Keysight 81989A tunable laser is vertically coupled to the chip via GCs, at an angle of 12° to center the bell-shaped [125] GC spectral response in 1550 nm, thus maximizing the coupled optical power at the design wavelength. The same angle is used for the fiber that collects the output power. Both input and output fibers are mechanically controlled on a

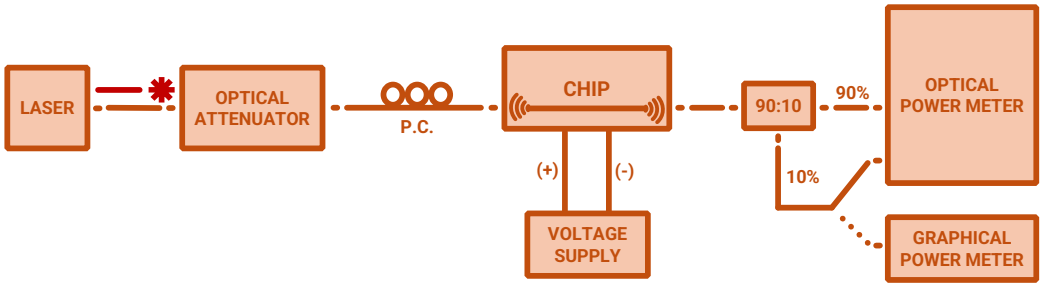

Figure 4.3 Schematic of the experimental setup used for most of the measurements on the chip under test. The light from a tunable laser is first attenuated by a VOA and controlled in polarization before being vertically injected into the chip under test via GC. The output light is then collected from the output GC and measured with an optical power meter.

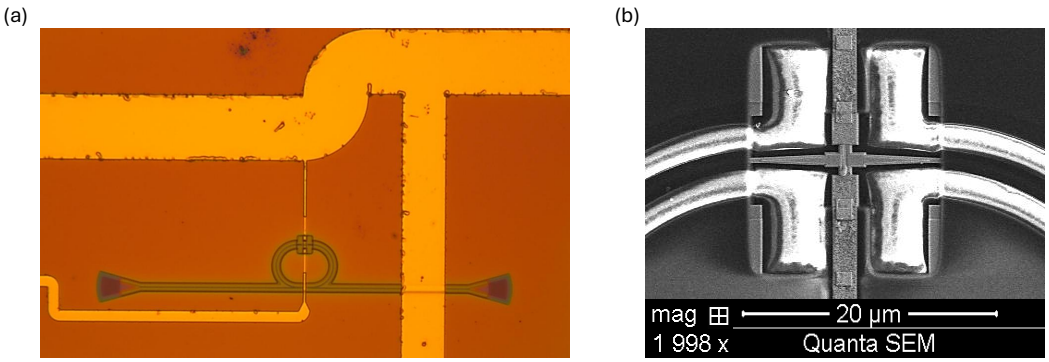

Figure 4.4 (a) Microscope image of the racetrack MRR under test. GCs and WGs are clearly visible. At the top of the MRR is also possible to appreciate the presence of the ultra-compact TOPS. (b) SEM image of the ultra-compact TOPS.

nanometer scale with a piezoelectric actuator that allows us to fine-tune their position in the 3D space to maximize the coupled and collected power. For the same reason, due to the polarization dependence of GCs [125], a Polarization Controller (PC) is included in the setup before the chip itself.

When electrical biasing is needed during the measurement procedure, the voltage is provided with a Keithley 2400 power supply through two Tungsten voltage probes in physical contact with the metal lines on the chip itself.

The light emitted by the tunable laser is first injected into a Santec OVA-100 Variable Optical Attenuator (VOA) that allows us to accurately control the power that will be injected into the chip, while keeping the power out of the laser at a fixed value, thus circumventing the limitations of the instrument itself.

The light collected at the output of the chip is routed to a multichannel Santec MPM-210H power meter. From Fig. 4.3, it is possible to observe that the light at the output of the setup is split with a 90:10 fiber splitter for monitoring purposes. Indeed, we connect the 10% splitter output to a Thorlabs PM400 graphical power meter while aligning the input and output fibers, in order to have simpler graphical feedback of the alignment accuracy. After the alignment phase, this 10% splitter output is connected to the Santec MPM-210H power meter during the measurement procedures to have the total power output by the chip.

Employing the test straight waveguides, it is also possible to isolate the GC coupling losses that will be used in subsequent phases of the analysis. These amount to $\alpha_{GC} = 6.09\,\text{dB}$. It is also possible to estimate the losses of the experimental setup before the chip to be $0.8\,\text{dB}$.

After the description of the experimental setup for the measurement of the reference chip, in the next section, we will continue with the description of the physical mechanisms that generate the nonlinear behavior of the IMOS-based MRRs, captured by the modified nonlinear model based on rate equations that was developed and employed for the fitting of the measured data.

4.3 MRR nonlinearities and nonlinear model

In this section, we will describe in detail the physical effects governing the nonlinear behavior of MRRs, which also underlie the operation of the complete NLAF device. In particular, we will focus the analysis on an IMOS racetrack MRR with $R = 20\,\mu\text{m}$ radius and $L_c = 13.3\,\mu\text{m}$ coupling length (amounting to a total ring length of $152.26\,\mu\text{m}$). An image of this reference device is reported in Fig. 4.4(a), while Fig. 4.4(b) contains a Scanning Electron Microscope (SEM) image of the ultra-compact TOPS contained in the racetrack MRR.

First, the main properties of the MRR were measured: for instance, we have obtained that the Free Spectral Range (FSR) is $\text{FSR} = 4.2\,\text{nm}$, its Extinction Ratio (ER) is $\text{ER} = 14\,\text{dB}$ at the $\lambda_0 = 1548.5\,\text{nm}$ resonance peak, its resonance wavelength is estimated to be $\lambda_{res} = 1552.525\,\text{nm}$, by measuring it at very low input power (at $< -30\,\text{dBm}$, assuming it to be the same as the unperturbed MRR). Instead, for increasing input power levels, it is observed experimentally that the resonance

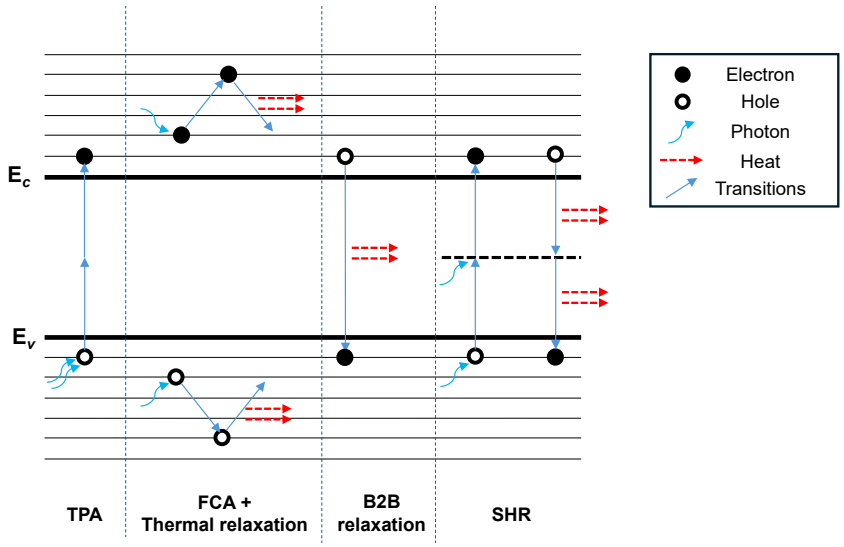

Figure 4.5 Schematic representation of the main generation/recombination processes that could affect an irradiated semiconductor. With TPA, an electron in VB gets excited as a free carrier to CB thanks to the energy of two absorbed photons; with FCA, free electrons in CB and free holes in VB (e.g., generated via TPA) absorb photons and experience intraband transitions towards higher-energy states; from these, they will be able to decay towards lower-energy states, for instance with thermal relaxation processes; free carriers can also experience non-radiative recombination processes from band to band, with consequent energy emission in the form of heat, transferred to the crystal lattice; finally, free carriers could recombine via trap-mediated SHR recombination processes, where the electrons relax to a trap level in the middle of the band gap before finally recombining with the holes in VB (both with heat transfer to the lattice).

wavelength red-shifts. Some Thermo-Optic (TO) measurements were also carried out in order to estimate the thermal tunability capabilities of the device, amounting to a thermal tuning efficiency of $\eta_{\text{tun}} = 0.9598\,\text{nm/mW}$, which is better than the η_{tun} of MRR with standard TOPSs (i.e. $0.2\,\text{nm/mW}$ or lower [126–129, 59]) and is comparable with advanced implementations (e.g., in the order of $1\,\text{nm/mW}$ for graphene-based TOPS [130]).

Then, some P_{out} vs. P_{in} measurements at different laser emission wavelengths λ_{L} have been carried out in order to create a dataset that will be used to analyze the MRR nonlinearities with an analytical model [116], which was properly adapted to describe the IMOS-based device. This modified model is analyzed in the next section.

4.3.1 MRR nonlinear model

The considered model was originally developed for the analysis of Bistability (BS) and Self Pulsation (SP) phenomena in high-Q Si MRRs [116]. Starting from the wave equations and employing CMT, it is possible to obtain two coupled rate equations, one describing the temporal evolution of the internal energy of the ring $u(t)$ in $\left[\sqrt{\mathrm{W\,s}}\right]$, one describing the temporal evolution of the free carrier density $N(t)$ in $\left[\mathrm{m}^{-3}\right]$, both related to the nonlinear effects taking place in the material (graphically summarized in Fig. 4.5). However, this model does not include a fundamental nonlinear effect, namely the SH phenomena that occur as a consequence of linear and nonlinear photon absorption. For this reason, we added a third rate equation for temperature $T(t)$ in [K] and a frequency shift term in the rate equation for $u(t)$ to the original model, for a more accurate description of the experimental evidence provided by the IMOS racetrack MRR.

$$\begin{aligned}\frac{\mathrm{d}u}{\mathrm{d}t} = &\Big\{ i\left(\xi_\omega - \xi_{\mathrm{OKE}}|u|^2 + \xi_{\mathrm{FCD}}N - \xi_{\mathrm{SH}}\right) + \\ &- \left(\alpha_{\mathrm{lin}} + \alpha_{\mathrm{TPA}}|u|^2 + \alpha_{\mathrm{FCA}}N\right)\Big\}u + \sqrt{\Gamma_{\mathrm{c}}P_{\mathrm{in}}}\end{aligned} \tag{4.1}$$

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\frac{N}{\tau_{\mathrm{carr}}} + \alpha_{\mathrm{N,TPA}}|u|^4 \tag{4.2}$$

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{1}{C_{\mathrm{th}}}\left(P_{\mathrm{abs}} - \frac{T - T_0}{R_{\mathrm{th}}}\right) \tag{4.3}$$

where ξ_ω is the frequency detuning $\xi_\omega = \omega_0 - \omega_{\mathrm{L}}$ between the wave equation reference frequency ω_0 [131] (chosen to be equal to the resonance frequency of the unperturbed, ideal MRR) and the emission frequency of the laser ω_{L}.

ξ_{OKE} represents the frequency shift caused by OKE:

$$\xi_{\mathrm{OKE}} = \frac{\omega_0}{n_0}\frac{n_2 c}{n_0 V_{\mathrm{Kerr}}} \tag{4.4}$$

with n_2 OKE coefficient, c speed of light, $n_0 = 3.165$ linear refractive index [132], and V_{Kerr} the OKE characteristic volume of interaction. Note that the $-$ sign in front of the OKE contribution in Eq. (4.1) is consistent with the red-shift that we expect from this effect [116].

ξ_{FCD} is the frequency shift caused by FCD:

$$\xi_{\mathrm{FCD}} = \frac{\omega_0}{n_0}\sigma_{\mathrm{r}} \tag{4.5}$$

with σ_{r} empirical FCD coefficient. In the model for Si MRR [116], FCD shift is modeled with the empirical equation $\sigma_{\mathrm{r1}}N + \sigma_{\mathrm{r2}}N^{0.8}$ in order to account for both free electron and hole populations. However, we opted to consider only the term $\sigma_{\mathrm{r}}N$ under the assumption that, for InP, the free electron and hole concentrations are comparable [133]. Note that the $+$ sign in front of the FCD contribution in Eq. (4.1) is consistent with the blue-shift that we expect from this effect [116, 120].

ξ_{SH} is the term that was added to the model of [116] in order to include the description of SH, due to the thermalization of carriers after the linear and nonlinear absorption of photons:

$$\xi_{\mathrm{SH}} = \frac{\omega_0}{n_0}\frac{\mathrm{d}n}{\mathrm{d}T}\left(T(t) - T_0\right) \tag{4.6}$$

with $\mathrm{d}n/\mathrm{d}T$ being the TO coefficient of InP (employing the value reported in Table 4.1) and with T_0 being the ambient temperature. This SH term is in a form conceptually similar to the one reported in [124], with the difference that it is not directly proportional to power, but it is proportional to the temperature difference, dynamically regulated by Eq. (4.3). Finally, this term appears in Eq. 4.1 with a $-$ sign to be consistent with the expected red-shift of the resonance frequency at higher input powers [120, 134]. The coefficient α_{lin} is related to the linear optical losses:

$$\alpha_{\mathrm{lin}} = \frac{\omega_0}{2Q_{\mathrm{L}}} \tag{4.7}$$

with Q_{L} loaded quality factor, normally defined as $Q_{\mathrm{L}} = \lambda_{\mathrm{res}}/\Delta\lambda_{-3\mathrm{dB}}$, where $\Delta\lambda_{-3\mathrm{dB}}$ is the Full Width at Half Maximum (FWHM) of the resonance peak in the transmission spectrum of the MRR. Q_{L} has been estimated from the spectral measurements to amount to ~ 1700, but it will be one of the parameters to be optimized during the experimental data fitting, to compensate for possible measurement uncertainty that could reduce the accuracy of this estimate.

α_{TPA} is related to the TPA nonlinear losses (the power lost because of the absorption of two photons that generate a free electron-hole pair) and it is defined as:

$$\alpha_{\mathrm{TPA}} = \frac{\beta_2 c^2}{2 n_0^2 V_{\mathrm{TPA}}} \tag{4.8}$$

with β_2 TPA coefficient and V_{TPA} the TPA characteristic volume of interaction.

α_{FCA} is related to the FCA nonlinear losses (the TPA-generated free carrier pairs absorb photons and undergo intraband transitions to higher energy states [121]) and it is defined as:

$$\alpha_{\mathrm{FCA}} = \frac{\sigma_{\mathrm{FCA}} c}{2 n_0} \tag{4.9}$$

with σ_{FCA} FCA coefficient.

Finally, Γ_{c} is a power coupling coefficient that represents the fraction of input power P_{in} coupled to the MRR through the bus WG:

$$\Gamma_{\mathrm{c}} = \frac{\omega_0}{Q_{\mathrm{e}}} \tag{4.10}$$

with $Q_{\mathrm{e}} = (1/Q_{\mathrm{L}} - 1/Q_0)^{-1}$ external quality factor, where $Q_0 = (2\pi n_0)/(\alpha \lambda_{\mathrm{res}})$ is the intrinsic Q-factor that depends on the linear loss of the waveguide α and on the MRR resonance wavelength λ_{res}.

Moving on to the terms of Eq. (4.2), τ_{carr} is the free carrier lifetime, which is related to the average time taken by a free electron-hole pair to recombine. $\alpha_{\mathrm{N,TPA}}$ is related to the generation of electron-hole pairs due to TPA (an electron absorbs two photons to obtain enough energy to be promoted to CB, with the consequent emission of a hole in VB) and is defined as

$$\alpha_{\mathrm{N,TPA}} = \frac{\beta_2 c^2}{2 \hbar \omega_0 n_0^2 V_{\mathrm{TPA}} V_{\mathrm{carr}}} \tag{4.11}$$

where $\hbar$ is the reduced Planck constant ($\hbar = \frac{h}{2\pi} = 1.054 \times 10^{-34}\,\mathrm{J\,s}$) and V_{carr} is the carrier effective volume of interaction.

For simplicity's sake, we employed the assumption that $V_{\mathrm{OKE}} = V_{\mathrm{TPA}} = V_{\mathrm{carr}} = 2\pi R A_{\mathrm{eff}}$, where A_{eff} is the effective area of the fundamental mode in the WG, computed by means of Ansys Lumerical MODE, amounting to $0.254\,\mu\mathrm{m}^2$. This assump-

tion holds thanks to the strong confinement of the fundamental mode provided by the IMOS platform [108], but also due to the fact that, for Si, these volumes have the same order of magnitude [135, 136]. Potential discrepancies between the three volumes will be accounted for when fitting the experimental data.

The optical power at the output of the MRR can be computed starting from the complex amplitude of the output light b_{out}, which is properly normalized in order to have $P_{\text{out}} = |b_{\text{out}}|^2$ [116]:

$$b_{\text{out}} = \sqrt{P_{\text{in}}} - \sqrt{\Gamma_{\text{c}}}u \tag{4.12}$$

Concerning Eq. (4.3), C_{th} is the thermal capacitance of InP that can be evaluated with

$$C_{\text{th}} = \rho c_{\text{p}} V \tag{4.13}$$

where $\rho = 4810\,\text{kg}\,\text{m}^{-3}$ is the InP density [137], $c_{\text{p}} = 310\,\text{J}\,\text{kg}^{-1}\,\text{K}^{-1}$ is the InP specific heat [138], and V is the geometric volume of the MRR; therefore, for the racetrack MRR under test, C_{th} amounts to $27.2\,\text{pJ}\,\text{K}^{-1}$. Then, P_{abs} is the fraction of absorbed power due to linear and nonlinear effects. In order to be consistent with the model of [116], it is defined as

$$P_{\text{abs}} = (\alpha_{\text{lin}} + \alpha_{\text{FCA}} N)\,|u|^2 \tag{4.14}$$

This absorbed power takes into account the thermalization of free carriers after the linear absorption and nonlinear FCA due to the TPA-generated free electrons and holes. Due to the presence of defects on the processed surface acting as non-radiative recombination centers, an SH contribution due to SRH could be considered. However, we decided to neglect this contribution, assuming that it is negligible with respect to the linear absorption coefficient. Finally, R_{th} is the thermal resistance. An accurate estimation of R_{th} could be performed with COMSOL Multiphysics®; however, we can expect it to be of the order of $1 \times 10^2 - 1 \times 10^3\,\text{K}\,\text{W}^{-1}$. This parameter will be evaluated while fitting the experimental data.

The set of Ordinary Differential Equations (ODEs) represented by Eq. (4.1), (4.2), and (4.3) is implemented in MATLAB® and solved at the steady-state employing the

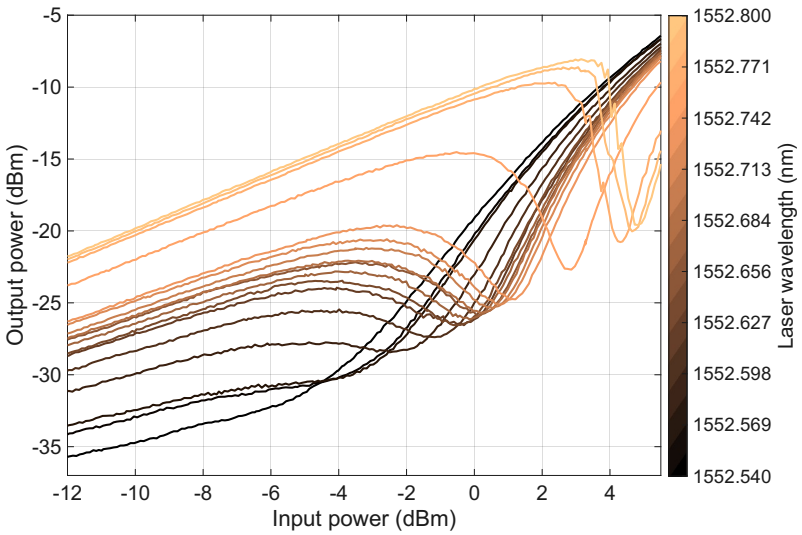

Figure 4.6 P_{in}-P_{out} MRR measurements for different values of input laser wavelength λ_{L}. Depending on the detuning with respect to the unperturbed MRR resonance wavelength λ_0, different nonlinear shapes are achieved. The reported input and output power exclude the GC coupling losses.

