Citation for published version (APA):
Bauwens, I., Bienstman, P., Verschaffelt, G., & Sande, G. V. D. (2025). Use of a Simple Passive Hardware Mask to Replace the Digital Masking Procedure in Photonic Delay-Based Reservoir Computing. IEEE Journal of Selected Topics in Quantum Electronics, 31(3). https://doi.org/10.1109/JSTQE.2024.3451113

Use of a simple passive hardware mask to replace the digital masking procedure in photonic delay-based reservoir computing

Ian Bauwens, Peter Bienstman, Guy Verschaffelt and Guy Van der Sande
Applied Physics Research Group, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
Photonics Research Group, Department of Information Technology, Ghent University-IMEC, Technologiepark Zwijnaarde 126, 9052 Ghent, Belgium

***Abstract*—Delay-based reservoir computing (RC) offers a conceptually simple approach to construct an RC system, often requiring only few components. Within this framework, one sequentially injects input data into the RC system. This data is time-multiplexed with a preprocessing mask to implement virtual nodes within a delay line of the reservoir. However, this preprocessing procedure limits the computing bandwidth of the system. In this work, we introduce a novel photonic RC system based on a hybrid approach of two different architectures. We investigate a combination of a passive spatially distributed integrated photonic RC system and a temporally distributed RC system - containing a semiconductor laser - into a compound architecture. Based on numerical simulations, we show that we are able to remove the complex mask preprocessing procedure which can ultimately increase the computing bandwidth of delay-based reservoir computing.**

***Index Terms*—Photonic reservoir computing, compound architecture, delay-based reservoir computing, spatially distributed reservoir computing, semiconductor laser.**

I. Introduction

THE use of artificial neural networks has gained a lot of interest in the last decade, and is mainly driven by the large amount of data which needs to be processed. Typically, artificial neural networks are trained using digital computers which rely on von Neumann architectures. As the number of trainable parameters of these networks is seen to increase dramatically in recent years, they can become very memory intensive and can require a large training time and energy consumption. Additionally, the end of Moore's law due to physical constraints on miniaturisation could limit the possibility to further increase the computational capacity with conventional electronic digital computing [1]–[4]. Photonic implementations of neural networks are therefore receiving considerable research interest nowadays, as these approaches potentially offer very fast and power-efficient processing, possibly further enhanced by the high amount of parallelism that can be achieved in photonics through various multiplexing schemes [5], [6].

Photonic reservoir computing (RC) represents such an approach to photonic neuromorphic computing. RC systems are a form of recurrent neural networks (RNNs) and consist of three distinct layers: an input layer, a reservoir and a readout layer. The input layer is designed to inject input data into the system. The reservoir constitutes a recurrent network layer which contains a large number of randomly interconnected nodes and which functions as a dynamical nonlinear system. The readout layer is where predictions are made using the input data and the reservoir state. With RC, the connection weights within the reservoir are fixed and remain unmodified. The trainable weights are solely located in the readout layer, drastically reducing the overall training duration compared to traditional recurrent neural networks [7]. Various (photonic) RC systems have shown their merit and have already been successfully used for solving complex tasks, such as nonlinear channel equalization tasks [8], [9], speech and pattern recognition [10]–[12] and time-series predictions [13]–[15].

Different approaches to implement photonic RC systems have been investigated, and successfully demonstrated. One such approach is temporally distributed RC, where a single nonlinear system is combined with delayed feedback to function as the reservoir. This system is referred to as delay-based RC (DB-RC). DB-RC systems typically have a very simple architecture where e.g. only a single microring resonator [16] or semiconductor laser (SL) needs to be used to implement the nonlinear node of the RC system [17]–[19]. For DB-RC, an external delay line is used to incorporate feedback within the system. However, this approach relies on a complex preprocessing of the input data - by the use of a preprocessing mask [20]–[26]. This masking procedure for the input data, which is explained in Section II, has to be performed at very high modulation speeds. This is often done using arbitrary waveform generators (AWGs), where a fixed preprocessing mask is applied to an input data sequence at fixed repetition rate. Because the nodes are constructed sequentially in time, this presents a bottleneck on the computing bandwidth. If we want to speed up the computation bandwidth of such RC systems, we have to make the temporal node separation of the preprocessing mask as small as possible, typically on timescales of the order of several ps [27]. This means that ultrafast modulation speeds are required for the data injection, where AWGs with bandwidths of several hundreds GHz or even THz have to be used to generate the preprocessing mask. Such AWG speeds are currently not commercially available.

Manuscript received April 30, 2024.

As such, we want to investigate other approaches which do not require the ultrafast generation of such mask signals. Such an approach can be realized using spatially distributed RC. In this approach, the nodes of the RC system are physically implemented using photonic components. Such RC systems can e.g. be implemented as integrated setups [7], [9], [28]–[30]. They have the advantage that they can operate faster compared to DB-RC, as they do not require any temporal multiplexing with a preprocessing mask. However, a drawback of these systems is that the nodes have to be physically connected in a photonic circuit, making it technologically challenging to create RC systems which many reservoir nodes. Moreover, if these nodes use active nonlinearities (e.g. making use of optical amplifiers to compensate for power losses in the reservoir network), this will result in a large power consumption. In this work, we combine the simplicity of the DB-RC network implementation with the speed advantage of the spatially distributed RC. Based on numerical simulations, we show that we are able to mitigate some of the disadvantages of both approaches by combining both into a single architecture. We will show that we can remove the costly preprocessing procedure of masking the input data, resulting in an overall speed-up of the system. We will also show that - when compared to passive spatially distributed RC - this scheme incorporates extra nonlinearity in a very inexpensive way.

The goal of this paper is to use a spatially distributed reservoir computing approach to generate the preprocessing mask for delay-based reservoir computing. To this end, we have to compare the performance of these approaches when used individually and when they are combined. We investigate three different implementations of combining the passive spatially distributed architecture with the DB-RC architecture. This paper is organised as follows. In Section II, we give an introduction to temporal DB-RC using an SL. In Section III, we give an introduction to spatially distributed RC using a passive four-port architecture (FPA). In Section IV, we introduce different architecture topologies where we combine both previously introduced approaches. In Section V, we numerically investigate the performance of the previously described architectures on the Santa Fe one-step ahead time-series prediction task, and in Section VI, we calculate their computational memory capacities. In Section VII, we give the conclusions of this paper.

II. Numerical implementation of delay-based reservoir computing using a semiconductor laser

The delay-based method for reservoir computing systems relies on a time-multiplexing approach to temporally implement virtual nodes [5], [8], [26], [31]–[35]. In this paper, we focus on one particular approach of this DB-RC scheme using a continuous wave (CW) single-mode SL where a delay (with external delay time τ_D) is introduced by self-coupling using optical feedback. This DB-RC scheme has been shown to yield good computational performance on several benchmark tasks [36], [37]. This approach is illustrated in Fig. 1. With DB-RC, one injects the input data samples u_k sequentially in the reservoir via the input layer. Before injecting the

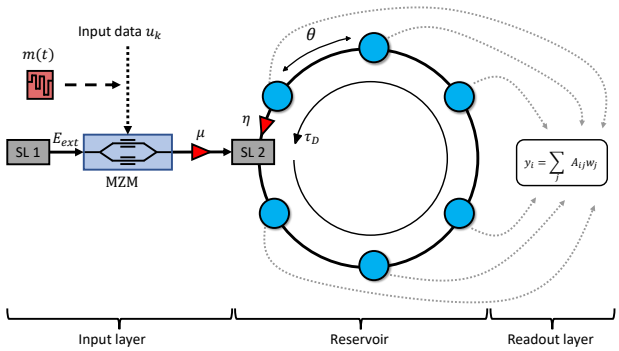

Fig. 1. Illustration of the DB-RC using a CW single-mode SL. The input data u_k is masked using a preprocessing mask $m(t)$ in the input layer and optically injected onto an external field E_{ext}, originating from SL 1, by a Mach-Zehnder modulator (MZM) at an injection rate μ. This signal is injected into the reservoir, which consists of a single-mode SL, SL 2, which is connected to itself with an external delay line with length τ_D, with a feedback rate η. The virtual nodes, shown as cyan circles, are defined within this delay line and have a node separation θ. The optical power at the virtual nodes (A_{ij}) is measured in the readout layer with photodetectors and combined with (trained) linear weights (w_j) to obtain the output (y_i).

input data into the reservoir, it is required that every discrete input sample u_k is stretched in time - to a duration τ_M - using a sample-and-hold procedure, and is multiplied with a preprocessing mask $m(t)$ (which has a duration of τ_M). This is done to ensure that the input data is coupled in a different way to the different virtual nodes and results in the input connection weights being different for different nodes. $m(t)$ typically consists of a piecewise constant function of N_N randomly selected sublevels, where each sublevel has a duration equal to the node separation θ. In this work, we have randomly selected the sublevels from 5 distinct values of $\{0, 0.2, 0.50, 0.75, 1\}$. After the reservoir, the intensity is measured by a photodetector in the readout layer. Recall that the input samples are time-multiplexed before they are injected in the reservoir. This means that the optical output - represented by the light intensity $|E(t)|^2$ - first needs to be de-multiplexed. This de-multiplexing is performed by sampling the intensity time trace at every θ for every input data sample. This results in N_N sampled intensities, which are stored in the columns of a matrix A_{ij}, which is done for all the input samples, stored in the rows of A_{ij}. The resulting matrix A_{ij} is referred to as the state matrix of the reservoir and is used for calculating weights w_j. This is done by using ridge regression so that the weights can be used for future testing with new, unseen input data samples. For the node separation, we use $\theta = 30\,\mathrm{ps}$, as this timescale results in the highest memory capacity [38] for delay-based reservoir computing when using an SL [39].