`fsolve` function. Since the four unknown variables of the model (i.e. $\text{Re}\{u\}$, $\text{Im}\{u\}$, N, and T) are characterized by wildly different magnitudes, in order to facilitate the convergence of the `fsolve` function, we opted to normalize these quantities according to the normalization of References [116, 61]:

$$|a|^2 = |u|^2\sqrt{\beta\sigma} \tag{4.15}$$

$$n = \sigma N \tag{4.16}$$

$$\tau = \frac{T}{T_0} \tag{4.17}$$

where the normalization constants σ and β are defined as follows:

$$\sigma \triangleq \sigma_{\text{r}}\frac{\omega_0}{n_0\Gamma_0} \tag{4.18}$$

$$\beta \triangleq \frac{c^2\beta_2}{2\Gamma_0\hbar\omega_0 n_0^2 V_{\text{TPA}}V_{\text{carr}}} \tag{4.19}$$

with $\Gamma_0 = \omega_0/Q_0$.

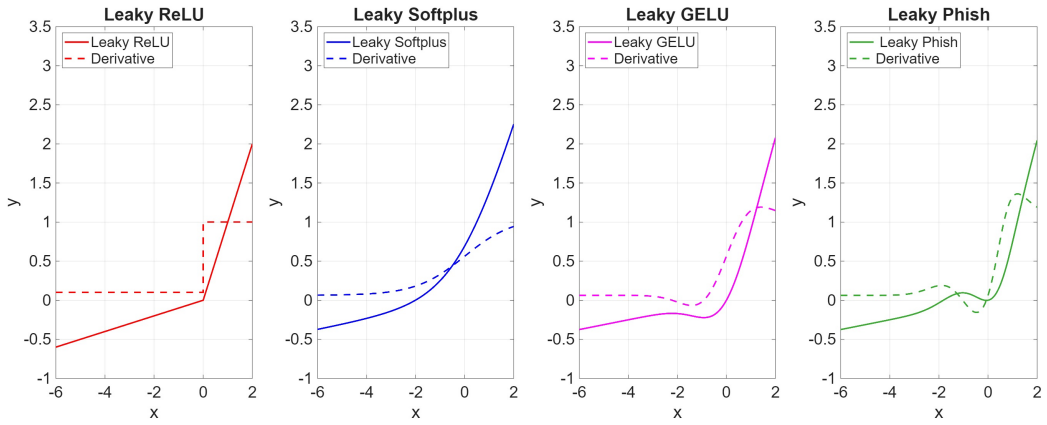

Figure 4.7 Graphical representation of the various analytical nonlinear activation functions (solid lines) that closely resemble the measured MRR P_{in}-P_{out} curves and their derivatives (dashed lines). In particular, here Leaky ReLU [139] (red), Softplus [140] (blue), Gaussian Error Linear Unit (GELU) [141] (magenta), and Phish [142] (green) and their derivatives are represented. Softplus, GELU, and Phish do not usually have a "leaky" variant, but, in order to better match the experimental curves, a $+x/10$ term was added to the equations, making them "leaky" for $x < 0$.

In order to ensure convergence, we also have to provide `fsolve` with an initial guess. For this analysis we opted to employ a "cold ring" initial condition, for which there are no preexisting internal energy nor free carrier population density ($a = 0.0 + \mathrm{i}0.0$, $n = 0.0$, and $\tau = 1$), because it is closer to how the measurements on the racetrack MRR under test have been performed. Another possibility is the "hot ring" initial condition, simulating a situation for which the ring is loaded with preexisting internal energy ($a = 1.0 + \mathrm{i}0.0$, $n = 1.5$, and $\tau > 1$ [116]). These two initial guesses give us different spectral responses, due to the presence of BS [116].

In the next subsection, the PSO-mediated fitting procedure of the measurement and its results will be discussed.

4.3.2 Experimental MRR data fitting

In this subsection, the experimental data fitting procedure will be detailed. Once again, this was done with the help of the PSO algorithm, explained in detail in Appendix A.

Fig. 4.6 shows the P_{in}-P_{out} measurements obtained from the racetrack MRR under test by applying a variable attenuation with the VOA for different values of input laser wavelength λ_{L}.

It is possible to appreciate how changing the laser wavelength λ_{L} and detuning it with respect to the reference resonance wavelength λ_0 causes a modification of the nonlinear response of the MRR (without needing any applied voltages, in a completely passive way). In particular, close to resonance, we obtain a Leaky ReLU-like response [139], characterized by two linear sections with different slopes. The experimental curves in Fig. 4.6 are actually closer to the Softplus activation function [140], which is a sort of smooth ReLU without the discontinuity. On the other hand, increasing λ_{L} with respect to λ_0, the nonlinear response will turn into a function similar to a Phish activation function [142, 16], where, for increasing P_{in}, P_{out} will first reach a relative maximum, followed by a cusp-like minimum, before starting to increase again towards a saturation power. For intermediate values of λ_{L} there will be hybrid nonlinear responses in between Leaky ReLUs and Phish functions, resembling the Sigmoid Linear Unit (SiLU) [143, 144] or any of its many variants (e.g., GELU [141]).

The following equations represent the analytical formulation of the mentioned nonlinear activation functions, which are also represented graphically in Fig. 4.7, alongside their first derivatives:

$$f_{\text{LReLU}}(x) = \max(0.1x,\, x) \tag{4.20}$$

$$f_{\text{SPlus}}(x) = \ln\left(\mathrm{e}^{x} + 1\right) \tag{4.21}$$

$$f_{\text{GELU}}(x) = \frac{x}{2}\left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{4.22}$$

$$f_{\text{Phish}}(x) = x \cdot \tanh\left(f_{\text{GELU}}(x)\right) \tag{4.23}$$

In order to limit the computational intensiveness of the optimization executed to fit the experimental data, out of all the curves of Fig. 4.6, 12 curves are selected to be part of the reference dataset and those are downsampled to 30 data points. This is done in order to reduce the computation time required to simulate each curve, but still maintaining the main information conveyed by the experimental curve (linear slope, position of maxima and minima, etc.).

Table 4.2 Target model parameters extracted with PSO to fit the experimental data and their associated ranges employed in order to limit the search space. These ranges are established based on values of the parameters that can be found in the literature [108] or based on our estimation of the parameter from the measurement themselves, as in the case of Q_L. Note that the ranges used during the PSO routine are usually slightly larger than the literature ranges, in order to facilitate convergence.

Parameter	Symbol	Unit	Range
OKE coefficient	n_2	m^2/W	$1 \times 10^{-19} - 5 \times 10^{-17}$
TPA coefficient	β_2	m/W	$1 \times 10^{-11} - 1 \times 10^{-9}$
FCA coefficient	σ_{FCA}	m^2	$1 \times 10^{-22} - 1 \times 10^{-20}$
FCD coefficient	σ_r	m^3	$1 \times 10^{-29} - 1 \times 10^{-26}$
Carrier lifetime	τ_{carr}	s	$1 \times 10^{-10} - 1 \times 10^{-8}$
Quality factor	Q_L	–	$2 \times 10^2 - 5 \times 10^3$
Thermal resistance	R_{th}	K/W	$1 \times 10^2 - 5 \times 10^4$

Table 4.3 Results of the PSO-mediated model parameter extraction. The rightmost column contains the typical values for IMOS that can be found in the literature (Table 4.1).

Parameter	Symbol	Unit	Value	IMOS
OKE coefficient	n_2	m^2/W	27.6×10^{-18}	27×10^{-18}
TPA coefficient	β_2	mW^{-1}	20.9×10^{-11}	$14.2 \times 10^{-11} - 33 \times 10^{-11}$
FCA coefficient	σ_{FCA}	m^2	9.92×10^{-21}	$4 \times 10^{-21} - 7 \times 10^{-21}$
FCD coefficient	σ_r	m^3	7.67×10^{-28}	–
Carrier lifetime	τ_{carr}	s	0.95×10^{-9}	$< 0.5 \times 10^{-9}$
Quality factor	Q_L	–	1516	Device dependent
Thermal resistance	R_{th}	KW^{-1}	627.9	Device dependent

As discussed in Appendix A, PSO is a bounded optimizer, meaning that it requires physically-meaningful ranges for each target parameter to limit the search space. In order to set ranges that are reasonable from a physics standpoint, values for the target model parameters from the literature [108] are selected as a starting point. Table 4.2 contains the parameters of the model presented in Section 4.3 that are adjusted via PSO to fit the measured data, together with their associated ranges. Note that the employed ranges are larger than those that can be found in the literature: this is done to avoid any limitation to the optimizer, thus ensuring optimal convergence. Note that, in the case of Q_L, the range is based on the estimation of the quality factor discussed in Subsection 4.3.1. Since this parameter is estimated from measurements that are inherently affected by uncertainty, Q_L is reoptimized anyway.

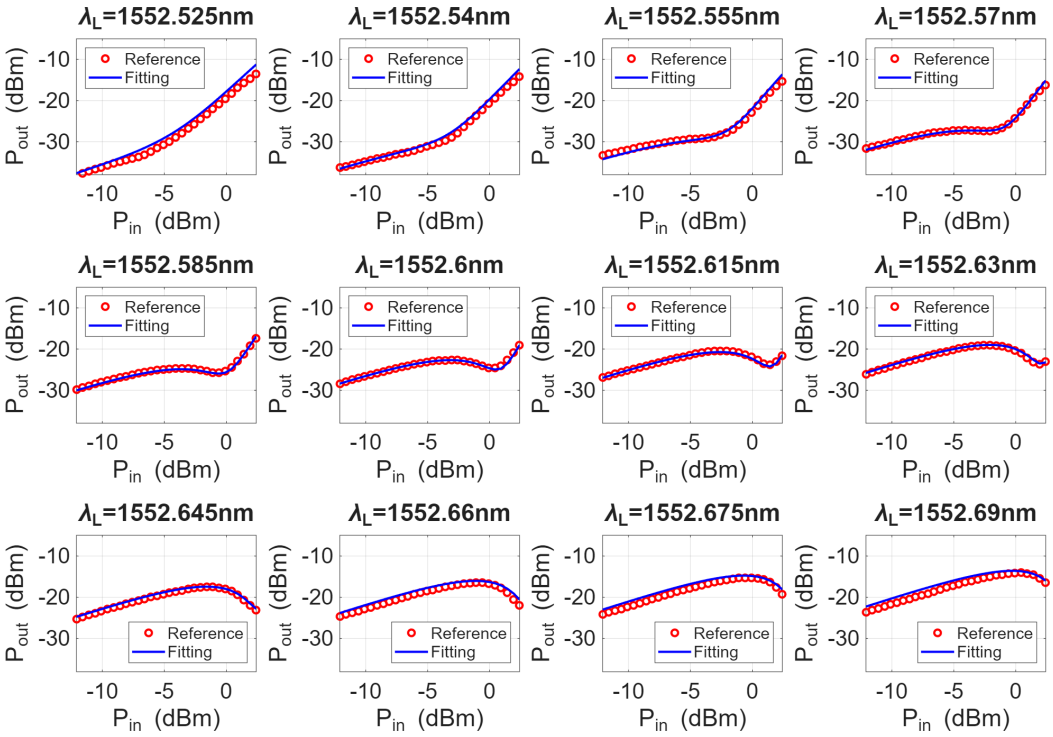

Figure 4.8 Results of the PSO-mediated fitting of the MRR measured data, for different values of input laser wavelength λ_L. The red circled curves represent the experimental MRR responses that have been downsampled and smoothed; the solid blue curves are the curves simulated by employing the extracted parameters of Table 4.3 in the nonlinear MRR model of Subsection 4.3.1. Both simulated and experimental curves have been depurated of the setup and GC losses.

Following the procedure described in Appendix A, we are able to extract the 8 target model parameters (see Table 4.2) that best match the experimental evidence (Fig. 4.6). In order to account for the finite precision of the laser used in the laboratory, a small correction factor for the reference resonance frequency was optimized as the eighth target parameter over the range ± 5 pm.

The values for the target parameters obtained from the optimization are reported in the "Value" column and they are compared with typical bulk InP or IMOS values that can be found in the literature reported under the "IMOS" column (already reported in Table 4.1). First of all, it is possible to observe that the PSO-extracted parameters (in the "Value" column of Table 4.3) are in line with the values of the corresponding parameters that can be found in the literature when present, even though the bounds set for the optimizer are much larger to favor convergence. For instance, the PSO fitting yielded an OKE coefficient n_2 which is one order of magni-

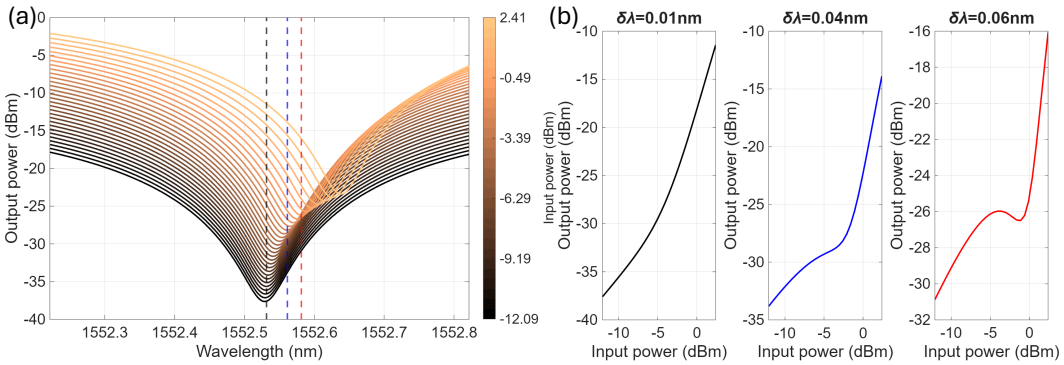

Figure 4.9 (a) Output power spectra obtained from the simulation of the parameter extracted with PSO (Table 4.3), for various values of input power. The vertical dashed lines represent the cuts at which the three nonlinear responses of Fig. 4.9(b) were taken. (b) Three examples of nonlinear P_{in}-P_{out} characteristics obtained by cutting the spectra at three different values of detuning with respect to the resonance: $\delta\lambda = \lambda_{\text{L}} - \lambda_0 = 0.01\,\text{nm}$ (black line) corresponding to a Leaky ReLU, $\delta\lambda = 0.04\,\text{nm}$ (blue line) corresponding to a GELU, and $\delta\lambda = 0.06\,\text{nm}$ (red line) corresponding to a Phish NLAF.

tude larger than that of SOI [108], meaning that this type of third-order nonlinearities will be triggered at lower powers with respect to an SOI-based device. It can also be observed that the obtained carrier recombination time τ_{carr} is compatible with typical InP values, which are in general shorter than SOI ones, due to the smaller and direct bandgap of InP. In Table 4.3, the λ_0 correction is not reported, but it amounts to $\delta\lambda_0 = -3.42\,\text{pm}$, which is in line with the wavelength accuracy of the Keysight 81989A tunable laser (equal to $\pm 5\,\text{pm}$ [145]). With this small correction, we are able to compensate for possible inaccuracy of the estimated λ_0 due to the measurement uncertainty and the finite precision of the measurement apparatus.

Fig. 4.8 reports the comparison between the 12 curves of the reference experimental dataset (red circled lines) and the corresponding curves obtained by simulating the physical model parameters extracted with PSO and listed in Table 4.3 (solid blue lines). From Fig. 4.8 it is clear that the fitted model is capable of reproducing the experimental evidence with excellent accuracy, yielding the correct nonlinear response at the given λ_{L}, with an error that never exceeds 1.8 dB. This indicates that the extracted parameters make sense from a physical standpoint, as it was implied by their resemblance to the values from the literature. In general, all the experimental responses (Leaky ReLU, Phish, GELU) and their main features (slope, location of the stationary points, etc.) are correctly predicted at the corresponding laser wavelength.

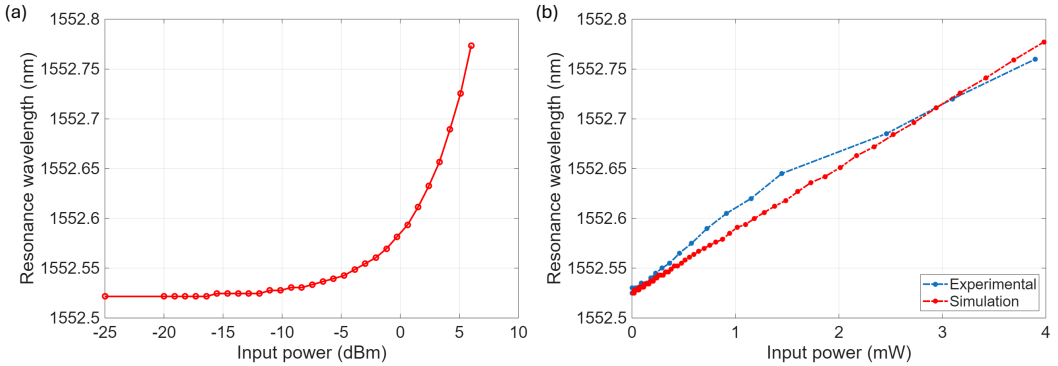

Figure 4.10 (a) Dependence of the MRR resonance wavelength on the input power P_{in} (in dBm). It can be observed that the resonance frequency is stable for low enough P_{in} values (unperturbed MRR), but, starting from approximately $-15\,\text{dBm}$, the resonance wavelength starts shifting exponentially. (b) Comparison of the dependence of the resonance wavelength on the input power P_{in} (in mW) between simulated results (red) and measured data (blue).

Since the shape of the nonlinear responses at various values of detuning of the laser wavelength λ_{L} with respect to the reference resonance wavelength λ_0 is correctly predicted, it is also implied that the simulated spectral responses at various input power levels are matching the experimental evidence (i.e. the resonance wavelength is red-shifted as the input power increases). Indeed, Fig. 4.9(a) shows that, for increasing levels of input power, the MRR resonance frequency λ_{res} shows a dependence on P_{in}, experiencing a red-shift and a reduction of the ER of the resonance peak. Fig. 4.9(b), instead, contains three examples of vertical cuts across the transmission spectra of Fig. 4.9(a), corresponding to the nonlinear P_{in}-P_{out} characteristics discussed before. These three images show the transition from Leaky ReLU-like responses to Phish ones, with GELU-like shapes in between, showing that this behavior is directly connected to the nonlinear red-shift of λ_{res} with respect to the unperturbed value λ_0. Basically, the Leaky ReLU shapes are obtained close to the reference resonance because, at low input powers, the output power increases linearly with the input since P_{in} is too low to trigger the nonlinearities that cause the resonance shift, but then, increasing the input, the slope changes because the resonance wavelength shifts. However, the behavior is still quasi-linear because the selected wavelength does not cross any shifted resonance peak. Instead, the GELU and Phish-like responses obtained farther away from the reference resonance exhibit multiple slope changes because, in this case, some of the higher input power shifted resonance peaks are crossed, resulting in lower output power and a strongly nonlinear overall P_{in}-P_{out} response.