The dynamics of the single-mode SL are governed by the rate equations for the complex amplitude of the slowly-varying electric field $E(t)$ around a center angular frequency and the number of carriers in the active medium available for lasing $N(t)$ (both dimensionless quantities):

$$\frac{dE(t)}{dt} = \frac{1}{2}(1+i\alpha)\xi N(t)\,E(t) + \mu E_{inj}(t) + \eta E(t-\tau_D) + \tilde{F}_\beta(t) \quad (1)$$

$$\frac{dN(t)}{dt} = \Delta J - \frac{N(t)}{\tau_e} - \left(g + \xi N(t)\right)\left|E(t)\right|^2, \quad (2)$$

with α the linewidth enhancement factor, and ξ and g the differential gain and linear threshold gain. μ is the injection rate and η is the feedback rate. ΔJ is the excess pump current normalized by the elementary charge, so that $\Delta J = J - J_{thr}$, with J the pump current rate, and J_{thr} the threshold pump current rate of the SL. τ_e is the electron lifetime. To simulate the spontaneous emission noise, we use complex Gaussian white noise $\tilde{F}_\beta(t)$, with $\langle \tilde{F}_\beta(t) \rangle = 0$ and $\langle \tilde{F}_\beta(t)\tilde{F}_\beta(t')^* \rangle = \frac{\beta N(t)}{\tau_e}\delta(t-t')$. β controls the strength of spontaneous emission noise. In Table I we give an overview of the different parameters used in the simulations, and the values used in this paper. These values are based on previous research, in which it was shown these values are reasonable and that they lead to proper RC operation [37], [39].

Continuing with Eqs. (1) and (2), $E(t-\tau_D)$ represents the slowly-varying envelope amplitude of the time-delayed optical feedback signal which is injected back into the single-mode SL. $E_{inj}(t)$ represents the slowly-varying envelope amplitude of the electric field which is injected into the reservoir, i.e. the single-mode SL. We use an unbalanced Mach-Zehnder modulator (MZM) to optically inject the discrete input data u_k, as shown in Fig. 1, where the slowly-varying amplitude of the electric field at the output of the MZM is given by

$$E_{MZM} = E_{ext}\left(1 + e^{i(A_{MZM}S(t)+\Phi_{MZM})}\right). \quad (3)$$

$S(t) = m(t)u(t)$ is the masked data stream, E_{ext} is the amplitude of the external field and $u(t)$ is the input data stream normalized between $[0,1]$ (or $[-1,1]$ when considering negative input values). The modulation amplitude is given by A_{MZM}, and the bias of the MZM by Φ_{MZM}. The resulting electric field E_{MZM} is subsequently injected into the reservoir at an injection rate μ, where the injected signal is $E_{inj} = E_{MZM}e^{i\Delta\omega t}$. Note that we set the angular frequency detuning $\Delta\omega$ between E_{ext} and $E(t)$ to zero, i.e. we use the same frequency for the injection laser and for the center frequency of $E(t)$ in Eq. (1). We have also opted to take the input sample duration τ_M equal to the external delay length τ_D. A mismatch between these two can be introduced, as in e.g. Refs. [40]–[44], which could be beneficial to improve the performance of RCs, as demonstrated by Hülser et al. in Ref. [45]. In this work, however, we do not consider such a mismatch for simplicity and take $\tau_D = \tau_M$. The reason for this is purely out of simplicity, due to the fact that the optimum value of such a mismatch would need to be scanned for each type of simulation to optimize performance.

TABLE I
PARAMETERS, AND THEIR RESPECTIVE VALUES, WHICH ARE USED IN THE SIMULATIONS, UNLESS STATED OTHERWISE.

Parameter	Symbol	Value
Effective refractive index	n_{eff}	2.276
Group refractive index	n_g	4.564
Waveguide losses		1.50 dB/cm
Wavelength	λ	1550 nm
Injection rate (DB-RC)	μ	$98.1\,\mathrm{ns}^{-1}$
Injection rate (FPA/compound arch.)	μ	$100\sqrt{10}\ \mathrm{s}^{-1}$
Amplitude of the injected external field	E_{ext}	100
Feedback rate to SL (DB-RC)	η	$7.8\,\mathrm{ns}^{-1}$
Feedback rate to SL (compound arch.)	η	$10\sqrt{2}\ \mathrm{ns}^{-1}$
Number of nodes (DB-RC)	N_N	Variable
Number of nodes (FPA)	N_N	$16N_t$
Number of nodes (compound arch.)	N_N	$15N_t$
Input duration per sample (FPA)	τ_M	30 ps
Input duration per sample (DB-RC)	τ_M	$N_N{\cdot}30$ ps
Node separation (DB-RC)	θ	30 ps
Node separation (FPA/compound arch.)	θ	$\frac{\tau_M}{N_t}$
Internal delay time between couplers	τ_C	$\in$ [2.5,60] ps
External delay time (DB-RC)	τ_D	$N_N\theta$
External delay time (compound arch.)	τ_D	5 ps
Linewidth enhancement factor	α	3
Threshold gain	g	$1\,\mathrm{ps}^{-1}$
Differential gain	ξ	$5{\times}10^{-9}\ \mathrm{ps}^{-1}$
Spontaneous emission noise factor	β	10^{-6}
Carrier lifetime	τ_e	1 ns
Threshold pump current rate	J_{thr}	$10^{17}\ \mathrm{s}^{-1}$
Excess pump current rate (DB-RC)	ΔJ	$1.02J_{thr}$
Excess pump current rate (compound arch.)	ΔJ	$2J_{thr}$
Modulation amplitude of MZM	A_{MZM}	$\frac{\pi}{2}$
Bias of MZM	Φ_{MZM}	$\frac{\pi}{4}$

III. NUMERICAL IMPLEMENTATION OF SPATIALLY DISTRIBUTED RESERVOIR COMPUTING USING THE FPA

As mentioned in the introduction, we want to combine in this work a DB-RC system with a spatially distributed reservoir with the goal being to simplify the masking procedure. Before we discuss these novel compound systems, let us first discuss the operation of the spatially distributed reservoir considered in this work. Different from DB-RC, where we temporally implement virtual nodes, we can implement the nodes physically with spatially distributed RC. This has as a consequence that the spatially distributed RC systems do not require a masking procedure for injecting the input data. This results in spatially distributed RC systems being inherently faster than DB-RCs, as the nodes are physically implemented and not temporally. Furthermore, the nodes can be read out simultaneously and individually, which is not possible with DB-RC.

In this work, we have opted for using the passive spatially distributed FPA introduced in Ref. [9], which is inspired on the architecture presented in Ref. [7]. The FPA consists of a passive linear photonic network of 16 physical nodes, shown in Fig. 2, with the nodes arranged in a square 4×4 grid and connected by multiple waveguides. Every node consists of a multi-mode interference (MMI) coupler, represented as a blue circle in Fig. 2. Each node receives input from two adjacent nodes and redistributes its signal back to two other adjacent nodes. The two remaining (one input and one output) ports of the MMIs are used for data injection and detection and are shown in Fig. 2 on the left and right side of the network architecture for every individual node. Note that in this work, the indexing of the nodes starts at 0.

This article has been accepted for publication in IEEE Journal of Selected Topics in Quantum Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSTQE.2024.3451113

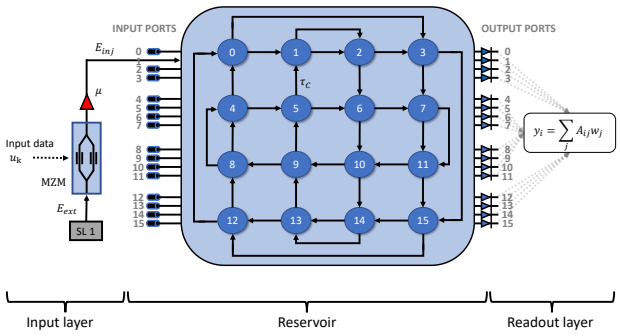

Fig. 2. Illustration of the FPA consisting of 16 multi-mode interference (MMI) couplers, shown as blue circles, which represent the nodes. Every node is connected to 4 ports of neighbouring nodes and contains an input and output port, used for data injection and detection. The input data u_k is optically modulated and injected by a Mach-Zehnder modulator (MZM) at an injection rate μ into the input port. The optical power at the output ports of the 16 physical nodes (A_{ij}) is measured using photodetectors in the readout layer and is used to train linear weights (w_j) corresponding to a target output (y_i). The internal delay time between nodes is represented by τ_C. In our simulations, port 1 is always used as input port.

An important parameter of the FPA is the length of the waveguide between two MMI couplers, which results in an internal delay time τ_C and a phase shift between nodes in the network. In practice, we can accurately control the internal delay time τ_C. However, the phase shift can fluctuate significantly due to fabrication imperfections and is less controllable compared to τ_C. The phase shift due to propagation between nodes is therefore chosen randomly from a uniform distribution $[0, 2\pi]$ for every waveguide, due to its uncontrollable nature in realistic setups, and is fixed at the initialization of the each simulation. Note that we do not incorporate an extra delay in our simulations for the MMI couplers. This effect would translate to a general delay for all MMI couplers, all assumed to be identical, and which can in practice be added to the internal delay time τ_C. The parameters we use for our simulations are shown in Table I. We use the PhotonTorch package [46] to numerically implement the FPA.

To inject the discrete input data samples u_k, we again use a sample-and-hold procedure where every input sample u_k is injected for a constant input duration per sample τ_M, resulting in a data stream $u(t)$. We have chosen to inject this input data into a single input port, arbitrarily chosen to be input port 1 here, as shown in Fig. 2. All other input ports are not being used. Due to the symmetry of the FPA, and some preliminary studies we have performed, it seems that the choice of the input port is not crucial as we obtain similar results when using other input ports. We again inject the data optically using an unbalanced MZM. The electric field after the MZM, E_{MZM}, is injected, at an injection rate μ, so that the injected signal $E_{inj} = E_{MZM} e^{i\Delta\omega t}$.

We construct the readout layer by measuring the optical power $|E(t)|^2$ of every output port of the FPA and sampling this time trace at every sampling event. In the conventional approach, this sampling event occurs once every τ_M. However, it is also possible to have multiple sampling events during one τ_M. In that case, N_t time-multiplexed temporal nodes can be defined and the sampling takes place every $\theta = \frac{\tau_M}{N_t}$, so that the total number of virtual nodes N_N increases linearly with increasing N_t. Remark that in the latter case, we do not need a preprocessing mask in the input layer to implement the temporal nodes. This is in stark difference to the approach described in Section II, which requires such a preprocessing mask to allow for transient dynamics at the virtual nodes. This results in N_N sampled intensities, which are stored in the columns of the state matrix A_{ij}, which is done for all the input samples, stored in the rows of A_{ij}. The state matrix of the reservoir, A_{ij}, is used for calculating linear weights, w_j, during training on labeled data, y_i, and for testing on unseen data. Note, that we set $N_t = 1$ for the FPA - unless mentioned otherwise. For the FPA, we use the output ports of all 16 nodes, which means that we are able to construct a total of $N_N = 16$ nodes. If $N_t > 1$, however, the total number of nodes is $N_N = 16 N_t$.

IV. Numerical implementation of combined architectures

In this section, we numerically investigate three different implementations of combining the passive FPA with the DB-RC architecture. The first architecture, in Section IV-A, is a proof-of-principle approach and consists of two separate parts: an FPA, in which we inject input data and measure the optical power at a fixed output port. The measured optical power is then optically injected into the input layer of the DB-RC architecture. Therefore, the FPA functions in this case as a separate mask generator. The goal of investigating this architecture is to verify if the FPA can be used to replace the mask generator. The second architecture, in Section IV-B, has a similar approach. Here, however, we directly inject the electric field output of the FPA into the DB-RC, instead of first measuring the optical power and optically injecting this signal, making this a simpler architecture for more realistic implementations due to its direct integration as a single setup. Lastly, the third architecture, in Section IV-C, consists of an FPA and SL in an external delay loop. This architecture combines both spatially and temporally distributed RC where we now use all of the available physical nodes of the FPA, together with temporal (i.e. virtual) nodes.

A. Proof-of-principle: FPA as separate mask generator for DB-RC

The first architecture we investigate consists of a combination of the FPA and the DB-RC architecture, where the FPA functions as a separate mask generator for DB-RC. This architecture is shown in Fig. 3. This approach is intended as a proof-of-principle to generate masks in an analog procedure. To this end, we inject input data into input port 1 of the FPA using an SL and MZM, identical to the techniques described in Section III using the sample-and-hold procedure. However, different to this section is that the masking is performed by the FPA and that we choose the input duration per sample fixed to $\tau_M = N_N \theta$, with θ the temporal node separation and N_N the number of nodes in the DB-RC. In this case, we

This article has been accepted for publication in IEEE Journal of Selected Topics in Quantum Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSTQE.2024.3451113

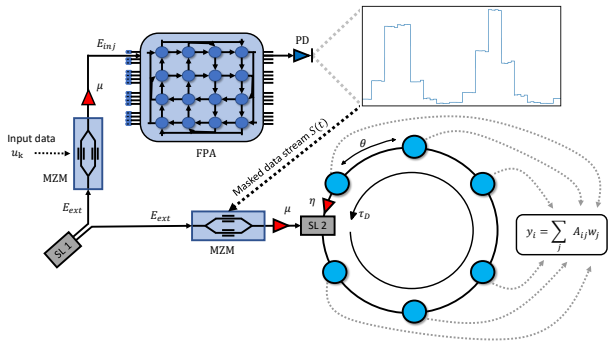

Fig. 3. Illustration of the first approach to use the FPA as mask generator for DB-RC. We optically modulate the input data u_k onto E_{ext} of SL 1 using a Mach-Zehnder modulator (MZM) and inject this signal at an injection rate μ into input port 1 of the FPA. The optical power at output port 1 of the FPA is subsequently measured by a photodetector (PD) and normalized to represent the masked data stream $S(t)$. $S(t)$ is then optically injected in the single-mode laser SL 2, using an MZM. SL 2 is connected to itself with an external delay line with length τ_D, with a feedback rate η. The virtual nodes, shown as cyan circles, are defined within this delay line and have a node separation θ.

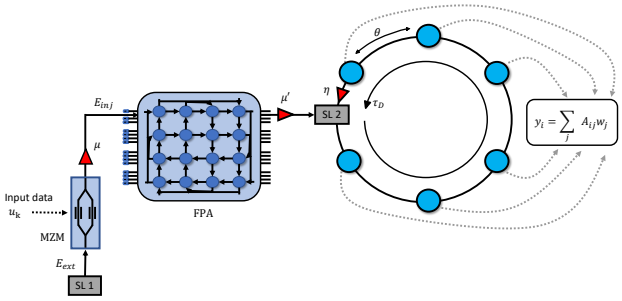

Fig. 4. Illustration of the second approach to use the FPA as mask generator for DB-RC. In this approach, we inject the (normalized) electric field resulting from the FPA directly into the DB-RC. The input data u_k is optically modulated onto E_{ext} of SL 1 and is injected by a Mach-Zehnder modulator (MZM) at an injection rate μ into input port 1 of the FPA. Output port 1 of the FPA is connected - at an injection rate μ' - to a single-mode SL, SL 2, which is connected to itself with an external delay line with length τ_D, with a feedback rate η. The virtual nodes, shown as blue circles, are defined within this delay line and have a node separation θ.

choose $\theta = \tau_C$, which is motivated by the fact that the output of the FPA shows variations at this timescale τ_C in a similar fashion as the mask preprocessing modulates the input signal at a timescale of θ. Thus, we again take $\theta = 30\,\text{ps}$, which was shown to be a good choice for θ [38], [39]. Therefore, instead of creating the masked signal's sublevels with a preprocessing mask, the masked signal is naturally generated from the optical signal propagating through the FPA. The resulting power time trace is subsequently measured at output port 1 of the FPA by a photodetector and normalized to values between 0 and 1, resulting in a masked data stream $S(t)$. $S(t)$ is subsequently modulated onto E_{ext} using an MZM. The output of this MZM is injected into the separate DB-RC architecture. The subsequent readout (including training and testing) is identical to the previously explained DB-RC approach, in Section II. We would like to emphasize that this method of measuring the power trace and optically injecting it using an MZM is rather inefficient due to the unnecessary conversion between electric and optical domain. The purpose of this implementation is therefore purely as a proof-of-principle demonstration: this approach is meant to check the feasibility of the FPA to function as mask generator instead of using a fast AWG to pre-process the input data stream.

B. FPA as integrated mask generator for DB-RC

The second architecture we investigate consists of a combination of the FPA and DB-RC architecture, where the FPA again functions as a mask generator for DB-RC. This architecture is shown in Fig. 4. We again use input port 1 and output port 1 of the FPA for data injection and output. Different from Section IV-A, we now directly inject the (normalized) output of the FPA into the DB-RC, without the requirement of measuring the power time trace using a photodetector. This is an improvement on the previous implementation, as we now do not require another AWG and MZM to optically inject the measured power time trace. Contrary to the previous section, this implementation is able to retain the phase information of the electric field of the output of the FPA, which is known to have an important role when injecting input data into DB-RC [39]. The subsequent readout (including training and testing) is identical to the previously explained DB-RC approach, in Section II. In order to have similar injection strengths as with other input configurations for DB-RC, the electric field - which is injected into the DB-RC - is normalized and injected at an injection rate of $\mu' = 100\mu$. This higher value for μ' is used in order for the signal injected into the DB-RC to be of similar strength as for the other setups in this work. The input duration per sample is identical to Section IV-A, at $\tau_M = N_N \cdot 30\,\text{ps}$.