The dependence of the MRR resonance wavelength on the input power P_{in} is exemplified in Fig. 4.10(a). Here, it can be observed that the resonance frequency does not experience any shift for low values of P_{in}, corresponding to the resonance of an unperturbed MRR. When increasing P_{in}, around approximately $-15\,\text{dBm}$, the resonance starts red-shifting nonlinearly towards longer wavelengths, as already visible from the nonlinear P_{in}-P_{out} characteristics and the output power spectra of Fig. 4.9(a). Fig. 4.10(b) instead reports a comparison of the dependence of the resonance wavelength on the input power that we obtained via simulation and the experimental counterpart. With the exclusion of measurement noise and finite precision of laboratory instrumentation, which may affect the experimental data, we can say that the model is able to predict with accuracy the resonance shift observed in the real device.

In the next subsection, the fitted MRR model will be employed to study the physical effects regulating the nonlinear behavior of the device, for instance, in order to understand which one of the nonlinear processes is contributing the most to the observed red-shift of the spectra for increasing input power levels.

4.3.3 Discussion of the MRR nonlinear physical effects

In this subsection, we further investigate the physical effects that regulate the nonlinear behavior of the MRR, exploiting the fitted model of Subsection 4.3.2 to perform targeted simulations that enable results that would be more difficult to obtain experimentally.

First, it is possible to further investigate the separate nonlinear effects and their relative magnitudes. From the discussion of Section 4.3, we expect OKE and SH to produce a red-shift of the output power spectra for increasing values of input power and FCD to cause a blue-shift. Moreover, from Fig. 4.9(a), we know that the racetrack MRR under test produces an overall red-shift of the resonance wavelength. However, it has been shown that IMOS-based MRRs provide blue-shifted spectra for increasing values of P_{in}, if the SH effects are counteracted with a Peltier cell [146]. This means that, when present, SH effects are giving a dominant contribution in generating the overall resonance red-shift, while OKE plays a secondary role with respect to FCD, since we would observe a blue-shift of the resonance if SH effects were to be avoided.

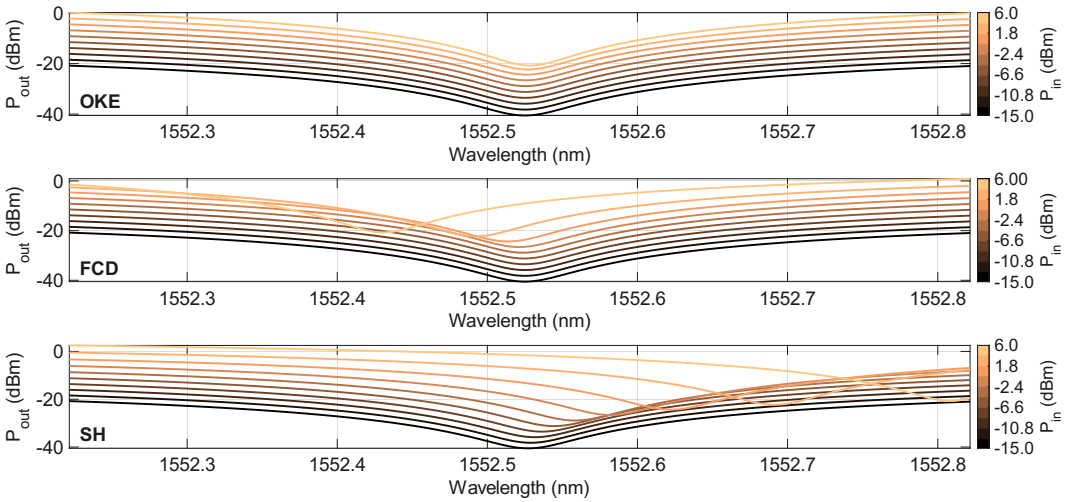

Figure 4.11 Output power spectra of the MRR for varying input power levels, obtained by simulating the model with all nonlinear effects turned off, except one at a time (OKE in the top panel, FCD in the middle one, SH in the bottom one).

We can verify this by simulating the fitted model with all nonlinear effects switched off except one at a time, so as to study the resonance shift produced by the single effects. Fig. 4.11 contains three panels showing, from top to bottom, the resonance shifts caused by OKE, FCD, and SH. All three graphs contain 10 output power spectra obtained for different values of input power (minimum $-15\,\mathrm{dBm}$, maximum $6\,\mathrm{dBm}$).

Starting from OKE, its resonance shift is contained in the top panel of Fig. 4.11. It is clear that its red-shift contribution, with the model parameters of Table 4.3, is almost negligible: indeed, by increasing the input power, the resonance wavelength is almost perfectly stable, apart from a red-shift of a couple of pm obtained for the curve with $P_{\mathrm{in}} = 6\,\mathrm{dBm}$ with respect to the unperturbed resonance wavelength (close to the resonance of the curve with $P_{\mathrm{in}} = -15\,\mathrm{dBm}$).

Continuing with FCD, whose resonance shift contribution is contained in the middle panel of Fig. 4.11, it is clear that it causes the resonance blue-shift that we expect from theory. In particular, we have that, with $P_{\mathrm{in}} = 6\,\mathrm{dBm}$, we obtain a $0.1\,\mathrm{nm}$ blue-shift with respect to the unperturbed resonance wavelength. This result is quantitatively consistent with the experimental data of the thermally-stabilized IMOS MRR with similar dimensions reported in Reference [146].

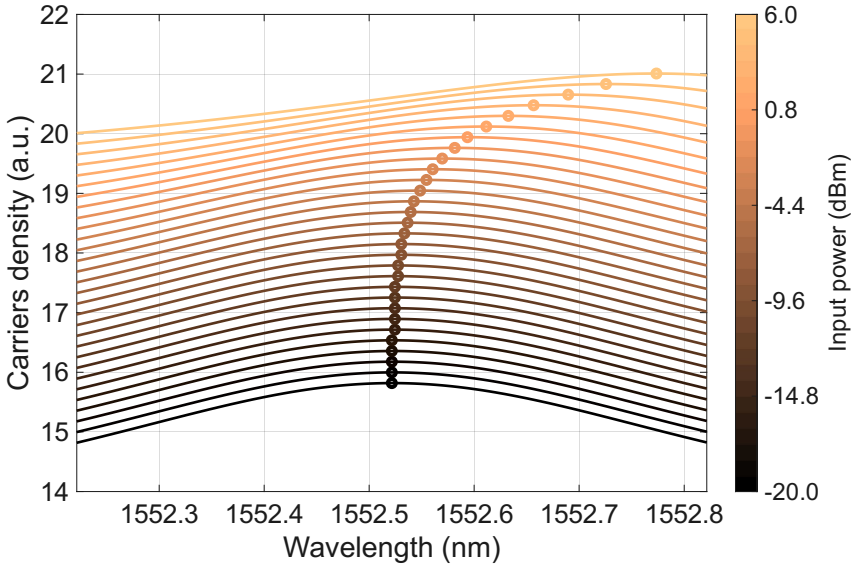

Figure 4.12 Dependence of the carrier density N on the input laser wavelength. For the sake of readability of the graph, each N curve is normalized between 0 and 1 and shifted so that the maximum is located in correspondence with the $\log_{10}$ of the maximum value.

Finally, we can analyze the SH with the bottom panel of Fig. 4.11. Here, we can observe immediately a significant red-shift of the resonance peak for high input power levels, in accordance with the theoretical expectations. For instance, with $P_{\text{in}} = 6\,\text{dBm}$, the resonance is red-shifted of 0.28 nm with respect to the unperturbed MRR resonance.

The simulated spectra of the three panels of Fig. 4.11 are in line with the theoretical expectations and the experimental data for similar devices that can be found in the literature. Indeed, the red-shift of the resonance that was obtained experimentally is mainly due to SH; the simulated fitted model also suggests that, if the thermal effects were to be counteracted, a blue-shift of the spectra would be observed, because FCD is dominant with respect to OKE. Although the general trends are perfectly reproduced (especially considering the accurately fitted curves of Fig. 4.8), to date, we do not possess the experimental data to confirm the magnitudes of the separate effects (e.g., output power spectra with the mitigation of SH effects, as in [146]). However, the reproduction of the experimental data at hand allows us to confidently assert that the model is able to correctly match the theoretical expectations and, as a consequence, the relative magnitudes of the separate nonlinear effects.

Another interesting analysis that can be performed with the fitted model, for instance, is the study of the dependence of the carrier density N with respect to the input laser wavelength, giving us insight into the generation processes occurring in the MRR.

This dependence is showcased in Fig. 4.12. Since N can assume completely different orders of magnitude depending on P_{in}, to improve readability, while maintaining the shape of the curves and a sense of their relative scale, each N curve at different P_{in} values is normalized between 1 and 0 and it is shifted so that the maximum corresponds to the non-normalized maximum value (in logarithmic scale). Fig. 4.12 also highlights the maximum value of each curve with a circle and it is possible to appreciate that the maximum value of N is located in correspondence with the resonance wavelength for that specific P_{in} value. This allows us to explain the resonant wavelength shift from a different angle: the resonant wavelength represents the wavelength for which we have the largest generation of free carriers via nonlinear effects (e.g., TPA), meaning that, for this λ, the system will also experience a higher rate of carrier-dependent phenomena, such as linear absorption, FCA, FCD, and SH.

In this section, we have analyzed at length the nonlinear behavior of an IMOS-based racetrack MRR, developing a model able to capture the interplay of physical effects responsible for this nonlinear behavior. With this model, not only are we able to reproduce the experimental evidence with accuracy, but also to gain deeper insight into the physical mechanisms characterizing the MRR with *ad-hoc* simulations to explore quantities that we would not be able to directly measure with the current laboratory instrumentation. In the next section, we will employ this nonlinear model to simulate the behavior of a racetrack MRR akin to the one analyzed up to now, which is part of a complete NLAF device for the generation of programmable nonlinear activation functions for neuromorphic applications.

4.4 NLAF structure and simulations

In this section, we will move on to the description of the complete NLAF device. First, we will describe the structure and expected behavior of the device, which was fabricated on the IMOS chip of Fig. 4.2. Then, we will describe the simulative model for the complete structure, which incorporates the nonlinear MRR model described

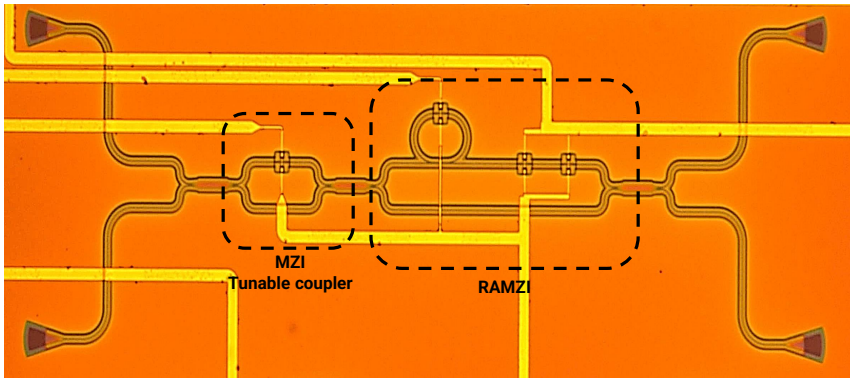

Figure 4.13 Microscope image of the NLAF device under test, realized with the IMOS technological platform. The black dashed boxes highlight the two main components in the device structure, namely the MZI coupler and the RAMZI (for which the racetrack MRR is visible on the upper arm). In the four corners of the picture, the two input and two output GCs are visible. The yellow strips are the metal lines to provide bias to the ultra-compact TOPSs.

in Section 4.3. Finally, results regarding the programmability of the NLAF device obtained with the presented model will be discussed.

4.4.1 Fabricated device on IMOS

The NLAF device considered for this analysis is part of the IMOS chip of Fig. 4.2, already discussed in the previous section of this chapter.

This device is made of an MZI tunable coupler cascaded with a RAMZI, whose racetrack MRR is equivalent to the one considered in the previous section (radius $R = 20\,\mu\text{m}$ and coupling length $L_c = 13.3\,\mu\text{m}$). The NLAF device also contains four ultra-compact TOPSs like the one in Fig. 4.4(b), one driving the MZI tunable coupler, one to potentially correct the MRR resonance, and two in parallel to modify the RAMZI arms' relative phase difference after the ring itself. A micrograph of the device under test is reported in Fig. 4.13, where the two constitutive components are highlighted with the two black dashed boxes.

The three components of this device perform the following tasks. The MRR is responsible for the nonlinear response, as explained thoroughly in Section 4.3. With its TO tunability, the MZI coupler is dedicated to controlling how much power is coupled into the upper arm of the RAMZI and, thus, into the ring: in this way, we can control the shape of the nonlinear response. Finally, the RAMZI itself provides a

degree of freedom with its thermal tunability. Indeed, whenever there is light in both arms of the RAMZI, the upper one will apply the nonlinearity through the MRR, the second one will maintain the MZI linear response. With the ultra-compact TOPS of the RAMZI, we are able to control the relative phase difference between the two arms and, consequently, the interference between the linear and nonlinear signals, with the potential of unlocking and programming additional nonlinear responses with respect to the standalone racetrack MRR.

From this point forward, the phase of the TOPS of the MZI tunable coupler will be referred to as θ, while the phase of the double TOPS of the RAMZI will be referred to as ϕ.

4.4.2 Model and simulation framework

In order to carry out the simulations of the NLAF device of Fig. 4.13, we implement an approach based on the transmission matrices obtained from CMT, similarly to what was done in Chapter 3 for the propagation through the linear MZI meshes. For this preliminary work, however, we limit as much as possible the non-idealities that– on the contrary– were considered in Chapter 3. In this case, we can avoid the inclusion of a thermal crosstalk model owing to the properties of the IMOS-based ultra-compact TOPSs (i.e. localized heating), but we assume these TOPSs to operate as point-like phase shifters. Moreover, for now, we consider the MMIs in the NLAF device to provide an ideal 3 dB splitting, without IL, and we also assume the absence of fabrication defects that could influence the working point of the MZIs. These non-idealities can be included at a later stage after a deeper experimental characterization of the NLAF device and its building blocks.

Similarly to Chapter 3, we need the transmission matrices for the building blocks of the circuit, namely MMIs and interferometric sections of MZI coupler and RAMZI. Starting from the tunable coupler, we need:

$$T_{\text{MMI}} = \begin{bmatrix} \frac{1}{\sqrt{2}} & j\frac{1}{\sqrt{2}} \\ j\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \tag{4.24}$$

$$T_{\theta} = \begin{bmatrix} e^{-\alpha L_1} e^{j\theta} & 0 \\ 0 & e^{-\alpha L_1} \end{bmatrix} \tag{4.25}$$

where α are the propagation losses in the WGs, L_1 is the length of the arms of the MZI tunable coupler, and θ is the phase shift introduced by the TOPS of the MZI coupler. For the propagation losses, we employ the nominal value for IMOS WGs, namely $\alpha = 6\,\mathrm{dB\,cm^{-1}}$, while the WG length is measured from the circuit mask to be $L_1 = 109.85\,\mu\mathrm{m}$. These two matrices correspond to the ones of Equations (3.1), (3.2), and (3.3), but without the non-idealities that were considered before. Note that, due to the absence of thermal effects, for simplicity, Eq. (4.25) does not consider the n_{eff} change caused by the TOPS, but models it directly as the phase shift θ.

Similarly, we can define a propagation matrix for the RAMZI:

$$T_\phi = \begin{bmatrix} e^{-\alpha L_2} e^{j\phi} & 0 \\ 0 & e^{-\alpha L_2} \end{bmatrix} \tag{4.26}$$

where L_2 is the length of the arms of the RAMZI and ϕ is the phase shift introduced by the TOPS of the RAMZI. The WG length is once again measured from the circuit mask and it is estimated to be $L_2 = 256.43\,\mu\mathrm{m}$.

With straightforward matrix-vector multiplications, we can propagate the input field components inside the NLAF structure. After the MZI tunable coupler, considering its two MMIs and its interferometric section, we will have:

$$\begin{aligned} E_1^{\mathrm{TC}} = {} & \frac{1}{2} e^{-\alpha L_1} \left[-E_2^{\mathrm{in}} \sin\theta + E_1^{\mathrm{in}} (\cos\theta - 1) \right] + \\ & + j \frac{1}{2} e^{-\alpha L_1} \left[E_1^{\mathrm{in}} \sin\theta + E_2^{\mathrm{in}} (\cos\theta + 1) \right] \end{aligned} \tag{4.27}$$

$$\begin{aligned} E_2^{\mathrm{TC}} = {} & -\frac{1}{2} e^{-\alpha L_1} \left[E_1^{\mathrm{in}} \sin\theta + E_2^{\mathrm{in}} (\cos\theta - 1) \right] + \\ & + j \frac{1}{2} e^{-\alpha L_1} \left[-E_2^{\mathrm{in}} \sin\theta + E_1^{\mathrm{in}} (\cos\theta + 1) \right] \end{aligned} \tag{4.28}$$

The E_1^{TC} output field component is the one entering the MRR. The response of the MRR is computed by solving the nonlinear model presented in Section 4.3 with the model parameters of Table 4.3. Let us indicate the field at the output of the MRR with the symbol E^{MRR}.

Now, we can compute the output field with the following equations, considering that light has to propagate through the interferometric section of the RAMZI and

through the output MMI:

$$E_1^{\text{out}} = \frac{1}{\sqrt{2}} e^{-\alpha L_2} \left[E^{\text{MRR}} e^{j\phi} + j E_2^{\text{TC}} \right] \tag{4.29}$$

$$E_2^{\text{out}} = \frac{1}{\sqrt{2}} e^{-\alpha L_2} \left[E^{\text{MRR}} + j E_2^{\text{TC}} e^{j\phi} \right] \tag{4.30}$$

These equations are implemented in MATLAB®, including the nonlinear MRR model solved at the steady-state with `fsolve`. In the next subsection, the simulations of the complete NLAF structure will be presented.

4.4.3 Simulated results

In this subsection, the simulated results of the complete NLAF device obtained with the model of Subsection 4.4.2 will be discussed. For this preliminary analysis, we analyze the effect of the phase shifts of the TOPSs θ and ϕ, in order to understand the programmability of the nonlinear activations with these two degrees of freedom. For now, for simplicity, the two phase shifts will be considered as additive phase terms (in radians), but, in the future, the ultra-compact IMOS TOPSs will be characterized to relate the input voltage to the phase shift experienced by the field in the WGs.

The performed simulations aim at mapping the changes of the nonlinear P_{in}-P_{out} characteristics based on the applied phase shifts θ and ϕ. The input power, injected into the upper input port, spans between -15 dBm and 3 dBm, while the θ phase shift is varied within the range $[0, \pi]$ rad and the ϕ phase shift is varied within the range $[0, 2\pi]$ rad. The latter is simulated up to a phase shift of 2π due to the inclusion of the double TOPS in the RAMZI section (see Fig. 4.13). Finally, the output power is collected at the upper output port. For now, the lower input and output ports are not employed, mimicking how the NLAF devices are operated in the complete neurons of the IMOS chip of Fig. 4.2; however, the employment of these unused ports can represent a degree of freedom to achieve additional nonlinear responses at different power levels and will be investigated in future works.