C. Compound RC: Exploiting the physical nodes of the FPA by placing it in the delay loop of DB-RC

The previous two architectures described in Sections IV-A and IV-B allow us to investigate the potential of the FPA as mask generator. However, both architectures still have a relatively slow computing bandwidth. This is because the nodes are time-multiplexed in the external delay line, where we have the requirement to roughly match the node separations to the response time of the SL to achieve a good performance.

It might be more useful to place the FPA in the delay loop of DB-RC, such that the different nodes of the FPA can be used as additional reservoir nodes while still not needing a preprocessing mask. This is what we want to explore in the next section. This novel compound architecture is shown in Fig. 5.

The injection of input data into this architecture is identical as the procedure described in Section III, i.e. using the sample-and-hold procedure without the preprocessing with a mask $m(t)$. The standard input port is again arbitrary chosen to be port 1 of the FPA, while all the other input ports of the architecture are not used for data injection. One of the output ports of the architecture is connected with an external delay line.

This article has been accepted for publication in IEEE Journal of Selected Topics in Quantum Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSTQE.2024.3451113

This results in the output of one node not being available for the readout layer, and which we refer to as the feedback port. For this feedback port we have chosen output port 1. This choice is supported by the fact that we can introduce the nonlinearity of the SL early on in the architecture (because we inject data in input port 1). In Ref. [47], we found that a large injection rate is needed to achieve injection locking of the SL, which is needed in order to obtain good computational performance e.g. on the Santa Fe benchmark task. In the case of a large injection rate, the power at the output ports of the FPA then becomes large enough to achieve injection locking of the SL. Furthermore, we found in Ref. [47] that the performance is largely independent of the delay of the external feedback loop (for τ_D between 5 and 60 ps) so that it does not need to be precisely optimized for the Santa Fe task. As a result, a small external delay time can be chosen, which results in a small footprint. The internal delay time τ_C plays a more important role, and needs to be chosen shorter than the input duration τ_M. The power of all other 15 remaining output ports are measured and are used for the readout layer. We conjecture that introducing extra nonlinearity - using a single output port as feedback port - has a positive effect on the performance which more than compensates for the loss in performance due to the loss of this output port for training. The corresponding reservoir state matrix, indicated by A_{ij} in Fig. 5, is used for calculating linear weights, w_j, during training. The optical signal from the feedback port is injected back into a single-mode SL (SL 2) using an external delay line. Therefore, the delayed feedback term of Eq. (1) is now replaced by the electric field at the feedback port of the FPA. The resulting field emitted by the SL is injected back into the 2×1 MMI coupler, which is connected to input port 1 of the FPA. The value for the external delay time τ_D of the delay line from the laser to the FPA is fixed to $\tau_D = 5$ ps. The waveguide between the MZM and 2×1 MMI coupler, and the waveguide between the 2×1 MMI coupler and FPA are both coded in our simulations to have a length of zero, so that they do not contribute any extra timing effects. This is because these delays can always be incorporated in the value of τ_C or τ_D. In our simulations, we fix $\tau_M = 30$ ps. We vary the values for the internal delay time τ_C, where τ_C takes values between 2.5 ps and 60 ps. The values used for simulating the compound architecture are given in Table I. The higher value for the excess pump current rate of SL 2, compared to DB-RC, was chosen such that we have a higher optical power within the FPA to compensate for propagation losses in the FPA.

The readout of the compound architecture is similar to that of the FPA, discussed in Section III. We construct the readout layer by measuring the optical power $|E(t)|^2$ of every output port of the FPA and sampling this time trace at every sampling event. This sampling event occurs every $\theta = \frac{\tau_M}{N_t}$, with N_t the number of temporal nodes, for every input sample. This results in N_N sampled intensities, which are stored in the columns of the state matrix A_{ij}, which is done for all the input samples, stored in the rows of A_{ij}. The state matrix of the reservoir, A_{ij}, is used for calculating linear weights, w_j, during training on labeled data, y_i, and for testing on unseen data. For the compound architecture, we use the output ports of 15 nodes, which means that we are able to construct a total of $N_N = 15N_t$ virtual nodes.

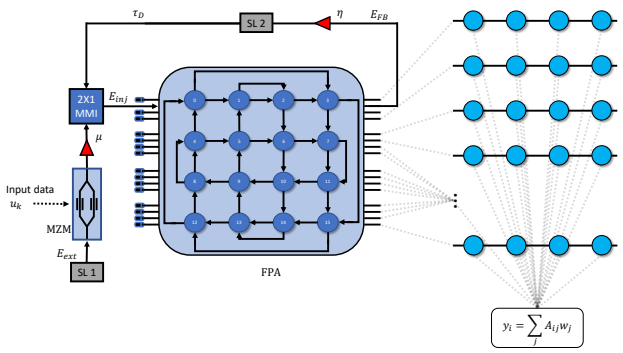

Fig. 5. Illustration of the compound architecture. The input data u_k is optically modulated and injected into the input port by a Mach-Zehnder modulator (MZM) at an injection rate μ and a 2×1 multi-mode interference (MMI) coupler. The feedback (FB) port is connected to an external delay line with delay time τ_D and a single-mode SL with feedback rate η. The optical power at the other 15 output ports of the FPA is measured in the readout layer, where every physical node (shown in blue) contains N_t virtual nodes (shown here in cyan for $N_t = 4$ virtual nodes). The internal delay time between nodes is represented by τ_C. In our simulations, we use input port 1 as input port and output port 1 as feedback port.

V. Numerical results: Santa Fe time-series prediction task

To investigate the performance of the various architectures which were described in the previous sections, we compare their performance on the Santa Fe time-series prediction task. We perform a one-step ahead prediction on this time-series, which contains just over 9000 data points sampled from a far-IR laser in a chaotic regime [48]. In order to train the weights of the readout layer, we take 3010 data samples from the beginning of this time-series. The resulting state matrix $\mathbf{A}$ is then used to find the weights $\mathbf{w}$,

$$\mathbf{w} = \mathbf{A}^{\dagger}\mathbf{y}, \tag{4}$$

where $\mathbf{y}$ is the labeled, expected output and with $\dagger$ being the Moore–Penrose pseudo-inverse. The next 1010 data samples of the Santa Fe time-series are used to perform the testing. The 10 first data samples of both sets are used for initializing the system and are discarded to remove any transients due to this initialization. These data samples are therefore not used for training and testing purposes. As a measure for the performance, we use the normalized mean squared error (NMSE) on the test set, where the NMSE is defined as

$$\mathrm{NMSE}(\mathbf{y},\hat{\mathbf{y}}) = \frac{\left\langle (\mathbf{y}-\hat{\mathbf{y}})^2 \right\rangle}{\left\langle \left(\mathbf{y}-\langle \mathbf{y}\rangle\right)^2 \right\rangle}, \tag{5}$$

with $\hat{\mathbf{y}}$ the predicted output data, where $\hat{\mathbf{y}} = \mathbf{A}\mathbf{w}$. The best performance corresponds to the lowest NMSE, with typical NMSE values for RC systems on the Santa Fe one-step ahead prediction task being around 10^{-2} [17], [39], [49].

This article has been accepted for publication in IEEE Journal of Selected Topics in Quantum Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSTQE.2024.3451113

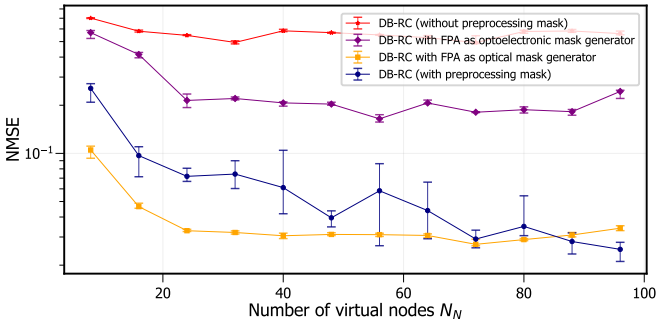

Fig. 6. Median and interquartile range of the NMSE as a function of the number of virtual nodes N_N of various architectures, over 10 simulation runs with different phases between nodes/mask initializations.

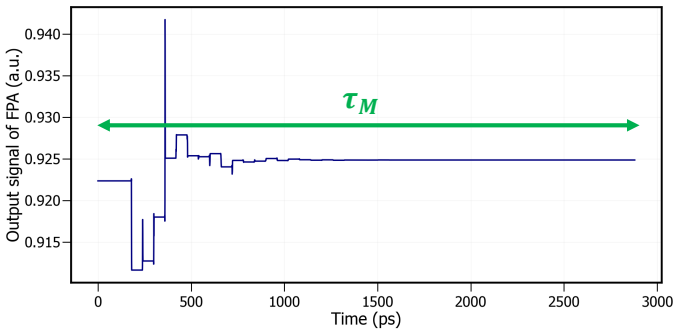

Fig. 7. Output signal of the FPA corresponding to one input sample of duration τ_M. This signal is used as masked data stream $S(t)$ in the approach described in Section IV-A, with the total number of virtual nodes $N_N = 96$.

A. Varying number of virtual nodes N_N

In this section, we investigate the feasibility of using the FPA as a mask generator for input data, instead of manually masking the input data with a preprocessing mask $m(t)$. To this end, we compare the DB-RC with preprocessing mask from Section II with the implementations (without preprocessing mask) from Sections IV-A and IV-B. In Fig. 6, we show the performance of these different approaches, when varying the number of virtual nodes N_N. We repeat the simulation 10 times, with 10 different random preprocessing masks (for the system of Section II) or 10 different random sets of phases between FPA nodes (for the systems of Sections IV-A and IV-B). In this figure, we also show the performance of using the FPA separately as mask generator to the DB-RC (described in Section IV-A) and using the FPA directly as mask generator (described in Section IV-B).

We observe that the DB-RC without any masking results in the worst performance as a function of the number of nodes. This is shown in Fig. 6 by a near constant NMSE around 6×10^{-1} for this system. When a masking procedure is used, we observe a much better performance, where the NMSE decreases with increasing number of nodes N_N. This indicates the importance of the masking procedure, which is crucial for DB-RC to achieve a good performance. The performance of the architecture where the FPA is separately used as mask generator for the DB-RC is situated between the previous two cases. This shows that the FPA is able to supply effective masking to the input signal such that DB-RC can be used to process the input data. However, the masking ability of the FPA seems to be inefficient when the number of nodes becomes large. In Fig. 6, we can see that the NMSE does not decrease when $N_N > 24$ for this approach. The other approach of using the FPA as integrated mask generator, however, shows the best performance of all considered implementations. The resulting NMSE of this implementation decreases up to NMSE $\approx 3 \times 10^{-2}$, at $N_N \approx 24$, and saturates at this NMSE for increasing N_N. Its performance, however, deteriorates for $N_N > 80$ with a slightly increasing NMSE.

We observe that the FPA as mask generator - for both the separate and integrated approach - shows a saturation in performance with increasing N_N. Recall that in both these approaches, we have chosen the input sample duration proportional to the total number of nodes, i.e. $\tau_M = N_N \cdot 30\,\text{ps}$. The fact that both approaches show a saturation in NMSE with increasing number of nodes N_N can most likely be explained by the fact that for small N_N, there are large temporal fluctuations in the sampled output signal, resulting in the signal of the various (virtual) nodes to be different. However, if N_N is large, the output signal's temporal fluctuations for a given input sample become small towards the end of the input sample. As a result, there will be little difference between the signals of virtual nodes at the end of the delay loop. Increasing N_N further will then have little effect on performance. This is illustrated in Fig. 7, where we show the intensity at output port 1 of the FPA for $N_N = 96$. This intensity time trace resembles a typical masked data stream. When a new data sample is presented at the input of the FPA, it will elicit temporal fluctuations at the FPA output that gradually become smaller. After a time-interval of about 1 ns, these fluctuations become negligibly small. This corresponds to around 20 virtual nodes that can effectively be used in the DB-RC (taking into account that $\theta = \tau_C$), and this number of effective virtual nodes is approximately equal to the number of spatial nodes in the FPA. The number of virtual nodes can in principle be further increased by using an FPA that contains more spatial nodes, as in that case the temporal fluctuations in the output of the FPA will remain substantial during a longer time interval. Additionally, the bandwidth of the approach of Sections IV-A and IV-B is limited by the fact that the nodes are implemented temporally within the external delay line. Nonetheless, the results of Figs. 6 and 7 show that the FPA has the ability to provide diversity to the constant input signal, without the need for high-speed AWGs to construct the masked input signal. This effectively results in the FPA to function as a passive mask generator for DB-RC, and works especially well when the FPA is directly integrated.

B. Varying internal delay time τ_C in the FPA and compound architecture

In the previous section we have shown that the FPA can be used as a passive mask generator for DB-RC, even outperforming the conventional approach of applying a fixed

This article has been accepted for publication in IEEE Journal of Selected Topics in Quantum Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSTQE.2024.3451113

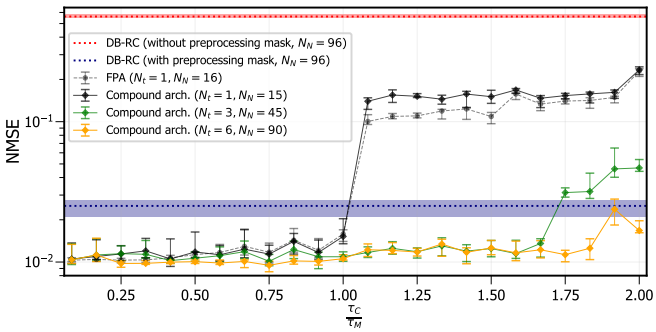

Fig. 8. Median and interquartile range of the NMSE as a function of τ_C/τ_M of the FPA and the compound architecture, over 10 simulation runs with different phases between nodes. The performances for the different number of temporal nodes N_t during one τ_M are shown in different colors, with solid and dashed lines for the different architectures. The median and interquartile range of the NMSE of the single-mode delay-based RC with and without preprocessing mask are also shown in dotted lines and colored regions, over 10 different preprocessing mask initializations when the mask is applied.

preprocessing mask to DB-RC. Inspired by these approaches, we investigate in this section the performance of the DB-RC and FPA, described in Sections II and III, and the newly introduced compound architecture, described in Section IV-C, for different N_t. We study the role of an important parameter of the FPA, the internal delay time τ_C, by scanning this value and measuring the performance on the Santa Fe one-step ahead prediction task. For the compound architecture, we fix the external delay time at $5\,\text{ps}$ in order to have a small delay coming from the SL, allowing for a small footprint. For the FPA and compound architecture, we use an input duration per sample $\tau_M = 30\,\text{ps}$. For the compound architecture, we vary the number of temporal nodes between $N_t = 1$ and $N_t = 6$. We repeat the simulations 10 times with different random phase initializations for the waveguides in order to incorporate some statistics. In Fig. 8 we show the median and interquartile range of the NMSE as a function of τ_C/τ_M for the FPA and compound architecture, while varying the number of virtual nodes N_t for the compound architecture.

The performance of the DB-RC with a single-mode SL - with and without a preprocessing mask - are shown with dotted lines and colored regions for $N_N = 96$ virtual nodes for 10 simulation runs with different preprocessing mask initializations. We have chosen this value for N_N because it corresponds to the best performance found in Section V-A for DB-RC with a preprocessing mask.

We observe in Fig. 8 that the performance of the compound architecture for $N_t = 1$ is similar to that of the FPA. At $\tau_C/\tau_M \leq 1$, the FPA and compound architecture have similar and good performance with an NMSE around $\approx 10^{-2}$. For $N_t = 1$, we observe that the performance of the compound architecture worsens at $\tau_C/\tau_M > 1$. If we increase N_t to 3, the performance of the compound architecture remains good (i.e. resulting in a low NMSE) up until $\tau_C/\tau_M > 1.65$. If N_t is further increased to 6, the region of good performance widens further up until $\tau_C/\tau_M > 1.85$. This means that for $N_t > 1$, we are able to achieve larger range of τ_C/τ_M where the NMSE remains low, at around $\approx 10^{-2}$. In Fig. 8, we observe that the performance of the DB-RC without a preprocessing mask again results in the worst performance, around NMSE $\approx 6 \times 10^{-1}$. This is in contrast to the performance of the DB-RC with a preprocessing mask which results in a much better NMSE, around 2.5×10^{-2}.

The explanation for the worsening performance of the compound architecture for $N_t = 1$ at $\tau_C/\tau_M > 1$ is as follows. If the internal delay time τ_C becomes longer than the input duration per sample τ_M, a segment of the input data stream still propagates between the nodes in the network and has yet to reach the readout layer before the next sampling event occurs. However, the previous segments will already have reached some of the nodes when this sampling event takes place. This means that the nodes only contain information of the older inputs, while we still need to predict the next input sample. This means that the reservoir is implicitly performing a multi-step ahead prediction, which is more difficult to solve compared to a one-step ahead prediction. When $N_t > 1$, we observe a good performance even when $\tau_C/\tau_M > 1$. The explanation for this is due to the increased number of temporal nodes, which allow us to use more information of the signals inside the architecture. As we now use more than one output sample for every input sample, this leads to more diversity within the (virtual) nodes and an increased memory (the latter we will show in Section VI). Note that the compound architecture has a better performance when compared to DB-RC with a single-mode SL with the same number of virtual nodes. This improvement in performance can be understood by the fact that instead of a fixed preprocessing mask, in case of the DB-RC, we instead have an artificial (analog) mask created by the topology of the FPA, which has a more diverse response to the input.