The idea is to simulate the device with different combinations of θ and ϕ, collecting the simulated output power curves, and classifying them based on the

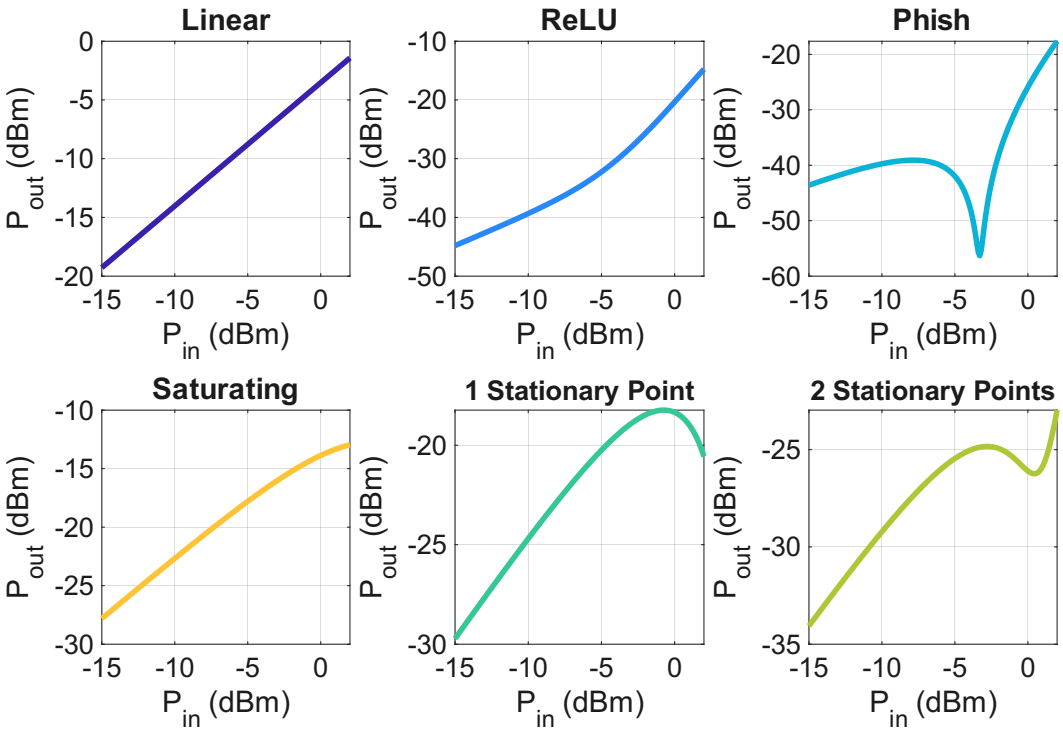

Figure 4.14 Different classes of P_{in}-P_{out} characteristics obtained from the simulation of the complete NLAF device.

retrieved nonlinear function. This could represent a helpful tool to predict the phase shift to be applied to the TOPSs in order to program the wanted nonlinear response.

By simulating the system with various (θ, ϕ) combinations, we can recognize different classes of nonlinear functions provided at the output, some of those related to the nonlinear characteristics provided by the racetrack MRR (Fig. 4.6). Indeed, in this case as well, we obtain ReLU, Phish, and purely linear curves. However, for some combinations of phase shifts, we also obtained curves that saturate at high input power, curves that present one stationary point, or curves that show two stationary points. All of these seem to belong to the same family of nonlinear characteristics, but with the nonlinearities onsetting for lower or higher values of P_{in}, thus providing a slightly different NLAF. All these P_{in}-P_{out} curves are represented in Fig. 4.14. In order to classify them automatically according to the classes reported in Fig. 4.14, the first and second derivatives of the curves are computed and analyzed.

Of course, the linear curves will be distinguished thanks to their constant first derivative. Curves with one or two stationary points can easily be recognized by counting the zeros of their first derivative (either one or two). Since both ReLUs

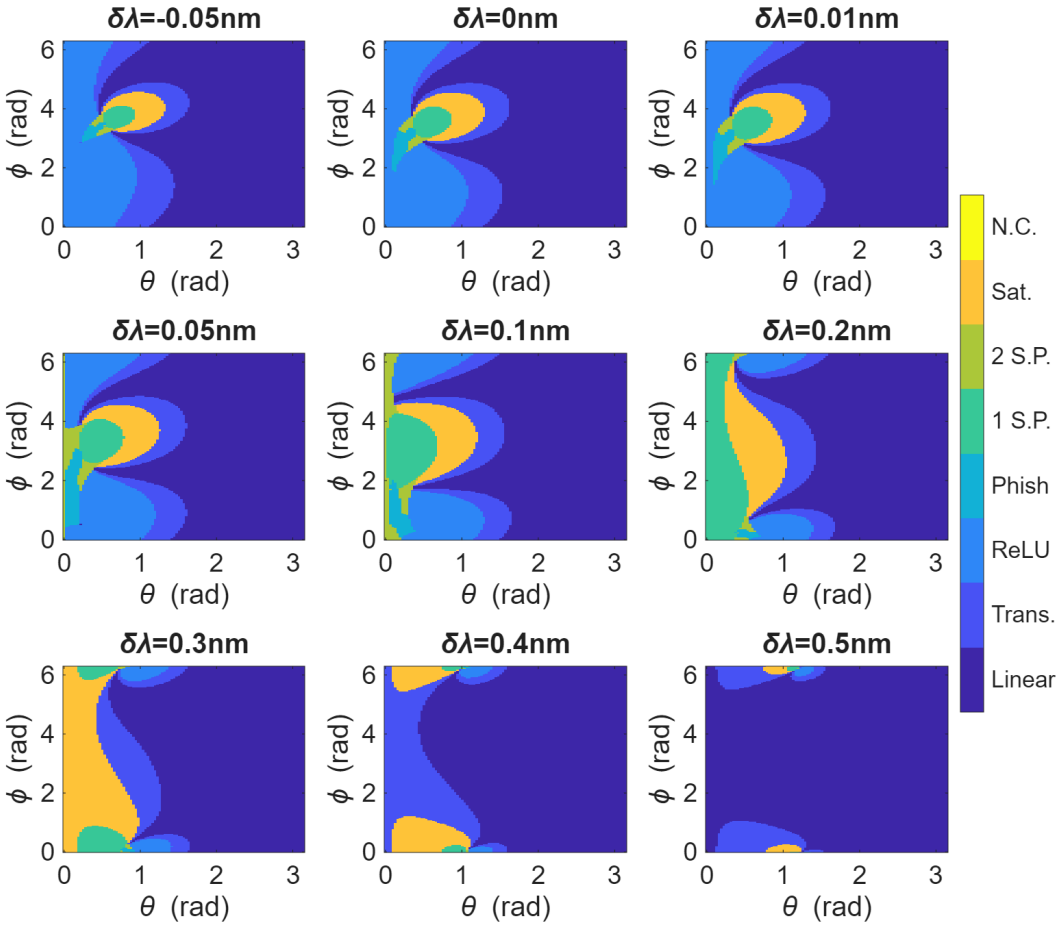

Figure 4.15 Nonlinear activation function θ-ϕ maps for different values of detuning $\delta\lambda$ with respect to the unperturbed MRR resonance. On the right-hand side of the image, a legend for the colors of the map is shown. "Trans." indicates transition curves between linear and ReLU curves, "Sat." indicates saturating curves, "S.P." means "stationary point", and "N.C." means "not classified".

and saturating curves do not have any stationary points, their first derivatives will not have any zeros, but saturating curves can be distinguished from ReLUs because their first derivative is always monotonically decreasing. Finally, Phish curves can be classified by analyzing their second-order derivative: in correspondence with the cusp minimum, the second derivative presents a positive maximum that is much higher than its minimum (always negative). If the product between the maximum and the minimum of the second derivative is negative and their difference is larger (in modulus) than a given threshold, the curve is assigned to the Phish class. Additionally, a "transition" class is defined to contain all those curves that mathematically would belong to some other class, but the nonlinearity is not strong enough, making them

effectively indistinguishable from the linear curves. Finally, an eighth class is defined to assign all those curves that might fall outside all other categories, possibly due to simulation errors ("not classified").

Fig. 4.15 contains multiple θ-ϕ maps of the activation functions, for different values of detuning $\delta\lambda$ of the input light wavelength with respect to the unperturbed MRR resonance (reported in the title of each map). These maps graphically represent the classification of the P_{in}-P_{out} curve obtained by simulating the NLAF device with a given combination of (θ, ϕ). The obtained curves have been classified according to the rules described above and the colors employed to represent the different classes are coherent with those of Fig. 4.14. From Fig. 4.15, it is immediately clear that the detuning from the MRR resonance has a noteworthy effect on the achievable nonlinear characteristics.

For negative or small positive values of detuning (i.e. $\delta\lambda = -0.05\,\text{nm}$, $\delta\lambda = 0\,\text{nm}$, and $\delta\lambda = 0.01\,\text{nm}$), we obtain similar maps with ReLUs dominating for smaller values of θ ($\theta < 1\,\text{rad}$) and linear curves dominating for higher values of θ, with transitional curves between them. For small values of θ and intermediate values of ϕ (between 2 rad and 4 rad), all the other classes of nonlinear curves (Phish, saturating, one and two stationary points) appear. The conclusions that we can draw from these three θ-ϕ maps are twofold: first, that it is possible to select different nonlinear curves only acting on the two TOPSs; second, when no phase shifts are applied and we are close to resonance, we obtain ReLU curves, because the MZI coupler routes all the light to the MRR (the MZIs of the IMOS chip of Fig. 4.2 are in bar state when no voltage is applied to their TOPS) and the RAMZI cannot apply any interference, so the NLAF will produce ReLU curves corresponding to the MRR response at resonance. These findings are in line with the preliminary measurements of the device with no bias applied, reported in Reference [71].

When increasing the input light wavelength past the MRR resonance, the θ-ϕ maps start to change, unlocking different nonlinear characteristics for the $\theta = 0\,\text{rad}$ and $\phi = 0\,\text{rad}$ (or $\phi < 1\,\text{rad}$) configuration. Indeed, in this configuration for the TOPSs, while increasing $\delta\lambda$, the NLAF device will produce Phish curves ($\delta\lambda = 0.05\,\text{nm}$), double stationary point curves ($\delta\lambda = 0.1\,\text{nm}$), single stationary point curves ($\delta\lambda = 0.2\,\text{nm}$), and stationary curves ($\delta\lambda = 0.3\,\text{nm}$). If the detuning is increased past $\delta\lambda = 0.4\,\text{nm}$, most of the θ-ϕ will yield linear or quasi-linear curves, as we might expect from the behavior of the standalone MRR. In any case, acting on

the detuning, it is possible to obtain most of the classes of nonlinear curves without having to apply any voltage to the TOPSs.

4.5 Summary

In this chapter, we discussed the implementation of an all-optical NLAF based on the IMOS technological platform.

Before dealing with the complete NLAF device, we first developed a rate equation-based model to describe the nonlinear behavior of IMOS MRRs, including OKE, TPA, FCA, FCD, and SH nonlinearities. The model has been fitted with PSO to match the measured data collected in the laboratory and has been used to investigate the physical mechanisms behind the experimental evidence. In particular, we obtained that the observed red-shift of the resonance wavelength for increasing input power is mainly related to the SH effects, while FCD-induced resonance blue-shift would dominate if the MRR were to be thermally stabilized.

Then, this nonlinear MRR model has been used to develop a simulator for a NLAF device, constructed with an MZI tunable coupler cascaded with a RAMZI, both provided with TOPSs for additional programmability. The simulator is based on CMT transmission matrices and it has been employed to investigate the programmability of the device, by changing the phase shifts θ and ϕ applied by the TOPSs. With these simulations, we have predicted that the NLAF is able to produce multiple different nonlinear P_{in}-P_{out} curves. These can be accessed both by acting on θ and ϕ (realistically, by applying a voltage to the TOPSs), but also with $\theta = 0\,\text{rad}$ and $\phi = 0\,\text{rad}$ by changing the detuning of the input light wavelength with respect to the MRR resonance. These preliminary results show how this structure is a promising all-optical, low-power option for the implementation of programmable nonlinear activation functions for photonic neuromorphic computing applications.

Chapter 5

Conclusions and outlook

5.1 Conclusions

This thesis addressed the modeling of Photonic Integrated Circuits (PICs) for neuromorphic computing applications, with a particular focus on circuit-level descriptions of both linear and nonlinear building blocks for Photonic Neural Networks (PNNs). The work is motivated by the growing computational demands of modern Artificial Intelligence systems and by the limitations of conventional electronic architectures in handling large-scale matrix operations efficiently. In this context, photonic technologies offer significant advantages due to their high bandwidth, low power consumption, and inherent parallelism.

The main goal of this thesis was to develop accurate and physically grounded models that can describe the behavior of photonic neuromorphic circuits at the device and system levels, starting from a physically accurate description of the building blocks. Particular attention was devoted to non-idealities that can strongly affect the accuracy and scalability of practical PNNs.

In **Chapter 2**, an in-depth overview of the state-of-the-art in photonic neuromorphic computing was presented. This chapter introduced the fundamental concepts of artificial neural networks and reviewed the main photonic approaches to neuromorphic computing, including Reservoir Computing (RC), Spiking Neural Networks (SNNs), and photonic Fully Connected Neural Networks (FCNNs). The analysis highlighted both the potential advantages of photonic platforms and the key technological challenges that still limit their widespread adoption. This review provided

the necessary background and clearly motivated the need for accurate circuit-level models capable of bridging device physics and system-level behavior.

In **Chapter 3**, the first major research topic of this thesis was presented, namely the modeling of linear neurons implemented using meshes of thermally tuned Mach-Zehnder Interferometers (MZIs). These structures are widely employed in photonic computing to perform weighted sums and matrix-vector multiplications. While MZI meshes are well understood from a theoretical perspective, their practical operation is strongly affected by non-ideal effects, especially thermal crosstalk, which both limits the scalability of these circuits and reduces their training accuracy in the context of neuromorphic computing. When multiple heaters are operated simultaneously, unwanted heat diffusion leads to spurious phase shifts in neighboring devices, reducing tuning accuracy and degrading the performance of the overall circuit.

To address this issue, we introduced a comprehensive circuit-level model capable of describing both optical propagation and thermal interactions within the MZI mesh. The proposed approach combines analytical models for the optical components with a compact thermal model that captures the spatial distribution of temperature variations induced by the microheaters. The proposed model was validated through direct comparison with experimental measurements obtained from a fabricated 3×3 MZI mesh with incomplete interconnections. The model showed good agreement with the measured responses across multiple input-output configurations, confirming its capability to accurately reproduce the behavior of a real circuit affected by thermal crosstalk.

In **Chapter 4**, the second main contribution of this thesis was presented, focusing on the modeling and analysis of all-optical Nonlinear Activation Functions (NLAFs), which, as in traditional neural networks, represent a critical component of PNNs. While linear operations can be efficiently implemented in the optical domain, the realization of compact, low-power, and reconfigurable nonlinearities remains one of the main challenges in photonic neuromorphic computing. The work in Chapter 4 addressed this challenge by studying a NLAF device implemented on the Indium Phosphide Membrane on Silicon (IMOS) platform, based on an MZI cascaded with a Ring-Assisted MZI.

In this chapter, first, we introduced a detailed physical model describing the nonlinear behavior of IMOS-based Microring Resonators (MRRs). The model

accounts for the dominant nonlinear effects influencing the device response, including free-carrier effects, thermal effects, and Kerr nonlinearity. Model parameters were extracted through fitting of experimental measurements using a procedure based on the Particle Swarm Optimization (PSO) algorithm. The resulting model was shown to accurately reproduce the measured input-output characteristics for different wavelength detunings and operating conditions.

Building on this MRR model, we then presented a simulation framework for the complete NLAF device. This approach enabled a systematic exploration of the device behavior as a function of its phase shifters and the detuning from the MRR resonance, showing its capability of generating a wide range of nonlinear transfer functions. In particular, depending on the operating point, the device was shown to approximate several activation functions commonly used in machine learning (e.g., Leaky ReLU). This programmability is a key feature for neuromorphic applications, where different nonlinear behaviors may be required depending on the network architecture and target task.

In conclusion, this thesis provides modeling approaches that are both physically grounded and computationally efficient for the study of photonic neuromorphic computing. The developed models support the design, analysis, and optimization of PNNs and can be extended to larger circuits and more complex architectures. While practical challenges related to scalability, variability, and integration with electronic control systems remain open, the methodologies presented in this work represent a valuable tool for the realization of robust and high-performance photonic hardware accelerators for future artificial intelligence applications.

5.2 Outlook

Future work may focus on additional experimental characterization of the NLAF device and of the 3×3 MZI mesh, to further assess the validity of the models presented in this thesis under a broader range of operating conditions. Further experimental characterization could help to evaluate the impact of factors such as fabrication variability, operating point selection, and long-term stability, while providing additional confidence in the proposed modeling approaches.

On the basis of these validated models, a possible direction for future research is the design and experimental investigation of more integrated photonic structures, such as complete photonic neurons combining linear and nonlinear functionalities. In this context, the models developed in this work could be used as supporting tools during the design and characterization phases, assisting in the interpretation of experimental results and in the identification of suitable operating regimes. Such studies would contribute to a deeper understanding of integrated photonic neuromorphic building blocks, without requiring immediate scaling to large or complex network architectures.

Appendix A

Particle Swarm Optimization

In this appendix, the Particle Swarm Optimization algorithm will be dissected. PSO was used in most of the works presented in this thesis as a tool for fitting experimental data and physical-model parameters extraction, especially when the measured responses were highly nonlinear. This happened, for instance, when adjusting the phase correction terms $\delta\varphi$ in the linear neuron model (Chapter 3) and when extracting the nonlinear MRR model for the NLAF (Chapter 4).

First, the basic algorithm and its core features will be explained; then, an advanced implementation of PSO called Adaptive Particle Swarm Optimization (APSO) will be presented, alongside its advantages with respect to the basic algorithm; finally, we showcase one of the applications in which we employed PSO for parameter extraction, i.e. the fitting of Vertical-Cavity Surface-Emitting Laser (VCSEL) power-current characteristics and small signal responses.

The contents of this appendix are based on the following published works [99, 106, 147].

A.1 The PSO algorithm

PSO is a nature-inspired, gradient-free optimization algorithm that was first developed in the 90s by James Kennedy and Russell C. Eberhart while developing a model for the social interactions within flocks of birds [98]. With few adjustments to the

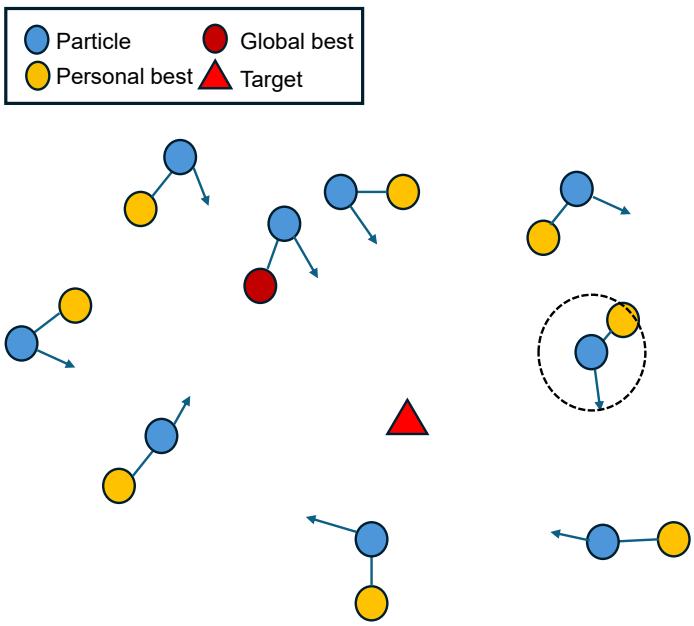

Figure A.1 Visualization of the PSO search procedure: the particles (blue dots) search their surroundings, finding new personal best positions (yellow dots) and converging towards a global best position (red dot), until convergence to the target (red triangle) is reached.

motion rules and the insertion of an optimization target, the bird flock behavior model becomes an effective optimization algorithm.

PSO is a social evolutionary algorithm, in which we have an ensemble (called "swarm") of agents (called "particles") that move in a bounded solution space to collectively reach a global minimum of the problem (e.g., minimizing an error metric, usually called "fitness"). Unlike typical evolutionary algorithms, where the agents compete with each other for survival, here the particles socially cooperate to find the best solution to the problem.