The advantage of the compound architecture is that its masking is performed implicitly by the FPA, which is a passive architecture. Furthermore, a different advantage is that the computing bandwidth of the compound architecture is much higher than that of the DB-RC. This is because the masking procedure, which is required for DB-RC to match the node separation with the timescales of the laser dynamics, dramatically decreases the computation bandwidth compared to the compound architecture. To illustrate this, the DB-RC architecture has a node separation around $\theta = 30\,\text{ps}$ with $N_t = 96$ nodes, resulting in $\tau_M = N_t\theta = 2.88\,\text{ns}$ for every input data sample. In contrast, the input duration per sample for the compound architecture is $\tau_M = 30\,\text{ps}$. This means that we are able to achieve a speed-up of almost two orders of magnitude compared to DB-RC.

We conclude that using the compound architecture allows us to improve the computing bandwidth - by removing the necessity of using a preprocessing mask for input data. We remark that both the FPA and compound architecture have a similar performance for $\tau_C/\tau_M \leq 1$ (with NMSE around 10^{-2}) on the Santa Fe prediction task. This limit on accuracy is also observed with other reservoir and echo state networks [50], which indicates that achieving a lower NMSE is hard.

However, the additional temporal nodes that can be sampled with the compound architecture (i.e. $N_t > 1$) increases the number of (virtual) nodes available for computing, which is expected to lead to better computational performance for difficult tasks. This will be investigated in more detail in the next section.

VI. Numerical results: Memory capacity

In the previous section, we have investigated the performance of various RC architectures on the one-step ahead Santa Fe time-series prediction task. However, the Santa Fe task does not generalize the performance of the architectures on other benchmark tasks. Instead of selecting other benchmark tasks, we have chosen to study a task-independent measure of the computational capacity of RC systems which can quantify their performance on other tasks. We quantify - based on numerical simulations - the linear and nonlinear memory capacity of DB-RC, the FPA and the compound architecture, for different values of N_t, using the techniques from Ref. [51]. For this, we inject an input data stream, originating from a sample-and-hold procedure of discrete input samples u_k which are drawn from a uniform distribution between -1 and 1. We then train the readout layer to reconstruct products of normalized Legendre polynomials of l-delayed previous input samples u_{k-l}. In this case, we limit this delayed input to $l \leq 10$. We inject a total of 500010 input samples from the uniform distribution, and discard the first 10 samples which are used for initialization of the architecture.

The target data y_i for the calculation of the memory capacity is constructed by the product of normalized Legendre polynomials of a given degree δ, from previous inputs.

The mean squared error (MSE) between the expected signal y_i and the predicted signal $\hat{y}_i$ is then calculated for all input samples. The MSE, for a specific product combination of Legendre polynomials δ and a set of specific indices of past inputs $(l_1, \dots)$, is used for calculating the memory capacity,

$$C_{\delta,(l_1,\dots)} = 1 - \frac{\left\langle (\mathbf{y} - \hat{\mathbf{y}})^2 \right\rangle}{\langle \mathbf{y}^2 \rangle}, \tag{6}$$

where one takes the average over all input samples and where $C_{\delta,(l_1,\dots)} \in [0, 1]$. Summing over all possible combinations of the delayed elements $(l_1, \dots)$, for a specific combination of Legendre polynomials δ, and summing over all possible Legendre combinations which result in the combined degree of d, we can find the memory capacity of degree d,

$$C_d = \sum_{\delta} C_\delta = \sum_{\delta} \sum_{(l_1,\dots)} C_{\delta,(l_1,\dots)}. \tag{7}$$

The sum of all memory capacities over all degrees results in the total computational memory capacity of the system, $MC = \sum_d C_d$. MC is limited to the total number of nodes within the architecture, which was proven in Ref. [51]. It is important to note that a better indication for the performance on different benchmark tasks is given by the individual memory capacities per degree C_d, shown in Ref. [52], in contrast to the total memory capacity MC. Note that we potentially overestimate the value of C_d due to the finite number of input data samples that are being used. To avoid this, we use a threshold capacity, where we set values for $C_{\delta,(l_1,\dots)}$ below C_{thr} equal to zero when calculating the memory capacities, as discussed in Ref. [51]. The value for C_{thr} changes depending on the total number of nodes, and varies from $C_{thr} \approx 2 \times 10^{-4}$ for $N_t = 1$ to $C_{thr} \approx 6 \times 10^{-4}$ for $N_t = 6$.

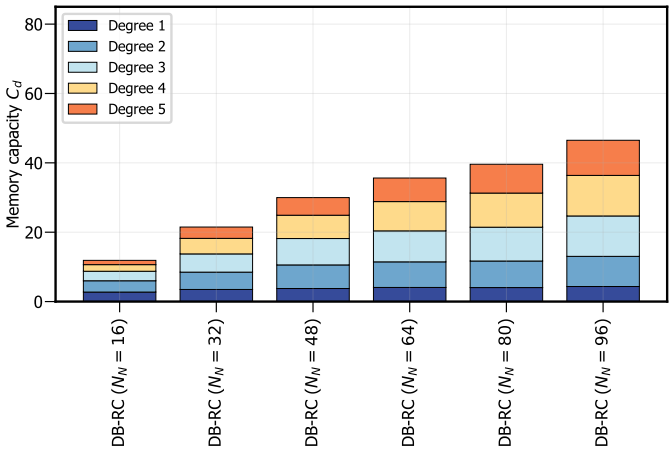

Fig. 9. Memory capacity up to the fifth degree for the DB-RC architecture, for $N_N = 16$ to $N_N = 96$.

In Figure 9, we show the memory capacity up to the first five degrees in the case of DB-RC. The reason for only calculating the memory capacities of the first five degrees is that this already gives a good indication of the nonlinearity within the systems. We observe that the total memory capacity increases with increasing number of nodes N_N, although not reaching the theoretically possible total memory capacity (which is equal to the number of nodes N_N). Note that for increasing N_N, the nonlinear memory capacity up to the fifth degree increases. These findings are used as reference for later results.

In Fig. 10(a)-(f), we show the memory capacity up to the first five degrees in the case of the FPA (left hand side of every subfigure) and compound architecture (right hand side of every subfigure) for $N_t = 1$ to $N_t = 6$. For the compound architecture, we have fixed the external delay time to $\tau_D = 5\,\text{ps}$. We have chosen the internal delay time τ_C ranging from $\tau_C = 4\,\text{ps}$ to $\tau_C = 14\,\text{ps}$, shown in Fig. 10(a)-(f). The specific values for τ_C shown in these subfigures are chosen to avoid τ_C being a multiple of the smallest sampling time, given by $\tau_M/N_t = 30\,\text{ps}/6 = 5\,\text{ps}$ (except for the interesting case where $\tau_C = \tau_D = 5\,\text{ps}$).

In Fig. 10(a)-(f), we observe that for all the investigated τ_C values, the total memory capacity MC increases with N_t for both the FPA and the compound architecture. This clearly indicates that taking multiple output samples during one input sample is beneficial, and thus results in temporal multiplexing as in DB-RC, however, without referring a masking proce-

This article has been accepted for publication in IEEE Journal of Selected Topics in Quantum Electronics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSTQE.2024.3451113

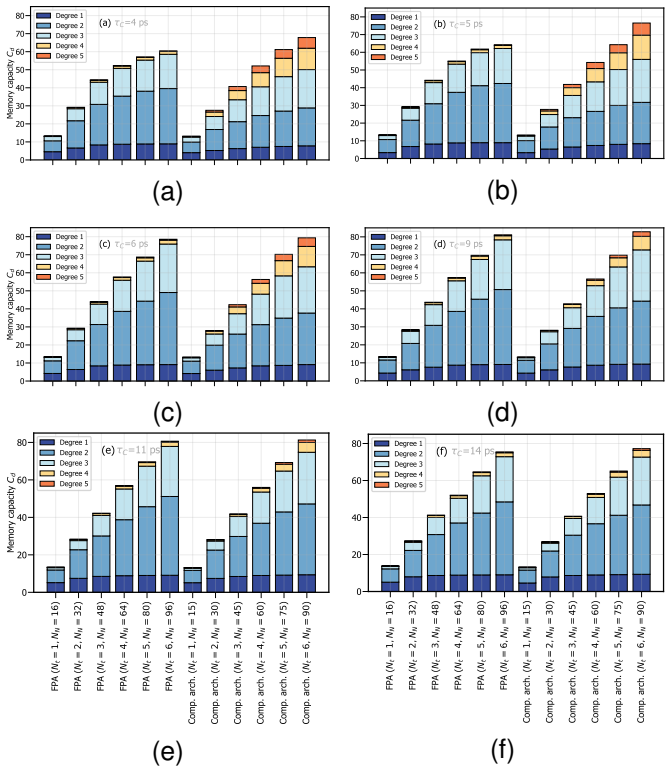

Fig. 10. Memory capacity up to the fifth degree for the FPA and compound architecture, for $N_t = 1$ to $N_t = 6$ (resulting in $N_N = 16N_t$ for the FPA and $N_N = 15N_t$ for the compound architecture), at different values for the internal delay time τ_C.

dure and without reducing the input sample rate. We also observe that the total memory capacity increases with τ_C, until $\tau_C \approx 9\,\text{ps}$. For $\tau_C > 9\,\text{ps}$, the total memory capacity of both architectures decreases with increasing τ_C. This is most likely due to increasing waveguide propagation losses when τ_C increases.