The solution space explored by the swarm is N-dimensional, where N represents the number of variables that are being optimized. The optimization performed by the swarm works thanks to the careful balance of two behaviors of the single particles: "exploration", namely the tendency of the particle to explore the space in its vicinity, and "exploitation", namely the tendency to be influenced by the position of the other particles in the swarm [148].

In order to effectively perform the optimization, the algorithm implements few simple motion rules inside the solution space. Starting from a random N-dimensional

position within the solution space, at each $k+1$-th iteration, the j-th particle will be characterized by a velocity $\mathbf{v}_j^{k+1}$, related to the exploration and exploitation tendencies of the particle itself [98]:

$$\mathbf{v}_j^{k+1} = c_\mathrm{i}\mathbf{v}_j^k + c_\mathrm{c} r_1(\mathbf{p}_j^k - \mathbf{x}_j^k) + c_\mathrm{s} r_2(\mathbf{p}_\mathrm{gl}^k - \mathbf{x}_j^k) \tag{A.1}$$

where c_i is the inertia coefficient (related to the direction of the particle at the previous iteration), c_c is cognitive acceleration coefficient (related to the exploration tendency), c_s is social acceleration coefficient (related to the exploitation tendency), r_1 and r_2 are random scaling factors, $\mathbf{p}_j^k$ is the personal best position for the j-th particle, namely the position with minimal fitness encountered by this particle up to this point of the optimization routine, and $\mathbf{p}_\mathrm{gl}^k$ is global best position, namely the position with the lowest fitness found by the entirety of the swarm. The velocity vector of Eq. (A.1) is employed to compute the position of the j-th particle at iteration $k+1$:

$$\mathbf{x}_j^{k+1} = \mathbf{x}_j^k + \mathbf{v}_j^{k+1} \tag{A.2}$$

After the new position for each particle has been evaluated with Eq. (A.1) and Eq. (A.2), the algorithm computes the fitness related to these new positions and updates personal and global best positions in case of lower errors. With these simple motion rules, the algorithm is able to optimize a large array of continuous nonlinear problems [98].

From Eq. (A.1), it is also possible to understand that finding a balance between social and cognitive acceleration coefficients (i.e. exploitation and exploration tendencies) is of paramount importance to obtain the best possible convergence: indeed, if the exploitation tendency is too strong (i.e. $c_\mathrm{s} \gg c_\mathrm{c}$), the swarm will swiftly converge to best position found at the beginning of the optimization, that is not necessarily the global optimum [149]; on the contrary, if the exploration tendency is predominant (i.e. $c_\mathrm{s} \ll c_\mathrm{c}$), the particles will continue exploring the space in the proximity of their initialization, without ever converging to a minimum as a swarm [149].

For this purpose, Kennedy and Eberhart prescribe the use of $c_\mathrm{s} = c_\mathrm{c} = 2.0$ [98], but this strategy has been shown not to be the best choice for any given problem [150]. For this reason, researchers explored possible solutions to this issue. The trivial starting point consists in performing multiple PSO optimization varying the values

of the coefficients (also called "hyperparameters"), until the best convergence is achieved statistically [99]. However, this approach is inherently time-consuming; therefore, we want to employ a method that is able to automatically establish the best hyperparameters during the optimization runtime. Among the plethora of possible approaches, for instance, we could think of including the coefficients of Eq. (A.1) in the list of target variables (i.e. the solution space will be $(N+3)$-dimensional) and optimize them with PSO itself. The solution employed in the works presented in Chapter 3 and Chapter 4 consists in using the APSO algorithm [148], where c_i, c_s, and c_c are adapted during the optimization runtime according to a set of additional rules that depend on the current convergence state of the swarm.

In the next section, the working principles of APSO will be illustrated in detail in order to explain its ability to ensure faster, more reliable convergence in the optimization of continuous, bounded nonlinear problems.

A.2 Adaptive PSO

A.2.1 Evolutionary factor and evolutionary state

The core idea behind APSO consists in adapting the optimizer hyperparameters from Eq. (A.1) (i.e. inertia c_i, social acceleration c_s, and cognitive acceleration c_c) during the optimization, according to the convergence degree of the swarm at the current iteration, called "evolutionary state". In order to have a measure of this evolutionary state, at each k-th iteration and for each j-th particle, we evaluate its average distance d_j from all the other particles of the swarm, with which we can compute the so-called "evolutionary factor":

$$f = \frac{d_\mathrm{gl} - d_\mathrm{min}}{d_\mathrm{max} - d_\mathrm{min}} \tag{A.3}$$

with d_gl average distance d_j computed for the global best particle, d_min and d_max minimum and maximum average inter-particle distances d_j in the swarm at the current iteration.

Based on the value f, we are able to perform a fuzzy assignment of the evolutionary state to four possible conditions:

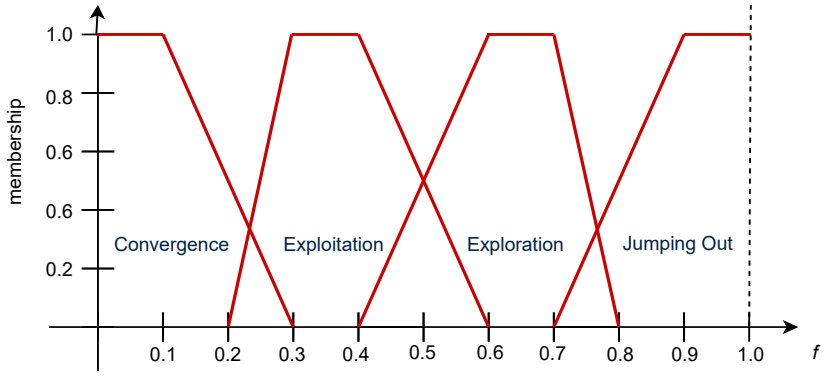

Figure A.2 Schematic of the fuzzy assignment to the four possible evolutionary states.

1. *Exploration*: the swarm is scattered almost uniformly inside the solution space while the particles explore it (medium to large f);

2. *Exploitation*: the particles are starting to converge towards a global optimum, while exploring their surroundings (smaller f);

3. *Convergence*: the swarm converged to the current global best and it is exploring its near vicinity (minimal f);

4. *Jumping-out*: a new global best has been found and it is far away from the rest because the swarm is exiting the current minimum (large f).

Fig. A.2 exemplifies the fuzzy assignment of the evolutionary state based on the evolutionary factor f. When f has a value that belongs to a single evolutionary state, that will be the assigned swarm state, while if it falls in an overlap region, the state will be assigned based on the evolutionary state during the previous iteration of the optimizer (according to the progression exploration > exploitation > convergence > jumping-out > exploration > ...).

Based on the assigned evolutionary state, the PSO hyperparameters can be modified to ensure the best possible convergence.

A.2.2 Velocity coefficients adaptation

First, the inertia coefficient is initialized to $c_{\mathrm{i}} = 0.9$ and, at each iteration, it is modified according to the following heuristic rule [148]:

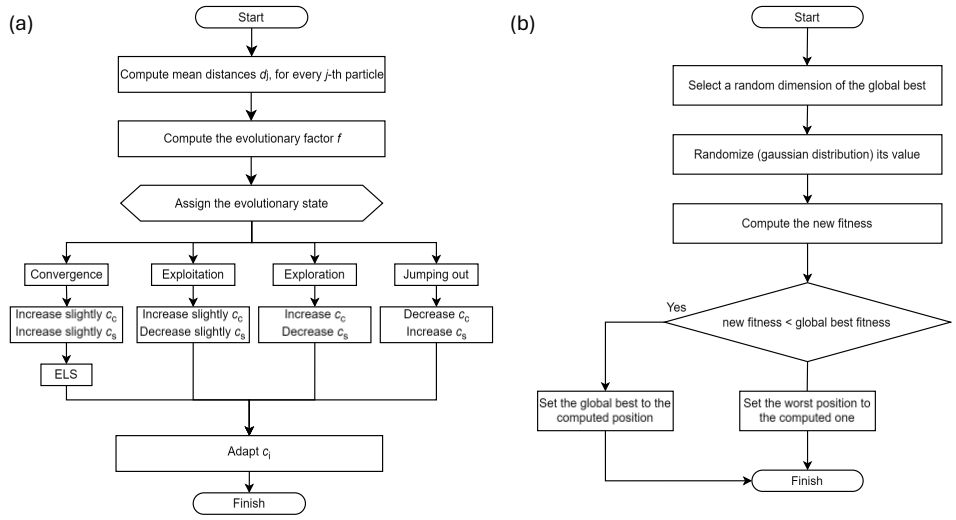

Figure A.3 (a) Flowchart for the adaptation of the velocity coefficients, performed at each iteration of APSO. (b) Flowchart for ELS, performed at each iteration in which the swarm is in convergence state.

$$c_i(f) = \frac{1}{1 + 1.5\exp(-2.6f)} \tag{A.4}$$

This rule is beneficial because it produces a small c_i when the swarm is in exploitation or convergence state to promote local search close to the found optimum and a larger c_i when we are in jumping-out or exploration state to benefit global search of an optimum.

Moving on to the social and cognitive acceleration coefficients, starting from the values $c_\mathrm{c} = 2.0$ and $c_\mathrm{s} = 2.0$, Zhan *et al.* [148] give the following "recipe" to adapt them based on the current evolutionary state:

1. *Exploration* $\rightarrow$ *increase* c_c*, decrease* c_s: in this case, the swam is scattered across the solution space, so we want to promote single particle exploration, reducing the pull towards the current global best, since it may be a local optimum, especially at the beginning of the optimization;

2. *Exploitation* $\rightarrow$ *increase* c_c *slightly, decrease* c_s *slightly*: in this state, usually the swarm has yet to find a global optimum; therefore, we want the swarm to keep on exploring without converging to the current global best position;

3. *Convergence* $\rightarrow$ *increase* c_c *slightly, increase* c_s *slightly*: in this state, the swarm found an optimum in the solution space, so we want the particles to converge there, while still maintaining the personal search of new personal and global best positions;

4. *Jumping-out* $\rightarrow$ *decrease* c_c*, increase* c_s: in this situation, we want the swarm to move to the newly-found global best position in order to start exploring that region of solution space.

In order to increase or decrease c_c and c_s, we employ a random value δ uniformly generated between 0.05 and 0.1 (when we have to increase or decrease "slightly", 0.5δ is used). Since the modification of the coefficients is weighted by a random coefficient and it depends on the random movement of the swarm across the solution space, we also want to have a method to control the new values of the coefficients, so as to avoid premature or delayed convergence. For this reason, the acceleration coefficients are clamped to the range $[1.5,\ 2.5]$ and, if $c_\mathrm{c} + c_\mathrm{s} \geq 4.0$, they are normalized with the following rule [148]:

$$c_k = \frac{c_k}{c_\mathrm{c} + c_\mathrm{s}} 4.0 \tag{A.5}$$

with $k = \{\mathrm{c},\ \mathrm{s}\}$.

The velocity coefficients adaptation procedure is summarized in the flowchart of Fig. A.3(a).

A.2.3 Elitist Learning Strategy

The Elitist Learning Strategy (ELS) is an empirical motion rule implemented to encourage the global best particle to exit the local optimum when the swarm is in convergence state, in order to allow the search of possibly better solutions to the problem [148]. This procedure is performed when and only when the swarm is in convergence state.

The core idea of ELS is to randomly select one of the dimensions of the global best position $\mathbf{p}_\mathrm{gl}$ and apply a Gaussian perturbation of its value. If j is the randomly selected dimension of $\mathbf{p}_\mathrm{gl}$, the global best position will be perturbed as follows:

$$\mathbf{p}_{\mathrm{gl},j} = \mathbf{p}_{\mathrm{gl},j} + (X_j^{\max} - X_j^{\min}) \cdot Gaussian(\mu, \sigma^2) \tag{A.6}$$

where $[X_j^{\min}, X_j^{\max}]$ are the bounds of the problem in the j-th dimension, $\mu = 0$ is the mean of the Gaussian random variable, and σ is its standard deviation, defined as [148]

$$\sigma = \sigma_{\max} - (\sigma_{\max} - \sigma_{\min}) \frac{g}{G} \tag{A.7}$$

where $\sigma_{\max} = 1.0$, $\sigma_{\min} = 0.1$, g is the current optimizer iteration, and G is the maximum iteration number.

After generating this ELS position $\mathbf{p}_{\mathrm{ELS}}$, its fitness is evaluated. If it is smaller than the $\mathbf{p}_{\mathrm{gl}}$ fitness, the ELS position will be saved as the new global best position, otherwise it will be assigned to the particle with the largest fitness (worst position of the swarm).

The main steps of the ELS procedure are summarized in the flowchart of Fig. A.3(b).

In this section, the APSO algorithm was described, with particular attention to the procedures for the adaptation of the hyperparameters of the base PSO algorithm. In the next section, a direct application of the optimizer for the parametric characterization of VCSELs from their measurements will be illustrated.

A.3 Applications: VCSELs

In this section, a direct application of APSO for the extraction of physical-model parameters of VCSELs will be presented in detail.

VCSELs are gaining increasing popularity due to their simple fabrication process, low production cost, and versatility, making them a suitable choice for a wide range of applications such as short-reach data center communications, optical sensing, and various industrial uses [151–153].

However, VCSELs are characterized by a complex physical behavior due to nonlinearities, strong thermal dependencies, and other effects (e.g., spatial hole burning), making them difficult to model effectively. PSO can be employed in order

to efficiently extract a set of parameters that are compatible with a model for VCSEL physics, either in conjunction with a ML agent [154], or directly, without requiring the generation of large datasets for AI [105].

In this case, the model of choice is a rate equation-based model [155], which also includes thermal effects and which is implemented in Synopsys Optsim™ [156]. The idea is to use PSO to extract parameters compatible with this model, in order to enable subsequent system-level simulations in the simulation tool, using a realistic digital twin of the experimental VCSEL.

A.3.1 VCSEL model

The reference model captures the physics of VCSELs by means of carrier and photon rate equations that describe their temporal evolution, while modeling thermal effects with a set of empirical equations [155].

For simplicity, the explicit spatial dependency of the carrier and photon densities is neglected by assuming a cylindrical geometry without azimuthal variations and the carrier distribution along the radial coordinate r is described through a two-term Bessel series expansion [155]:

$$N(r,t) = N_0(t) - N_1(t) J_0\left(\sigma_1 \frac{r}{R}\right) \tag{A.8}$$

with σ_1 first nonzero root of J_1, J_0 and J_1 Bessel functions of the first kind, and R effective radius of the active layer. $N_0(t)$ and $N_1(t)$ are the first two expansion coefficients of the carrier number [155] and their temporal evolution is given by the following spatially independent rate equations:

$$\frac{\mathrm{d}N_0}{\mathrm{d}t} = +\frac{\eta_\mathrm{i} I}{q} - \frac{N_0}{\tau_\mathrm{n}} - \frac{I_1(N_0,T)}{q} - \frac{G(T)\left[\gamma_{00}(N_0 - N_\mathrm{tr}(T)) - \gamma_{01} N_1\right]}{1+\varepsilon S} S \tag{A.9}$$

$$\frac{\mathrm{d}N_1}{\mathrm{d}t} = -\frac{N_1}{\tau_\mathrm{n}}(1+h_\mathrm{diff}) + \frac{G(T)\left[\phi_{100}(N_0 - N_\mathrm{tr}(T)) - \phi_{101} N_1\right]}{1+\varepsilon S} S \tag{A.10}$$

where η_i is the injection efficiency, I is the injected current, q is the electron charge, τ_n is the carrier lifetime, T is the temperature, $G(T)$ is the gain coefficient, $N_\mathrm{tr}(T)$ is the transparency carrier number, $I_1(N_0,T)$ is the leakage current, ε is the gain

compression factor, h_{diff} is the diffusion coefficient. The coefficients γ_{00}, γ_{01}, ϕ_{100}, ϕ_{101} quantify the overlap between the fundamental transverse mode (with Gauss-Laguerre profile) and the active region [157].

The temporal evolution of the photon population density S is described by an additional rate equation:

$$\frac{\mathrm{d}S}{\mathrm{d}t} = -\frac{S}{\tau_{\text{p}}} + \frac{\beta_{\text{sp}} N_0}{\tau_{\text{n}}} + \frac{G(T)\left[\gamma_{00}(N_0 - N_{\text{tr}}(T)) - \gamma_{01} N_1\right]}{1 + \varepsilon S} S \tag{A.11}$$

with τ_{p} photon lifetime and β_{sp} spontaneous emission coefficient.

It is possible to compute the output power P_{out} starting from S with a proper coupling coefficient k_{f}:

$$P_{\text{out}} = k_{\text{f}} S \tag{A.12}$$

An additional rate equation is employed to estimate the device temperature and to take into account the thermal dependencies [158]:

$$T = T_{\text{amb}} + (IV - P_{\text{out}}) R_{\text{th}} - \tau_{\text{th}} \frac{\mathrm{d}T}{\mathrm{d}t} \tag{A.13}$$

with V applied voltage, T_{amb} ambient temperature, R_{th} thermal impedance, and $\tau_{\text{th}} = 1\,\mu\text{s}^{-1}$ thermal time constant.

The thermal phenomena affecting the device are described via three empirical equations for the temperature-dependent gain coefficient $G(T)$, transparency carrier number $N_{\text{tr}}(T)$, and leakage current $I_{\text{l}}(N_0, T)$ [155]:

$$G(T) = G_0 \frac{a_{\text{g}_0} + a_{\text{g}_1} T + a_{\text{g}_2} T^2}{b_{\text{g}_0} + b_{\text{g}_1} T + b_{\text{g}_2} T^2} = G'_0 \frac{1 + a'_{\text{g}_1} T + a'_{\text{g}_2} T^2}{1 + b'_{\text{g}_1} T + b'_{\text{g}_2} T^2} \tag{A.14}$$

$$N_{\text{tr}}(T) = N_{\text{tr}_0}\left(C_{\text{n}_0} + C_{\text{n}_1} T + C_{\text{n}_2} T^2\right) = N'_{\text{tr}_0}\left(1 + C'_{\text{n}_1} T + C'_{\text{n}_2} T^2\right) \tag{A.15}$$

$$I_{\text{l}}(N_0, T) = I_{\text{l}_0} \exp\left(\frac{-a_0 + a_1 N_0 + a_2 N_0 T - a_3 / N_0}{T}\right) \tag{A.16}$$

Table A.1 Investigated VCSEL model parameters, with their associated range for PSO and their unit of measurement.

Parameter	Range	Unit
Injection efficiency η_{i}	$0.70 - 1.00$	
Power coefficient k_{f}	$10.00 - 60.00$	nW
Carrier lifetime τ_{n}	$0.50 - 5.00$	ns
Photon lifetime τ_{p}	$1.50 - 3.50$	ps
Gain coefficient G'_0	$-360.0 - -11.1$	ms^{-1}
Gain coefficient a'_{g_1}	$-5.00 - -0.50$	kK^{-1}
Gain coefficient a'_{g_2}	$-50.00 - -2.00$	kK^{-2}
Gain coefficient b'_{g_1}	$-100 - 0$	kK^{-1}
Gain coefficient b'_{g_2}	$5.56 - 900.0$	kK^{-2}
Gain saturation factor ε	$1 \times 10^{-6} - 3 \times 10^{-6}$	
Transparency number N'_{tr_0}	$2.00 \times 10^{6} - 1.00 \times 10^{8}$	
Transparency number coefficient C'_{n_1}	$-100.00 - -1.00$	kK^{-1}
Transparency number coefficient C'_{n_2}	$0.00 - 100.00$	kK^{-2}
Leakage current factor I_{l_0}	$1.00 - 2.00$	A
Leakage current coefficient a_0	$2.00 \times 10^{3} - 1.00 \times 10^{4}$	K
Leakage current coefficient a_1	$0.00 - 3.00 \times 10^{-4}$	K
Leakage current coefficient a_2	$1.00 \times 10^{-9} - 4.00 \times 10^{-8}$	
Diffusion parameter h_{diff}	$1.00 - 20.00$	
Thermal impedance R_{th}	$5.00 \times 10^{2} - 8.00 \times 10^{3}$	K/W
Spontaneos emission coeff. β_{sp}	$1 \times 10^{-5} - 9 \times 10^{-6}$	

In Eq. (A.14) and Eq. (A.15), the rightmost expressions are introduced to substitute those from [155] to make sure that the parameters involved are independent from each other and uniquely defined [99].