For small τ_C, between $\tau_C = 4\,\text{ps}$ and $\tau_C = 5\,\text{ps}$ - shown in Fig. 10(a)-(b) - we observe very distinct distributions of the memory capacity for the FPA and compound architecture. Most fascinating is that the total memory capacity for $N_t = 6$ is much higher for the compound architecture, compared to the FPA. This is remarkable, as the theoretical upper limit for the total memory capacity for the compound architecture is lower (maximum $MC = 15N_t$) compared to the FPA (maximum $MC = 16N_t$). We observe that the linear memory capacity (i.e. $C_{d=1}$) remains fairly constant when increasing N_t, both for the FPA and compound architecture. However, the quadratic and cubic memory capacities (i.e. $C_{d=2}$ and $C_{d=3}$) increase with increasing N_t for both architectures, but are higher for the FPA compared to the compound architecture. We also observe that the compound architecture has much higher nonlinear memory capacities for high degrees (i.e. $C_{d>3}$) compared to the FPA, for $N_t > 1$. This indicates that the nonlinearity is significantly increased at the very fast sampling regime. For the FPA, these memory capacities are not significantly increased when N_t increases.

For τ_C between $\tau_C = 6\,\text{ps}$ and $\tau_C = 9\,\text{ps}$ - shown in Fig. 10(c)-(d) - we observe similar total memory capacities for the FPA and compound architecture, for the same N_t. However, the distributions of these memory capacities again differ between both architectures. The compound architecture has a much larger nonlinear memory capacities (of degree 4 and higher) compared to the FPA. Conversely, the linear, quadratic and cubic memory capacities are again larger for the FPA compared to the compound architectures, and increase for increasing N_t for both architectures. The linear memory capacity (i.e. $C_{d=1}$) also here remains fairly constant when increasing N_t, both for the FPA and compound architecture.

For τ_C between $\tau_C = 11\,\text{ps}$ and $\tau_C = 14\,\text{ps}$ - shown in Fig. 10(e)-(f) - we observe that the FPA has similar memory capacities as the compound architecture. We again observe that the linear memory capacity remains fairly constant when increasing N_t, both for the FPA and the compound architecture. The quadratic and cubic memory capacities again increase with increasing N_t for both architectures. The memory capacities of higher degrees remain small for both networks, independent of N_t.

We have also investigated the linear and nonlinear memory capacity of larger τ_C (not shown here). We observe that the total memory capacity tends to decrease with increasing τ_C. This negative effect is more pronounced for the FPA than for the compound architecture, where the compound architecture is still able to retain a higher total memory capacity compared to the FPA for identical τ_C. For example, at $\tau_C = 30\,\text{ps}$ (not shown here) we observed that the FPA has a low total memory capacity ($MC \approx 12$), independent of N_t. This is in contrast to the compound architecture, which still has a very high MC and which increases with N_t (i.e. $MC \approx 12$ at $N_t = 1$ up to $MC \approx 72$ at $N_t = 6$).

The increased high nonlinear memory capacity at small τ_C of the compound architecture for high degrees, $C_{d>3}$ - compared to the FPA - indicate that increasing the number of temporal nodes (i.e. increasing N_t) has a positive effect on the amount of nonlinearity of the compound architecture. This shows that the nonlinearity of the compound architecture is used effectively when $N_t > 1$ and τ_C is small. When we compare the memory capacity of the compound architecture to the DB-RC, we find that it is able to achieve, at e.g. $\tau_C = 5\,\text{ps}$, a higher total memory capacity and with higher nonlinear memory capacities. Therefore, we expect that the compound architecture with aptly chosen values of τ_C and N_t results in a good computational performance for complex tasks where a higher order nonlinearity is required.

Future work aimed at investigating the topology of the compound architecture could be useful. For example, one could compare the effects of integrating multiple external delay lines coupled with different nodes in the architecture, or distributing the feedback signal to multiple nodes instead of just one. Furthermore, it would be interesting to investigate the role of the external delay time τ_D. These extensions of our current work are however outside of the scope of this paper.

VII. Conclusion

In this paper, we numerically investigate a photonic RC system consisting of a spatially and temporally distributed architecture. We show that in this combined network, the spatially distributed architecture is able to function as a passive mask generator to the temporally distributed architecture. This is useful because temporally distributed RC typically requires a complex masking, i.e. time-multiplexing, procedure of the input data which decreases the computing bandwidth of the reservoir computing system and often requires fast equipment to generate the mask. These findings motivated us to investigate a new RC system by efficiently combining these spatially and temporally distributed architecture: the compound RC system. This compound RC system can make use of temporally distributed nodes, originating from temporally distributed DB-RC, and is constructed by combining a passive spatially distributed architecture with a semiconductor laser in an external delay line. We found that it mitigates the requirement of needing a masking procedure, potentially increasing its bandwidth compared to traditional temporally distributed RC architectures due to not having to time-multiplex the input data with the preprocessing mask. Furthermore, by using compound reservoir computing with additional temporally distributed nodes - which is also used in temporally distributed DB-RC - we are able to easily expand the number of nodes by combining the physical nodes present in spatially distributed systems and the virtual nodes from temporally distributed systems.

We show that we are able to outperform the computational performance of the temporally distributed DB-RC with the compound architecture on the Santa Fe time-series prediction task, with the same number of nodes, and at a much higher computing bandwidth. This increased performance is valid for a large range of internal time delays of the compound architecture, indicating that precise tuning of this time delay is not needed for this benchmark task. However, we also found that the time delay between the physical nodes in the compound architecture plays an important role in the performance for accessing the higher order nonlinearity within the architecture. This is reflected by the large nonlinear computational memory capacities at small internal time delays. Our study also shows that temporally distributed sampling in the employed spatially distributed RC system is a simple yet effective tool to increase the number of nodes. When compared to passive spatially distributed RC, the nonlinear memory capacity is significantly enhanced for the compound architecture when we apply additional temporal nodes. Although we used specific reservoir systems in our study, we think that the main conclusions will also be applicable for other reservoir implementations.

Acknowledgments

This research was funded by the Research Foundation Flanders (FWO) under grants G028618N, G029519N and G006020N. Additional funding was provided by the EOS project "Photonic Ising Machines". This project (EOS number 40007536) has received funding from the FWO and F.R.S.-FNRS under the Excellence of Science (EOS) programme.

References

[1] G. E. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[2] M. M. Waldrop, "The chips are down for Moore's law," *Nature News*, vol. 530, no. 7589, p. 144, 2016.

[3] T. N. Theis and H.-S. P. Wong, "The end of Moore's law: A new beginning for information technology," *Computing in science & engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[4] D. Mamaluy and X. Gao, "The fundamental downscaling limit of field effect transistors," *Applied Physics Letters*, vol. 106, no. 19, 2015.

[5] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[6] T. F. De Lima, B. J. Shastri, A. N. Tait, M. A. Nahmias, and P. R. Prucnal, "Progress in neuromorphic photonics," *Nanophotonics*, vol. 6, no. 3, pp. 577–599, 2017.

[7] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature communications*, vol. 5, no. 1, p. 3541, 2014.

[8] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific reports*, vol. 2, no. 1, pp. 1–6, 2012.

[9] S. Sackesyn, C. Ma, A. Katumba, J. Dambre, and P. Bienstman, "A power-efficient architecture for on-chip reservoir computing," in *Artificial Neural Networks and Machine Learning–ICANN 2019: Workshop and Special Sessions: 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17–19, 2019, Proceedings 28*. Springer, 2019, pp. 161–164.

[10] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[11] M. R. Salehi, E. Abiri, and L. Dehyadegari, "An analytical approach to photonic reservoir computing–a network of SOA's–for noisy speech recognition," *Optics Communications*, vol. 306, pp. 135–139, 2013.

[12] D. Verstraeten, B. Schrauwen, and D. Stroobandt, "Reservoir-based techniques for speech recognition," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 1050–1053.

[13] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[14] E. S. Skibinsky-Gitlin, M. L. Alomar, E. Isern, M. Roca, V. Canals, and J. L. Rossello, "Reservoir computing hardware for time series forecasting," in *2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. IEEE, 2018, pp. 133–139.

[15] D. Canaday, A. Griffith, and D. J. Gauthier, "Rapid time series prediction with a hardware-based reservoir computer," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 12, p. 123119, 2018.

[16] G. Donati, C. R. Mirasso, M. Mancinelli, L. Pavesi, and A. Argyris, "Microring resonators with external optical feedback for time delay reservoir computing," *Optics Express*, vol. 30, no. 1, pp. 522–537, 2022.