A.3.2 VCSEL parameter extraction

The target parameters for the VCSEL model that we want to extract with PSO are listed in Table A.1, together with their search ranges for the optimization.

The reference that we are trying to match during the optimization is a set of power-current characteristics LI (the laser output power for varying values of driving current), for different values of external room temperature, and a set of small-signal modulation S21 responses, for different values of room temperature and bias current. With this set of curves, we are able to capture both the thermal dependencies and the dynamic behavior of the VCSEL under test.

Another fundamental component of the PSO algorithm is the fitness, which, as explained in Section A.1, is the measure of how far a solution found by a particle

is from the target. In order to compute the fitness for this particular problem, we compute the average sample-by-sample MSE for LI and S21 curves separately. For the j-th particle, at the k-th step, we can compute

$$\mathrm{MSE}_{\mathrm{LI},j}^{k} = \frac{1}{M_{\mathrm{LI}}} \sum_{m=1}^{M_{\mathrm{LI}}} \frac{||P_{\mathrm{target}}^{m} - P_{\mathrm{PSO},j}^{m,k}||}{||P_{\mathrm{target}}^{m}||} \tag{A.17}$$

$$\mathrm{MSE}_{\mathrm{S21},j}^{k} = \frac{1}{M_{\mathrm{S21}}} \sum_{m=1}^{M_{\mathrm{S21}}} \frac{||S21_{\mathrm{target}}^{m} - S21_{\mathrm{PSO},j}^{m,k}||}{||S21_{\mathrm{target}}^{m}||} \tag{A.18}$$

where $||\cdot||$ indicates the norm-2 of the vectors containing the power P and small signal response $S21$ data samples. M_{LI} and M_{S21} respectively indicate the number of LI and S21 curves considered in the reference, for which the average errors $\mathrm{MSE}_{\mathrm{LI},j}^{k}$ and $\mathrm{MSE}_{\mathrm{S21},j}^{k}$ are computed.

Finally, the fitness f_j^k is computed as the arithmetic average of $\mathrm{MSE}_{\mathrm{LI},j}^{k}$ and $\mathrm{MSE}_{\mathrm{S21},j}^{k}$; during the optimization, the quantity $f^k = \min_j f_j^k$ is minimized. This makes it a so-called "joint optimization" approach [159], which can be seen as an implicit Multi-Objective Optimization (MOO), since it optimizes multiple objectives reduced to a single fitness function, instead of optimizing a trade-off for the fitness functions of each objective.

The procedure of extracting VCSEL model parameters has been performed both on synthetic data (i.e. simulated LI and S21 curves directly using the rate equation model) and experimental data (i.e. measurements of an unknown device to be characterized).

Starting from the first case, we employed a set of $M_{\mathrm{LI}} = 3$ LI curves (at 25 °C, 40 °C, and 55 °C) and $M_{\mathrm{S21}} = 6$ S21 responses (at 2 mA, 4 mA, and 6 mA for $T =$ 25 °C, at 2 mA and 4 mA for $T = 40$ °C, and at 5 mA for $T = 55$ °C), generated by solving the rate equations with a known set target parameters. Each curve contains 16 samples, for a total of 144 data points that the algorithm has to reproduce by minimizing the fitness [99].

For this first work, the basic PSO implementation has been employed. In order to ensure optimal convergence (without the automatic adaptation of the coefficients), we opted to perform a hyperparameter optimization by re-running PSO, modifying

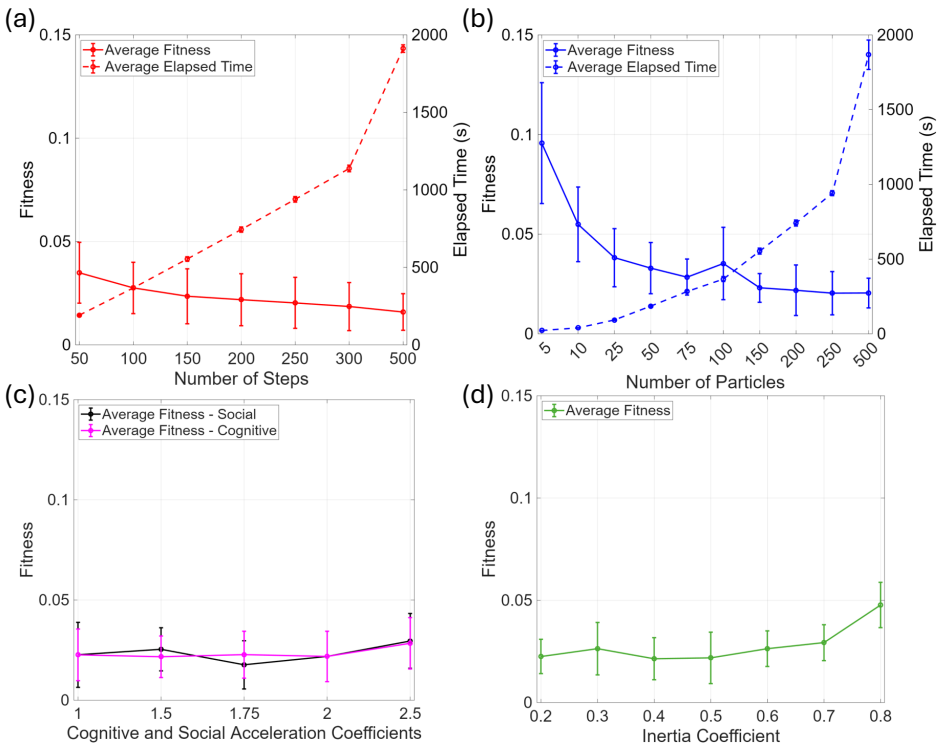

Figure A.4 (a) Analysis of the dependency of fitness and computation time on maximum number of steps N_s. (b) Analysis of the dependency of fitness and computation time on number of particles N_p. (c) Analysis of the dependency of fitness on social c_s and cognitive c_c acceleration coefficients. (d) Analysis of the dependency of fitness on the inertia coefficient c_i.

one hyperparameter at a time, until finding an optimum. With this procedure, we are able to set $c_i = 0.4$, $c_s = 2.0$, $c_c = 2.0$ (in line with the values prescribed by Kennedy and Eberhart [98]), with $N_p = 200$ particles and for $N_s = 200$ iterations. Fig. A.4 visually shows the results of the hyperparameter analysis that led to these choices.

Fig. A.5(a) and Fig. A.5(b) contain the comparison between reference LI and S21 curves and the PSO predictions. Clearly, the algorithm is able to extract a set of model parameters that can reproduce the synthetic target extremely accurately.

Due to the randomness of the movement of the particles, we also wanted to show that these accurate parameter extraction results are reproducible and generalizable to different VCSEL specimens. For this purpose, we generated 100 sets of model parameters, with which we generated just as many sets of reference curves for the PSO to be applied to. PSO performances are compared to two built-in algorithms

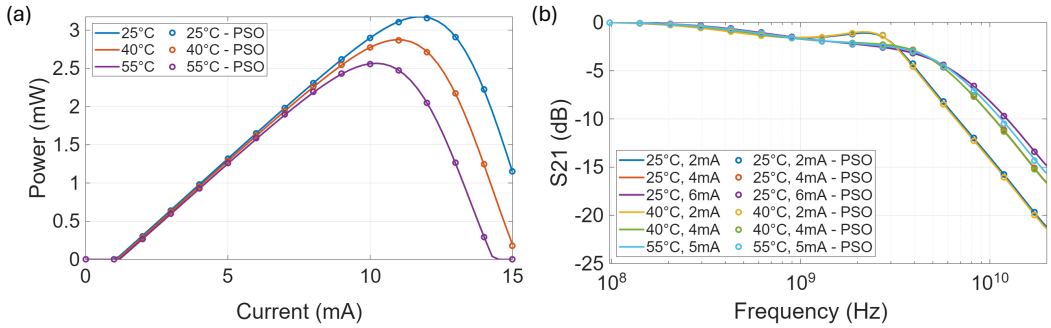

Figure A.5 (a) Comparison between the synthetic LI reference curves and the LI curves obtained with the PSO predictions. (b) Comparison between the synthetic S21 reference responses and the S21 curves obtained with the PSO predictions.

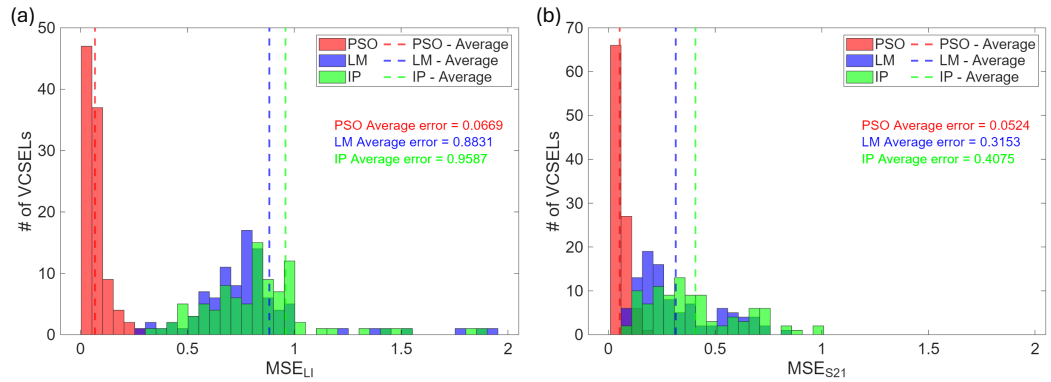

Figure A.6 (a) Distributions of the average error for the LI characteristics of 100 simulated devices. (b) Distributions of the average error for the S21 responses of 100 simulated devices. The distributions are obtained using the custom PSO algorithm (red), the Levenberg–Marquardt optimizer (blue), and the Interior Point optimizer (green). The dashed vertical lines represent the mean values of the corresponding distributions.

from the MATLAB® Optimization Toolbox [160], namely the Levenberg-Marquardt (LM) and the Interior Point (IP) algorithms.

The results of this statistical analysis are reported in Fig. A.6(a) and Fig. A.6(b). From these, it is clear that our PSO methodology has much more consistent results with respect to the MATLAB® built-in optimizers, especially with regard to the LI predictions, where LM and IP clearly struggle when predicting these nonlinear characteristics.

It is possible to argue that the excellent predictions presented thus far are facilitated by the synthetic reference curves: indeed, they have been generated with the same model that we employ in the PSO methodology, yet experimental curves

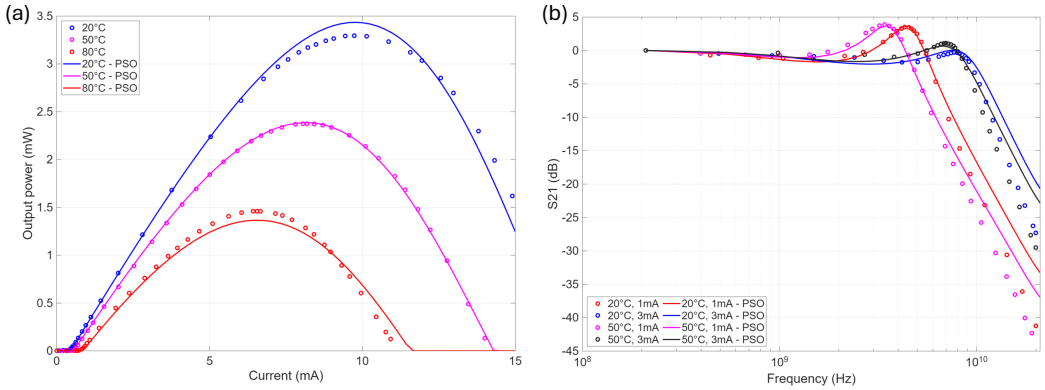

Figure A.7 (a) Comparison between the experimental LI reference curves [161] and the LI curves obtained with the APSO predictions. (b) Comparison between the experimental S21 reference responses [162] and the S21 curves obtained with the APSO predictions.

could be affected by additional physical phenomena or sources of noise that are not included in the rate equations. For this reason, this PSO methodology is applied on experimental LI and S21 measurements [161, 162] of an unknown device to be characterized. In this case, we apply the APSO algorithm to the measurements of a GaAs/AlGaAs 850 nm VCSEL, characterized by a 1λ-cavity with three 8 nm GaAs quantum wells [161, 162]. The set of reference curves is made up of $M_{\mathrm{LI}} = 3$ LI curves (at 20 °C, 50 °C, and 80 °C) and $M_{\mathrm{S21}} = 4$ S21 responses (1 mA and 3 mA, for both $T = 20\,^{\circ}\mathrm{C}$ and $T = 50\,^{\circ}\mathrm{C}$).

Fig. A.7(a) and Fig. A.7(b) contain the comparison between the experimental LI and S21 reference curves and the PSO predictions. As it was expected, there is a reduction in the accuracy of the predictions obtained while trying to match this set of experimental curves, since these curves will be affected by multiple non-ideal effects (e.g., noise, measurement uncertainty, etc.) that cannot be directly taken into account with the rate equation model. In order to give the optimizer the best chance to reach a good solution, the curves have been smoothed and sampled non-uniformly, so as to have more points in critical parts of the curve (e.g., for the LI curves, more points near the threshold and near the thermal roll-off point).

Despite the minor discrepancies between the reference data and the curves obtained with the APSO-generated parameters, it is possible to appreciate that we successfully captured the threshold, slope, and thermal roll-off of the LI curves, as well as the peak frequency and the −3 dB bandwidth of the S21 responses.

A.4 Summary

In conclusion, in this appendix, we presented in detail not only the PSO algorithm, with its inner workings and its capabilities, but also its improved version called APSO, which is able to address some of the limitations of the original implementation, especially the optimization of the PSO hyperparameters (i.e. the velocity coefficients). Both PSO and APSO have been employed extensively across the different avenues of research presented in this thesis, whenever experimental curve fitting was involved. In this appendix, the specific application of a VCSEL model parameter extraction has been presented, in order to showcase the strengths and flexibility of this tool, which is capable of reliably matching the reference experimental data that it is provided with.

References

[1] Near Margalit, Chao Xiang, Steven M. Bowers, Alexis Bjorlin, Robert Blum, and John E. Bowers. Perspective on the future of silicon photonics and electronics. *Appl. Phys. Lett.*, 118(22):220501, 06 2021. ISSN 0003-6951. doi: 10.1063/5.0050117.

[2] Industry Research. Photonic integrated circuit (PIC) market overview. https://tinyurl.com/bdcwhx4b, 2026. [Online; accessed 15-January-2026].

[3] Claudia Hoessbacher, Benedikt Baeuerle, Eva De Leo, Wolfgang Heni, Stephan Koch, and Juerg Leuthold. System-on-chip photonic integrated circuits in silicon photonics and the role of plasmonics. In *2023 Optical Fiber Communication Conference (OFC)*, page Tu2E.5. Optica Publishing Group, 2023. doi: 10.1364/OFC.2023.Tu2E.5.

[4] Isaac Rodríguez-Ruiz, Tobias N. Ackermann, Xavier Muñoz-Berbel, and Andreu Llobera. Photonic Lab-on-a-Chip: Integration of optical spectroscopy in microfluidic systems. *Analytical Chemistry*, 88(13):6630–6637, 2016. doi: 10.1021/acs.analchem.6b00377.

[5] Alberto Politi, Jonathan C.F. Matthews, Mark G. Thompson, and Jeremy L. O'Brien. Integrated quantum photonics. *IEEE J. Sel. Top. Quantum Electron.*, 15(6):1673–1684, 2009. doi: 10.1109/JSTQE.2009.2026060.

[6] IDTechEx. Silicon photonics and photonic integrated circuits 2025-2035: Technologies, market, forecasts. https://tinyurl.com/2snnwcjd, 2026. [Online; accessed 16-January-2026].

[7] OpenAI. Introducing ChatGPT, 2022. URL https://openai.com/index/chatgpt/.

[8] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133, 1943. doi: 10.1007/BF02478259.

[9] Alan M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950. doi: 10.1093/mind/LIX.236.433.

[10] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408, 1958. doi: 10.1037/h0042519.

[11] Shun'ichi Amari. A theory of adaptive pattern classifiers. *IEEE Trans. Comput.*, EC-16(3):299–307, 1967. doi: 10.1109/PGEC.1967.264666.

[12] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0.

[13] Yann Le Cun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In *1989 International Conference on Neural Information Processing Systems (NIPS)*, page 396–404. MIT Press, 1989.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *2012 Advances in Neural Information Processing Systems (NIPS)*, volume 25. Curran Associates, Inc., 2012. doi: 10.1145/3065386.

[15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[16] Vladimír Kunc and Jiří Kléma. Three decades of activations: A comprehensive survey of 400 activation functions for neural networks, 2024. URL https://arxiv.org/abs/2402.09092.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[18] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, 2015. URL https://arxiv.org/abs/1502.01852.

[20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

[21] Thomas Ferreira de Lima, Hsuan-Tung Peng, Alexander N. Tait, Mitchell A. Nahmias, Heidi B. Miller, Bhavin J. Shastri, and Paul R. Prucnal. Machine learning with neuromorphic photonics. *J. Lightwave Technol.*, 37(5):1515–1534, 2019. doi: 10.1109/JLT.2019.2903474.

[22] M. Mitchell Waldrop. More than Moore. *Nature News*, 530:144, 02 2016. doi: 10.1038/530144a.

[23] Tingzhao Fu, Jianfa Zhang, Run Sun, Yuyao Huang, Wei Xu, Sigang Yang, Zhihong Zhu, and Hongwei Chen. Optical neural networks: progress and challenges. *Light Sci. Appl.*, 13(1):263, 2024. doi: 10.1038/s41377-024-01590-3.

[24] Johannes Feldmann, Nathan Youngblood, David Wright, Harish Bhaskaran, and Wolfram H. P. Pernice. All-optical spiking neurosynaptic networks with self-learning capabilities. *Nature*, 569(7755):208–214, 2019. doi: 10.1038/s41586-019-1157-8.

[25] Hsuan-Tung Peng, Mitchell A. Nahmias, Thomas Ferreira de Lima, Alexander N. Tait, and Bhavin J. Shastri. Neuromorphic photonic integrated circuits. *IEEE J. Sel. Top. Quantum Electron.*, 24(6):1–15, 2018. doi: 10.1109/JSTQE.2018.2840448.

[26] Epoch AI. Data on ai models, 07 2025. URL https://epoch.ai/data/ai-models. Accessed: 2025-11-10.

[27] Dave Steinkraus, Ian Buck, and Patrice Y. Simard. Using GPUs for machine learning algorithms. In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pages 1115–1120 Vol. 2, 2005. doi: 10.1109/ICDAR.2005.251.

[28] Kaiyuan Guo, Lingzhi Sui, Jiantao Qiu, Jincheng Yu, Junbin Wang, Song Yao, Song Han, Yu Wang, and Huazhong Yang. Angel-Eye: A complete design flow for mapping CNN onto embedded FPGA. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 37(1):35–47, 2018. doi: 10.1109/TCAD.2017.2705069.

[29] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017. doi: 10.1145/3079856.3080246.

[30] Lorenzo De Marinis, Marco Cococcioni, Piero Castoldi, and Nicola Andriolli. Photonic neural networks: A survey. *IEEE Access*, 7:175827–175841, 2019. doi: 10.1109/ACCESS.2019.2957245.

[31] Lorenzo De Marinis, Marco Cococcioni, Odile Liboiron-Ladouceur, Giampiero Contestabile, Piero Castoldi, and Nicola Andriolli. Photonic integrated reconfigurable linear processors as neural network accelerators. *Appl. Sci.*, 11(13), 2021. doi: 10.3390/app11136232.

[32] Min Teng, Amirmahdi Honardoost, Yousef Alahmadi, Sajad Saghaye Polkoo, Keisuke Kojima, He Wen, C. Kyle Renshaw, Patrick LiKamWa, Guifang Li, Sasan Fathpour, Reza Safian, and Leimeng Zhuang. Miniaturized silicon photonics devices for integrated optical signal processors. *J. Lightwave Technol.*, 38(1):6–17, 2020. doi: 10.1109/JLT.2019.2943251.

[33] George Dabos, Dimitris V. Bellas, Ripalta Stabile, Miltiadis Moralis-Pegios, George Giamougiannis, Apostolos Tsakyridis, Angelina Totovic, Elefterios Lidorikis, and Nikos Pleros. Neuromorphic photonic technologies and architectures: scaling opportunities and performance frontiers. *Opt. Mater. Express*, 12(6):2343–2367, 2022. doi: 10.1364/OME.452138.

[34] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Netw.*, 115:100–123, 2019. doi: 10.1016/j.neunet.2019.03.005.

[35] Min Yan, Can Huang, Peter Bienstman, Peter Tino, Wei Lin, and Jie Sun. Emerging opportunities and challenges for the future of reservoir computing. *Nat. Comm.*, 15(1):2056, 2024.

[36] Mitsumasa Nakajima, Kenji Tanaka, and Toshikazu Hashimoto. Scalable reservoir computing on coherent linear photonic processor. *Commun. Phys.*, 4(1):20, 2021. doi: 10.1038/s42005-021-00519-1.

[37] Dongliang Wang, Yikun Nie, Gaolei Hu, Hon Ki Tsang, and Chaoran Huang. Ultrafast silicon photonic reservoir computing engine delivering over 200 TOPS. *Nat. Commun.*, 15(1):10841, 2024. doi: 10.1038/s41467-024-55172-3.

[38] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191.

[39] Gandhi Manjunath and Herbert Jaeger. Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural Comput.*, 25(3):671–696, 03 2013. doi: 10.1162/NECO_a_00411.

[40] Izzet B. Yildiz, Herbert Jaeger, and Stefan J. Kiebel. Re-visiting the echo state property. *Neural Netw.*, 35:1–9, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.07.005.

[41] Kento Saito, Kazutaka Kanno, and Atsushi Uchida. Controlling nonlinearity and memory by feedback delay time in delay-based reservoir computing. *Nonlinear theory appl. IEICE*, 15(4):764–783, 2024. doi: 10.1587/nolta.15.764.

[42] Guy Van der Sande, Daniel Brunner, and Miguel C. Soriano. Advances in photonic reservoir computing. *Nanophotonics*, 6(3):561–576, 2017. doi: doi:10.1515/nanoph-2016-0132.

[43] Kristof Vandoorne, Pauline Mechet, Thomas Van Vaerenbergh, Martin Fiers, Geert Morthier, David Verstraeten, Benjamin Schrauwen, Joni Dambre, and Peter Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.*, 5(1):3541, 2014. doi: 10.1038/ncomms4541.

[44] Daniel Brunner and Ingo Fischer. Reconfigurable semiconductor laser networks based on diffractive coupling. *Opt. Lett.*, 40(16):3854–3857, Aug 2015. doi: 10.1364/OL.40.003854.

[45] Quentin Vinckier, François Duport, Anteo Smerieri, Kristof Vandoorne, Peter Bienstman, Marc Haelterman, and Serge Massar. High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica*, 2(5):438–446, 05 2015. doi: 10.1364/OPTICA.2.000438.

[46] Irene Estébanez, Janek Schwind, Ingo Fischer, and Apostolos Argyris. Accelerating photonic computing by bandwidth enhancement of a time-delay reservoir. *Nanophotonics*, 9(13):4163–4171, 2020. doi: 10.1515/nanoph-2020-0184.

[47] Wikimedia Commons. File:neuron. https://commons.wikimedia.org/wiki/File:Neuron.svg, 2025. [Online; accessed 20-November-2025].

[48] Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain Sci.*, 12(7), 2022. doi: 10.3390/brainsci12070863.

[49] Alexander N. Tait, Mitchell A. Nahmias, Bhavin J. Shastri, and Paul R. Prucnal. Broadcast and weight: An integrated network for scalable photonic spike processing. *J. Lightwave Technol.*, 32(21):4029–4041, 2014. doi: 10.1109/JLT.2014.2345652.

[50] Mitchell A. Nahmias, Bhavin J. Shastri, Alexander N. Tait, and Paul R. Prucnal. A leaky integrate-and-fire laser neuron for ultrafast cognitive computing. *IEEE J. Sel. Top. Quantum Electron.*, 19(5):1–12, 2013. doi: 10.1109/JSTQE.2013.2257700.

[51] Shuiying Xiang, Zhenxing Ren, Ziwei Song, Yahui Zhang, Xingxing Guo, Genquan Han, and Yue Hao. Computing primitive of fully VCSEL-based all-optical spiking neural network for supervised learning and pattern classification. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(6):2494–2505, 2021. doi: 10.1109/TNNLS.2020.3006263.

[52] Giovanni Donati, Stefano Biasi, Lorenzo Pavesi, and Antonio Hurtado. All-optical spiking processing and reservoir computing with a passive silicon microring and wavelength-time division multiplexing. *Photon. Res.*, 13(9):2641–2653, 09 2025. doi: 10.1364/PRJ.558405.

[53] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.*, 73: 58–61, 07 1994. doi: 10.1103/PhysRevLett.73.58.

[54] William R. Clements, Peter C. Humphreys, Benjamin J. Metcalf, W. Steven Kolthammer, and Ian A. Walmsley. Optimal design for universal multiport interferometers. *Optica*, 3(12):1460–1465, 12 2016. doi: 10.1364/OPTICA.3.001460.

[55] Farhad Shokraneh, Simon Geoffroy-gagnon, and Odile Liboiron-Ladouceur. The diamond mesh, a phase-error- and loss-tolerant field-programmable MZI-based optical processor for optical neural networks. *Opt. Express*, 28(16): 23495–23508, 08 2020. doi: 10.1364/OE.395441.

[56] Nicholas C. Harris, Jacques Carolan, Darius Bunandar, Mihika Prabhu, Michael Hochberg, Tom Baehr-Jones, Michael L. Fanto, A. Matthew Smith, Christopher C. Tison, Paul M. Alsing, and Dirk Englund. Linear programmable nanophotonic processors. *Optica*, 5(12):1623–1631, 12 2018. doi: 10.1364/OPTICA.5.001623.

[57] Kaveh (Hassan) Rahbardar Mojaver, Bokun Zhao, Edward Leung, S. Mohammad Reza Safaee, and Odile Liboiron-Ladouceur. Addressing the programming challenges of practical interferometric mesh based optical processors. *Opt. Express*, 31(15):23851–23866, 07 2023. doi: 10.1364/OE.489493.

[58] Hui Zhang, Mile Gu, Xu Dong Jiang, Jayne Thompson, Hong Cai, Stefano Paesani, Raffaele Santagati, Anthony Laing, Yi Zhang, Man Hong Yung, Yu Zhi Shi, Faeyz Karim Muhammad, Guo Qiang Lo, Xian Shu Luo, Bin Dong, Dim Li Kwong, Leong Chuan Kwek, and Ai Qun Liu. An optical neural chip for implementing complex-valued neural network. *Nat. Commun.*, 12(1):457, 2021. doi: 10.1038/s41467-020-20719-7.

[59] Alexander N. Tait, Thomas Ferreira de Lima, Mitchell A. Nahmias, Heidi B. Miller, Hsuan-Tung Peng, Bhavin J. Shastri, and Paul R. Prucnal. Silicon photonic modulator neuron. *Phys. Rev. Appl.*, 11(6):064043, 2019.

[60] Ian A. D. Williamson, Tyler W. Hughes, Momchil Minkov, Ben Bartlett, Sunil Pai, and Shanhui Fan. Reprogrammable electro-optic nonlinear activation functions for optical neural networks. *IEEE J. Sel. Topics Quantum Electron.*, 26(1):1–12, 2020. doi: 10.1109/JSTQE.2019.2930455.

[61] Aashu Jha, Chaoran Huang, and Paul R. Prucnal. Reconfigurable all-optical nonlinear activation functions for neuromorphic photonics. *Opt. Lett.*, 45(17): 4819–4822, 09 2020. doi: 10.1364/OL.398234.

[62] Bo Wu, Hengkang Li, Weiyu Tong, Jianji Dong, and Xinliang Zhang. Low-threshold all-optical nonlinear activation function based on a Ge/Si hybrid structure in a microring resonator. *Opt. Mater. Express*, 12(3):970–980, 2022. doi: 10.1364/OME.447330.

[63] Kun Liao, Chentong Li, Tianxiang Dai, Chuyu Zhong, Hongtao Lin, Xiaoyong Hu, and Qihuang Gong. Matrix eigenvalue solver based on reconfigurable photonic neural network. *Nanophotonics*, 11(17):4089–4099, 2022.

[64] Yang Shi, Junyu Ren, Guanyu Chen, Wei Liu, Chuqi Jin, Xiangyu Guo, Yu Yu, and Xinliang Zhang. Nonlinear germanium-silicon photodiode for activation and monitoring in photonic neuromorphic networks. *Nat. Commun.*, 13(1): 6048, 2022.

[65] Chuyu Zhong, Kun Liao, Tianxiang Dai, Maoliang Wei, Hui Ma, Jianghong Wu, Zhibin Zhang, Yuting Ye, Ye Luo, Zequn Chen, et al. Graphene/silicon heterojunction for reconfigurable phase-relevant activation function in coherent optical neural networks. *Nat. Commun.*, 14(1):6939, 2023.

[66] Chenduan Chen, Zhan Yang, Tao Wang, Yalun Wang, Kai Gao, Jiajia Wu, Jun Wang, Jianrong Qiu, and Dezhi Tan. Ultra-broadband all-optical nonlinear activation function enabled by MoTe2/optical waveguide integrated devices. *Nat. Commun.*, 15(1):9047, 2024.

[67] Hengkang Li, Bo Wu, Weiyu Tong, Jianji Dong, and Xinliang Zhang. All-optical nonlinear activation function based on germanium silicon hybrid asymmetric coupler. *IEEE J. Sel. Top. Quantum Electron.*, 29(2: Optical Computing):1–6, 2022.

[68] Thorsten S. Rasmussen, Yi Yu, and Jesper Mork. All-optical non-linear activation function for neuromorphic photonic computing using semiconductor Fano lasers. *Opt. Lett.*, 45(14):3844–3847, 2020.

[69] Weizhen Yu, Shuang Zheng, Zhenyu Zhao, Bin Wang, and Weifeng Zhang. Reconfigurable low-threshold all-optical nonlinear activation functions based on an add-drop silicon microring resonator. *IEEE Photonics J.*, 14(6):1–7, 2022.

[70] Gordon H. Y. Li, Ryoto Sekine, Rajveer Nehra, Robert M. Gray, Luis Ledezma, Qiushi Guo, and Alireza Marandi. All-optical ultrafast ReLU function for energy-efficient nanophotonic deep learning. *Nanophotonics*, 12 (5):847–855, 2023. doi: doi:10.1515/nanoph-2022-0137.

[71] Antonio Lechiara, Andrea Marchisio, Bin Shi, Yi Wang, Andrea Carena, Yuqing Jiao, and Ripalta Stabile. Ultra-compact leaky ReLU nonlinear function on IMOS. In *2025 European Conference on Optical Communications (ECOC)*, pages 1–4, 2025. doi: 10.1109/ECOC66593.2025.11263433.

[72] Albert Ryou, James Whitehead, Maksym Zhelyeznyakov, Paul Anderson, Cem Keskin, Michal Bajcsy, and Arka Majumdar. Free-space optical neural network based on thermal atomic nonlinearity. *Photon. Res.*, 9(4):B128–B134, 04 2021. doi: 10.1364/PRJ.415964.

[73] George Mourgias-Alexandris, Apostolos Tsakyridis, Nikolaos Passalis, Anastasios Tefas, Konstantinos Vyrsokinos, and Nikos Pleros. An all-optical neuron with sigmoid activation function. *Opt. Express*, 27(7):9620–9630, 04 2019. doi: 10.1364/OE.27.009620.

[74] Daniel Chelladurai, Manuel Kohli, Joel Winiger, David Moor, Andreas Messner, Yuriy Fedoryshyn, Mohammed Eleraky, Yuqi Liu, Hua Wang, and Juerg Leuthold. Barium titanate and lithium niobate permittivity and Pockels coefficients from megahertz to sub-terahertz frequencies. *Nat. Mater.*, 24(6): 868–875, 2025. doi: 10.1038/s41563-025-02158-1.

[75] Daniel Perez, Ivana Gasulla, Lee Crudgington, David Thomson, Ali Khokhar, Ke Li, Wei Cao, Goran Mashanovich, and Jose Capmany. Multipurpose silicon photonics signal processor core. *Nat. Comm.*, 8, 09 2017. doi: 10.1038/s41467-017-00714-1.

[76] Lorenzo Tunesi, Hasan Awad, Andrea Carena, Vittorio Curri, and Paolo Bardella. Toggleable transparency states in thermally-shifted multi-MRR cascaded filters. In Graham T. Reed and Jonathan Bradley, editors, *Silicon Photonics XX*, volume 13371, page 133710T. International Society for Optics and Photonics, SPIE, 2025. doi: 10.1117/12.3044329.

[77] Ali Cem, Siqi Yan, Yunhong Ding, Darko Zibar, and Francesco Da Ros. Data-driven modeling of Mach-Zehnder Interferometer-based optical matrix multipliers. *J. Lightwave Technol.*, 41(16):5425–5436, 2023. doi: 10.1109/JLT.2023.3263235.

[78] Ali Cem, Siqi Yan, Uiara Celine de Moura, Yunhong Ding, Darko Zibar, and Francesco Da Ros. Comparison of models for training optical matrix multipliers in neuromorphic PICs. In *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, pages 1–3, 2022. doi: 10.1364/OFC.2022.M2G.5.

[79] Sudip Shekhar, Wim Bogaerts, Lukas Chrostowski, John E Bowers, Michael Hochberg, Richard Soref, and Bhavin J Shastri. Roadmapping the next generation of silicon photonics. *Nat. Comm.*, 15(1):751, 2024. doi: 10.1038/s41467-024-44750-0.

[80] Andrea Marchisio, Ali Cem, Yunhong Ding, Vittorio Curri, Andrea Carena, Francesco Da Ros, and Paolo Bardella. Optimization of 3x3 neuromorphic photonic network for programmable Boolean operations. In *SPIE Photonic West: Physics and Simulation of Optoelectronic Devices XXXII*, volume 12880, pages 191104–8, 2024. doi: 10.1117/12.3002953.

[81] Andrea Marchisio, Francesco Da Ros, Vittorio Curri, Andrea Carena, and Paolo Bardella. Comprehensive model of MZI-based circuits for photonic computing applications. *Communications Physics*, 8(1):277, 2025. doi: 10.1038/s42005-025-02176-0.

[82] Andrea Marchisio, Lorenzo Tunesi, Hasan Awad, Enrico Ghillino, Vittorio Curri, Andrea Carena, and Paolo Bardella. Comprehensive thermal crosstalk model of meshed MZI topologies for neuromorphic computing. In *AI and Optical Data Sciences VI*, volume 13375, pages 90–94. SPIE, 2025. doi: 10.1117/12.3043230.

[83] Teng Ma, Andrea Marchisio, Francesco Da Ros, Vittorio Curri, Andrea Carena, and Paolo Bardella. Wideband operation of 3x3 Mach-Zehnder interferometer mesh for programmable photonic computing. In *2025 25th Anniversary International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, 2025. doi: 10.1109/ICTON67126.2025.11125174.

[84] Yunhong Ding, Valerija Kamchevska, Kjeld Dalgaard, Feihong Ye, Rameez Asif, Simon Gross, Michael J Withford, Michael Galili, Toshio Morioka, and Leif Katsuo Oxenløwe. Reconfigurable SDM switching using novel silicon photonic integrated circuit. *Sci. Rep.*, 6(1):39058, 2016. doi: 10.1038/srep39058.

[85] Ali Cem. *Modeling Photonic Integrated Circuits for Optical Computing using Machine Learning*. PhD thesis, Technical University of Denmark, 2023.

[86] Wei-Ping Huang. Coupled-mode theory for optical waveguides: an overview. *J. Opt. Soc. Am. A*, 11(3):963–983, 03 1994. doi: 10.1364/JOSAA.11.000963.

[87] Dan-Xia Xu, André Delâge, Pierre Verly, Siegfried Janz, Shurui Wang, Martin Vachon, PengHui Ma, Jean Lapointe, Daniele Melati, Pavel Cheben, and Jens H. Schmid. Empirical model for the temperature dependence of silicon refractive index from O to C band based on waveguide measurements. *Opt. Express*, 27(19):27229–27241, 09 2019. doi: 10.1364/OE.27.027229.

[88] Amin Shafiee, Sanmitra Banerjee, Krishnendu Chakrabarty, Sudeep Pasricha, and Mahdi Nikdast. Analysis of optical loss and crosstalk noise in MZI-based coherent photonic neural networks. *J. Lightwave Technol.*, pages 1–16, 2024. doi: 10.1109/JLT.2024.3373250.

[89] Ali Najjar Amiri, Aycan Deniz Vit, Kazim Gorgulu, and Emir Salih Magden. Deep photonic network platform enabling arbitrary and broadband optical functionality. *Nat. Comm.*, 15(1):1432, 2024. doi: 10.1038/s41467-024-45846-3.

[90] Meisam Bahadori, Alexander Gazman, Natalie Janosik, Sébastien Rumley, Ziyi Zhu, Robert Polster, Qixiang Cheng, and Keren Bergman. Thermal rectification of integrated microheaters for microring resonators in silicon photonics platform. *J. Lightwave Technol.*, 36(3):773–788, 2018. doi: 10.1109/JLT.2017.2781131.

[91] RSoft™. RSoft™ Photonic Device Tools . https://www.synopsys.com/photonic-solutions/rsoft-photonic-device-tools/rsoft-products.html, 2024. [Online; accessed 2-December-2024].

[92] Peter L. McMahon. The physics of optical computing. *Nat. Rev. Phys.*, 5(12): 717–734, 2023. doi: 10.1038/s42254-023-00645-5.

[93] Yanbing Zhang, Chad Husko, Simon Lefrancois, Isabella H. Rey, Thomas F. Krauss, Jochen Schr"oder, and Benjamin J. Eggleton. Non-degenerate two-photon absorption in silicon waveguides: analytical and experimental study. *Opt. Express*, 23(13):17101–17110, Jun 2015. doi: 10.1364/OE.23.017101.

[94] Bradley J. Frey, Douglas B. Leviton, and Timothy J. Madison. Temperature-dependent refractive index of silicon and germanium. In *Optomechanical Technologies for Astronomy*, volume 6273, page 62732J, 2006. doi: 10.1117/12.672850.

[95] Jens H. Schmid, Marc Ibrahim, Pavel Cheben, Jean Lapointe, Siegfried Janz, Przemek J. Bock, Adam Densmore, Boris Lamontagne, Rufus Ma, Winnie N. Ye, and Dan Xia Xu. Temperature-independent silicon subwavelength grating waveguides. *Opt. Lett.*, 36(11):2110–2112, 2011. doi: 10.1364/OL.36.002110.

[96] David W. Hahn and M. Necati Ozişik. Heat conduction fundamentals. In *Heat Conduction*, chapter 1, pages 1–39. John Wiley & Sons, New York, 2012. doi: 10.1002/9781118411285.ch1.

[97] Pedro Andrei Krochin Yepez, Ulrike Scholz, Jan Niklas Caspers, and Andre Zimmermann. Novel measures for thermal management of silicon photonic optical phased arrays. *IEEE Photonics J.*, 11(4):1–15, 2019. doi: 10.1109/JPHOT.2019.2925138.

[98] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995. doi: 10.1109/ICNN.1995.488968.

[99] Andrea Marchisio, Enrico Ghillino, Vittorio Curri, Andrea Carena, and Paolo Bardella. Particle swarm optimization-assisted approach for the extraction of VCSEL model parameters. *Opt. Lett.*, 49(1):125–128, 1 2024. doi: 10.1364/OL.506958.

[100] Zhiwei Xue, Tiankuang Zhou, Zhihao Xu, Shaoliang Yu, Qionghai Dai, and Lu Fang. Fully forward mode training for optical neural networks. *Nature*, 632(8024):280–286, 2024. doi: 10.1038/s41586-024-07687-4.

[101] Weipeng Zhang, Tengji Xu, Jiawei Zhang, Bhavin J. Shastri, Chaoran Huang, and Paul Prucnal. Online training and pruning of photonic neural networks. In *2023 IEEE Photonics Conference (IPC)*, pages 1–2, 2023. doi: 10.1109/IPC57732.2023.10360757.

[102] Hui Zhang, Jayne Thompson, Mile Gu, Xu Dong Jiang, Hong Cai, Patricia Yang Liu, Yuzhi Shi, Yi Zhang, Muhammad Faeyz Karim, Guo Qiang Lo, Xianshu Luo, Bin Dong, Leong Chuan Kwek, and Ai Qun Liu. Efficient on-chip training of optical neural networks using genetic algorithm. *ACS Photonics*, 8(6):1662–1672, 2021. doi: 10.1021/acsphotonics.1c00035.

[103] Yuepeng Wu, Hongxiang Guo, Bowen Zhang, Jifang Qiu, Zhisheng Yang, and Jian Wu. Integrated photonic modular arithmetic processor. *Photon. Res.*, 12(11):2676–2690, Nov 2024. doi: 10.1364/PRJ.527762.

[104] Ihtesham Khan, Andrea Marchisio, Lorenzo Tunesi, Muhammad Umar Masood, Enrico Ghillino, Vittorio Curri, Andrea Carena, and Paolo Bardella. A machine learning-based model for characterizing stationary-and-dynamic behavior of VCSEL. In *CLEO: Fundamental Science*, pages JW2A–141, 2023. doi: 10.1364/CLEO_AT.2023.JW2A.141.

[105] Andrea Marchisio, Ihtesham Khan, Lorenzo Tunesi, Muhammad Umar Masood, Enrico Ghillino, Vittorio Curri, Andrea Carena, and Paolo Bardella. Automated model for characterization of vcsel circuit-level parameters using machine learning. In *2023 European Conference on Integrated Optics (ECIO)*, pages 264–266, 2023.

[106] Andrea Marchisio, Enrico Ghillino, Vittorio Curri, Andrea Carena, and Paolo Bardella. Particle swarm optimization hyperparameters tuning for physical-model fitting of VCSEL measurements. In *Vertical-Cavity Surface-Emitting Lasers XXVIII*, volume 12904, page 129040N. International Society for Optics and Photonics, SPIE, 2024. doi: 10.1117/12.3002576.

[107] Jorge Bravo-Abad, Alejandro Rodriguez, Peter Bermel, Steven G. Johnson, John D. Joannopoulos, and Marin Soljačić. Enhanced nonlinear optics in photonic-crystal microcavities. *Opt. Express*, 15(24):16161–16176, 11 2007. doi: 10.1364/OE.15.016161.

[108] Yuqing Jiao, Jos van der Tol, Vadim Pogoretskii, Jorn van Engelen, Amir Abbas Kashi, Sander Reniers, Yi Wang, Xinran Zhao, Weiming Yao, Tianran Liu, Francesco Pagliano, Andrea Fiore, Xuebing Zhang, Zizheng Cao, Rakesh Ranjan Kumar, Hon Ki Tsang, Rene van Veldhoven, Tjibbe de Vries, Erik-Jan Geluk, Jeroen Bolk, Huub Ambrosius, Meint Smit, and Kevin Williams. Indium phosphide membrane nanophotonic integrated circuits on silicon. *Phys. Status Solidi A*, 217(3):1900606–1900618, 2020. doi: 10.1002/pssa.201900606.

[109] Jos van der Tol, Yuqing Jiao, Jorn P. Van Engelen, Vadim Pogoretskiy, Amir Abbas Kashi, and Kevin Williams. InP Membrane on Silicon (IMOS) Photonics. *IEEE J. Quantum Electron.*, 56(1):1–7, 2020. doi: 10.1109/JQE.2019.2953296.

[110] Antonio Lechiara, Andrea Marchisio, Bin Shi, Yi Wang, Andrea Carena, Yuqing Jiao, and Ripalta Stabile. Reprogrammable all-optical ultra-compact nonlinear activation function for neuromorphic computing on indium phosphide membrane on silicon. In *2025 IEEE Photonics Benelux Annual Symposium*, page 1, 2025.

[111] Luc M. Augustin, Rui Santos, Erik den Haan, Steven Kleijn, Peter J. A. Thijs, Sylwester Latkowski, Dan Zhao, Weiming Yao, Jeroen Bolk, Huub Ambrosius,

Sergei Mingaleev, André Richter, Arjen Bakker, and Twan Korthorst. InP-based generic foundry platform for photonic integrated circuits. *IEEE J. Sel. Top. Quantum Electron.*, 24(1):1–10, 2018. doi: 10.1109/JSTQE.2017.2720967.

[112] S. Ehsanfar, Faramarz Kanjouri, Hasan Tashakori, and Amirhosein Esmailian. First-principles study of structural, electronic, mechanical, thermal, and phonon properties of III-phosphides (BP, AlP, GaP, and InP). *J. Electron. Mater.*, 46, 06 2017. doi: 10.1007/s11664-017-5640-4.

[113] Kevin Williams, Erwin Bente, Dominik Heiss, Yuqing Jiao, Katarzyna Ławniczuk, Xaveer Leijtens, Jos van der Tol, and Meint Smit. InP photonic circuits using generic integration [invited]. *Photon. Res.*, 3(5):B60–B68, 10 2015. doi: 10.1364/PRJ.3.000B60.

[114] Yi Wang, Victor Dolores-Calzadilla, Kevin A. Williams, Meint K. Smit, and Yuqing Jiao. Ultra-compact and efficient microheaters on a submicron-thick InP membrane. *Journal of Lightwave Technology*, 41(6):1790–1800, 2023. doi: 10.1109/JLT.2022.3225110.

[115] Bin Shi. *Semiconductor Optical Amplifier-based Photonic Integrated Deep Neural Networks*. Phd thesis, Electrical Engineering, 06 2022. Proefschrift.

[116] Shaowu Chen, Libin Zhang, Yonghao Fei, and Tongtong Cao. Bistability and self-pulsation phenomena in silicon microring resonators based on nonlinear optical effects. *Opt. Express*, 20(7):7454–7468, 03 2012. doi: 10.1364/OE.20.007454.

[117] Francesco G. Della Corte, Giuseppe Cocorullo, Mario Iodice, and Ivo Rendina. Temperature dependence of the thermo-optic coefficient of InP, GaAs, and SiC from room temperature to 600 K at the wavelength of 1.5 um. *Appl. Phys. Lett.*, 77(11):1614–1616, 09 2000. doi: 10.1063/1.1308529.

[118] J. Komma, Christophe Schwarz, Gerd Hofmann, Daniel Heinert, and Ronny Nawrodt. Thermo-optic coefficient of silicon at 1550 nm and cryogenic temperatures. *Appl. Phys. Lett.*, 101(4):041905, 07 2012. doi: 10.1063/1.4738989.

[119] Mansoor Sheik-Bahae and Michael P. Hasselbeck. Third-order optical nonlinearities. In Michael Bass, editor, *Handbook of Optics: Volume IV - Optical Properties of Materials, Nonlinear Optics, Quantum Optics*. McGraw-Hill Education, New York, 3rd edition, 2010.

[120] Salvatore Salpietro, Marco Novarese, Cristina Rimoldi, and Mariangela Gioannini. Study of microring nonlinearities in silicon photonics for neuromorphic computing. In *Silicon Photonics XX*, volume 13371, page 133710L. SPIE, 2025. doi: 10.1117/12.3040062.

[121] Dieter K. Schroder, R. Noel Thomas, and J.C. Swartz. Free carrier absorption in silicon. *IEEE J. Solid-State Circuits*, 13(1):180–187, 1978. doi: 10.1109/JSSC.1978.1051012.

[122] Younghyun Kim, Mitsuru Takenaka, Takenori Osada, Masahiko Hata, and Shinichi Takagi. Strain-induced enhancement of plasma dispersion effect and free-carrier absorption in SiGe optical modulators. *Sci. Rep.*, 4(1):4683, 2014. doi: 10.1038/srep04683.

[123] Xuezhe Zheng, Ying Luo, Guoliang Li, Ivan Shubin, Hiren Thacker, Jin Yao, Kannan Raj, John E. Cunningham, and Ashok V. Krishnamoorthy. Enhanced optical bistability from self-heating due to free carrier absorption in substrate removed silicon ring modulators. *Opt. Express*, 20(10):11478–11486, 05 2012. doi: 10.1364/OE.20.011478.

[124] Marco Novarese, Sebastian Romero Garcia, Stefania Cucco, Don Adams, Jock Bovington, and Mariangela Gioannini. Study of nonlinear effects and self-heating in a silicon microring resonator including a Shockley-Read-Hall model for carrier recombination. *Opt. Express*, 30(9):14341–14357, 04 2022. doi: 10.1364/OE.446739.

[125] Li He, Yang Liu, Christophe Galland, Andy Eu-Jin Lim, Guo-Qiang Lo, Tom Baehr-Jones, and Michael Hochberg. A high-efficiency nonuniform grating coupler realized with 248-nm optical lithography. *IEEE Photonics Technol. Lett.*, 25(14):1358–1361, 2013. doi: 10.1109/LPT.2013.2265911.

[126] Chao Li, Jeong Hwan Song, Jing Zhang, Huijuan Zhang, Shiyi Chen, Mingbin Yu, and G. Q. Lo. Silicon polarization independent microring resonator-based optical tunable filter circuit with fiber assembly. *Opt. Express*, 19(16):15429–15437, 08 2011. doi: 10.1364/OE.19.015429.

[127] Hasitha Jayatilleka, Kyle Murray, Miguel Ángel Guillén-Torres, Michael Caverley, Ricky Hu, Nicolas A. F. Jaeger, Lukas Chrostowski, and Sudip Shekhar. Wavelength tuning and stabilization of microring-based filters using silicon in-resonator photoconductive heaters. *Opt. Express*, 23(19):25084–25097, 09 2015. doi: 10.1364/OE.23.025084.

[128] Shoma Nakamura, Kota Sekiya, Shinichiro Matano, Yui Shimura, Yuuki Nakade, Kenta Nakagawa, Yasuaki Monnai, and Hideyuki Maki. High-speed and on-chip optical switch based on a graphene microheater. *ACS Nano*, 16 (2):2690–2698, 2022. doi: 10.1021/acsnano.1c09526.

[129] Jinha Lim, Joonsup Shim, Inki Kim, S. K. Kim, Dae Myeong Geum, and Sang Hyeon Kim. Thermally tunable microring resonators based on germanium-on-insulator for mid-infrared spectrometer. *APL Photonics*, 9(10): 106109, 10 2024. doi: 10.1063/5.0220382.

[130] Siqi Yan, Xiaolong Zhu, Lars Hagedorn Frandsen, Sanshui Xiao, N. Asger Mortensen, Jianji Dong, and Yunhong Ding. Slow-light-enhanced energy

efficiency for graphene microheaters on silicon photonic crystal waveguides. *Nat. Commun.*, 8, 2017. doi: 10.1038/ncomms14411.

[131] Mattia Rossetti, Paolo Bardella, and Ivo Montrosset. Time-domain travelling-wave model for quantum dot passively mode-locked lasers. *IEEE J. Quantum Electron.*, 47(2):139–150, 2011. doi: 10.1109/JQE.2010.2055550.

[132] Mikhail N. Polyanskiy. Refractiveindex.info database of optical constants. *Sci. Data*, 11(1):94, 2024. doi: 10.1038/s41597-023-02898-2.

[133] Brian R. Bennett, Richard A. Soref, and Jesús A. Del Alamo. Carrier-induced change in refractive index of InP, GaAs and InGaAsP. *IEEE J. Quantum Electron.*, 26(1):113–122, 1990. doi: 10.1109/3.44924.

[134] Srinivasa Reddy Tamalampudi, Ghada Dushaq, Solomon M. Serunjogi, Nitul S. Rajput, and Mahmoud S. Rasras. Simultaneous optical power insensitivity and non-volatile wavelength trimming using 2D $In_{4/3}P_2Se_6$ integration in silicon photonics. *npj 2D Mater. Appl.*, 8(1):46, 2024. doi: 10.1038/s41699-024-00481-w.

[135] Toshitsugu Uesugi, Bong-Shik Song, Takashi Asano, and Susumu Noda. Investigation of optical nonlinearities in an ultra-high-Q Si nanocavity in a two-dimensional photonic crystal slab. *Opt. Express*, 14(1):377–386, 01 2006. doi: 10.1364/OPEX.14.000377.

[136] Alfredo de Rossi, Michele Lauritano, Sylvain Combrié, Quynh Vy Tran, and Chad Husko. Interplay of plasma-induced and fast thermal nonlinearities in a gaas-based photonic crystal nanocavity. *Phys. Rev. A*, 79:043818, 04 2009. doi: 10.1103/PhysRevA.79.043818.

[137] David R. Lide. *CRC handbook of chemistry and physics / ed. by David R. Lide*. CRC Press, Boca Raton, 74th ed. edition, 1993.

[138] Ioffe Institute. Indium phosphide thermal properties. https://www.ioffe.ru/SVA/NSM/Semicond/InP/thermal.html, 2025. [Online; accessed 12-November-2025].

[139] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing*, volume 30, page 3, 2013.

[140] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000.

[141] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), 2023. URL https://arxiv.org/abs/1606.08415.

[142] Philip Naveen. Phish: a novel hyper-optimizable activation function, 2022. URL https://www.techrxiv.org/doi/full/10.36227/techrxiv.17283824.v2.

[143] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.*, 107:3–11, 2018. doi: 10.1016/j.neunet.2017.12.012. Special issue on deep reinforcement learning.

[144] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2017. URL https://arxiv.org/abs/1710.05941.

[145] Keysight. 81989a compact tunable laser source, 1465 nm to 157 5nm. https://www.keysight.com/us/en/product/81989A/compact-tunable-laser-source-1465nm-1575nm.html, 2025. [Online; accessed 4-December-2025].

[146] Rakesh Ranjan Kumar, Marina Raevskaia, Vadim Pogoretskii, Yuqing Jiao, and Hon Ki Tsang. Entangled photon pair generation from an inp membrane micro-ring resonator. *Appl. Phys. Lett.*, 114(2):021104, 01 2019. ISSN 0003-6951. doi: 10.1063/1.5080397.

[147] Andrea Marchisio, Andrea Carena, Vittorio Curri, and Paolo Bardella. PSO-assisted extraction of VCSEL parameters from LI and S21 measurements. In *2024 International Conference on Numerical Simulation of Optoelectronic Devices (NUSOD)*, pages 49–50. IEEE, 2024.

[148] Zhi-Hui Zhan, Jun Zhang, Yun Li, and Henry Shu-Hung Chung. Adaptive particle swarm optimization. *IEEE Trans. Syst. Man. Cybern. B Cybern.*, 39 (6):1362–1381, 2009. doi: 10.1109/TSMCB.2009.2015956.

[149] James Kennedy. The particle swarm: social adaptation of knowledge. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, pages 303–308, 1997. doi: 10.1109/ICEC.1997.592326.

[150] Ponnuthurai Nagaratnam Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)*, volume 3, pages 1958–1962, 1999. doi: 10.1109/CEC.1999.785514.

[151] Martin Grabherr. New applications boost VCSEL quantities: recent developments at Philips. In *Vertical-Cavity Surface-Emitting Lasers XVIV*, pages 9381–9394. SPIE, 2015. doi: 10.1117/12.2081790.

[152] Anjin Liu, Philip Wolf, James A. Lott, and Dieter Bimberg. Vertical-cavity surface-emitting lasers for data communication and sensing. *Photonics Res.*, 7(2):121, 2019. doi: 10.1364/PRJ.7.000121.

[153] Jim Tatum. *VCSEL Fundamentals*, chapter 2, page 29. John Wiley & Sons, Ltd, 2021. doi: 10.1002/9781119782223.ch2.

[154] Zihao Ma and Yu Li. Parameter extraction and inverse design of semiconductor lasers based on the deep learning and particle swarm optimization method. *Opt. Express*, 28(15):21971, 2020. doi: 10.1364/OE.389474.

[155] Pablo V. Mena, J.J. Morikuni, Sung-Mo Kang, Austin V. Harton, and K.W. Wyatt. A comprehensive circuit-level model of vertical-cavity surface-emitting lasers. *J. Lightw. Technol.*, 17(12):2612, 1999. doi: 10.1109/50.809684.

[156] Synopsys OptSim™. Optsim™ Photonic Simulations. https://www.synopsys.com/photonic-solutions/optsim.html, 2025. [Online; accessed 21-October-2025].

[157] Pablo V. Mena, J.J. Morikuni, and K.W. Wyatt. Compact representations of mode overlap for circuit-level VCSEL models. In *2000 IEEE Lasers and Electro-Optics Society Annual Meeting (LEOS)*, volume 1-6, pages 234–235, 2000. doi: 10.1109/LEOS.2000.890762.

[158] Naveen Bewtra, D.A. Suda, Gen Lin Tan, F. Chatenoud, and Jimmy M. Xu. Modeling of quantum-well lasers with electro-opto-thermal interaction. *IEEE J. Sel. Topics Quantum Electron.*, 1(2):331, 1995. doi: 10.1109/2944.401212.

[159] Alirio Melgar, Varghese A. Thomas, and Stephen E. Ralph. Multi-objective laser rate equation based parameter extraction using VCSEL small signal response and RIN spectra. *J. Lightwave Technol.*, 38(23):6437–6445, 2020. doi: 10.1109/JLT.2020.3015800.

[160] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 2 edition, 07 2006. doi: 10.1007/978-0-387-40065-5.

[161] Pierluigi Debernardi, Alberto Tibaldi, Markus Daubenschuz, Rainer Michalzik, Michele Goano, and Francesco Bertazzi. Probing thermal effects in VCSELs by experiment-driven multiphysics modeling. *IEEE J. Sel. Top. Quantum Electron.*, 25(6):1–14, 2019. doi: 10.1109/JSTQE.2019.2927580.

[162] A. Gullino, Alberto Tibaldi, Francesco Bertazzi, Michele Goano, Markus Daubenschuz, Rainer Michalzik, and Pierluigi Debernardi. Modulation response of VCSELs: a physics-based simulation approach. In *2020 Conference on Numerical Simulation of Optoelectronic Devices (NUSOD)*, pages 65–66, 2020. doi: 10.1109/NUSOD49422.2020.9217650.