[17] R. M. Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van der Sande, "Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics," *Optics express*, vol. 22, no. 7, pp. 8672–8686, 2014.

[18] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, "Parallel photonic information processing at gigabyte per second data rates using transient states," *Nature communications*, vol. 4, no. 1, pp. 1–7, 2013.

[19] J. Bueno, D. Brunner, M. C. Soriano, and I. Fischer, "Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback," *Optics express*, vol. 25, no. 3, pp. 2401–2412, 2017.

[20] A. Argyris, J. Bueno, and I. Fischer, "Photonic machine learning implementation for signal recovery in optical communications," *Scientific reports*, vol. 8, no. 1, p. 8487, 2018.

[21] Y. Kuriki, J. Nakayama, K. Takano, and A. Uchida, "Impact of input mask signals on delay-based photonic reservoir computing with semiconductor lasers," *Optics express*, vol. 26, no. 5, pp. 5777–5788, 2018.

[22] Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar, "High-performance photonic reservoir computer based on a coherently driven passive cavity," *Optica*, vol. 2, no. 5, pp. 438–446, 2015.

[23] J. Nakayama, K. Kanno, and A. Uchida, "Laser dynamical reservoir computing with consistency: an approach of a chaos mask signal," *Optics express*, vol. 24, no. 8, pp. 8679–8692, 2016.

[24] B. Schneider, J. Dambre, and P. Bienstman, "Using digital masks to enhance the bandwidth tolerance and improve the performance of on-chip reservoir computing systems," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 12, pp. 2748–2753, 2015.

[25] S. Ortín, M. C. Soriano, L. Pesquera, D. Brunner, D. San-Martín, I. Fischer, C. Mirasso, and J. Gutiérrez, "A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron," *Scientific reports*, vol. 5, no. 1, p. 14945, 2015.

[26] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nature communications*, vol. 2, no. 1, pp. 1–6, 2011.

[27] R. M. Nguimdo, G. Verschaffelt, J. Danckaert, and G. Van der Sande, "Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics," *Optics express*, vol. 22, no. 7, pp. 8672–8686, 2014.

[28] K. Vandoorne, W. Dierckx, B. Schrauwen, D. Verstraeten, R. Baets, P. Bienstman, and J. Van Campenhout, "Toward optical signal processing using photonic reservoir computing," *Optics express*, vol. 16, no. 15, pp. 11 182–11 192, 2008.

[29] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE transactions on neural networks*, vol. 22, no. 9, pp. 1469–1481, 2011.

[30] M. R. Salehi and L. Dehyadegari, "Optical signal processing using photonic reservoir computing," *Journal of Modern Optics*, vol. 61, no. 17, pp. 1442–1451, 2014.

[31] M. C. Soriano, S. Ortín, L. Keuninckx, L. Appeltant, J. Danckaert, L. Pesquera, and G. Van der Sande, "Delay-based reservoir computing: noise effects in a combined analog and digital implementation," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 2, pp. 388–393, 2014.

[32] H. Toutounji, J. Schumacher, and G. Pipa, "Optimized temporal multiplexing for reservoir computing with a single delay-coupled node," in *The 2012 International Symposium on Nonlinear Theory and its Applications (NOLTA 2012)*, 2012.

[33] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Physical Review X*, vol. 7, no. 1, p. 011015, 2017.

[34] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing," *Optics express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[35] M. C. Soriano, J. García-Ojalvo, C. R. Mirasso, and I. Fischer, "Complex photonics: Dynamics and applications of delay-coupled semiconductors lasers," *Reviews of Modern Physics*, vol. 85, no. 1, p. 421, 2013.

[36] K. Harkhoe, G. Verschaffelt, A. Katumba, P. Bienstman, and G. Van der Sande, "Demonstrating delay-based reservoir computing using a compact photonic integrated chip," *Optics express*, vol. 28, no. 3, pp. 3086–3096, 2020.

[37] K. Harkhoe and G. Van der Sande, "Delay-based reservoir computing using multimode semiconductor lasers: Exploiting the rich carrier dynamics," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 25, no. 6, pp. 1–9, 2019.

[38] ——, "Task-independent computational abilities of semiconductor lasers with delayed optical feedback for reservoir computing," in *Photonics*, vol. 6, no. 4. Multidisciplinary Digital Publishing Institute, 2019, p. 124.

[39] I. Bauwens, K. Harkhoe, P. Bienstman, G. Verschaffelt, and G. Van der Sande, "Influence of the input signal's phase modulation on the performance of optical delay-based reservoir computing using semiconductor lasers," *Optics Express*, vol. 30, no. 8, pp. 13 434–13 446, 2022.

[40] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific reports*, vol. 2, no. 1, pp. 1–6, 2012.

[41] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

[42] F. Duport, A. Smerieri, A. Akrout, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," *Scientific reports*, vol. 6, no. 1, p. 22381, 2016.

[43] F. Stelzer, A. Röhm, K. Lüdge, and S. Yanchuk, "Performance boost of time-delay reservoir computing by non-resonant clock cycle," *Neural Networks*, vol. 124, pp. 158–169, 2020.

[44] S. Ortín and L. Pesquera, "Tackling the trade-off between information processing capacity and rate in delay-based reservoir computers," *Frontiers in Physics*, vol. 7, p. 210, 2019.

[45] T. Hülser, F. Köster, L. Jaurigue, and K. Lüdge, "Role of delay-times in delay-based photonic reservoir computing," *Optical Materials Express*, vol. 12, no. 3, pp. 1214–1231, 2022.

[46] F. Laporte, J. Dambre, and P. Bienstman, "Highly parallel simulation and optimization of photonic circuits in time and frequency domain based on the deep-learning framework pytorch," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.

[47] I. Bauwens, K. Harkhoe, E. Gooskens, P. Bienstman, G. Verschaffelt, and G. Van der Sande, "Combining a passive spatial photonic reservoir computer with a semiconductor laser increases its nonlinear computational capacity," *Optics Express*, vol. 32, no. 14, pp. 24 328–24 345, 2024.

[48] A. Weigend and N. Gershenfeld, "The Santa Fe time series competition data," 1991.

[49] M. C. Soriano, S. Ortín, D. Brunner, L. Larger, C. R. Mirasso, I. Fischer, and L. Pesquera, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," *Optics express*, vol. 21, no. 1, pp. 12–20, 2013.

[50] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE transactions on neural networks*, vol. 22, no. 1, pp. 131–144, 2010.

[51] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, no. 1, pp. 1–7, 2012.

[52] T. Hülser, F. Köster, K. Lüdge, and L. Jaurigue, "Deriving task specific performance from the information processing capacity of a reservoir computer," *Nanophotonics*, vol. 12, no. 5, pp. 937–947, 2022.

VIII. Biography Section

Ian Bauwens Ian Bauwens was born in Belgium, in 1997. He received a M.Sc. degree in physics and astronomy from Ghent University, Ghent, Belgium, in 2020. He is currently pursuing a Ph.D. degree at the Applied Physics Group at Vrije Universiteit Brussel, in collaboration with the Photonics Research Group at UGent-IMEC of Ghent University. His current research focuses on various topologies of photonic reservoir computing. His research interests include neuromorphic hardware, machine learning, deep learning, and recurrent neural networks.

Peter Bienstman Peter Bienstman was born in Ghent, Belgium, in 1974. He received a degree in electrical engineering from Ghent University, Belgium, in 1997 and a Ph.D. from the same university in 2001, at the Department of Information Technology (INTEC), where he is currently a full professor. His research interests include several applications of nanophotonics (biosensors, photonic information processing, ...) as well as nanophotonics modelling. He has published over 110 papers and holds several patents. He has been awarded an ERC starting grant for the Naresco-project: Novel paradigms for massively parallel nanophotonic information processing.

Guy Verschaffelt Guy Verschaffelt was born in Belgium in 1973. He received the M.E. degree in photonics from the Vrije Universiteit Brussel (VUB) in 1996 and the Ph.D. degree from the same university in 2000. He is currently professor at the Applied Physics research group of the VUB. His research is focused on emission properties of (broad-area) semiconductor lasers, speckle and speckle reduction techniques, the study of non-linear dynamics in photonic systems, photonic implementations of reservoir computing and Ising machines. He is co-author of more than 80 journal papers and 70 conference papers, and teaches to bachelor and master students in Science and Engineering.

Guy Van der Sande Guy Van der Sande was born in Belgium in 1978. He received the Master's degree in electrotechnical engineering with a major in photonics from the Vrije Universiteit Brussel (VUB), Brussels, Belgium, in 2001. He received the title of Doctor in the applied sciences from the Department of Applied Physics and Photonics, VUB, in 2005. His Ph.D. program was awarded with the prize "Ignace Vanderschueren" for the best PhD thesis in Basic, Natural and Applied Science at the Vrije Universiteit Brussel. In 2006, he was postdoctoral researcher at Optique Nonlinéaire Théorique (ULB) with prof. Thomas Erneux, where he did research on nonlinear dynamics vertical-cavity surface-emitting lasers, metamaterials and dynamics of spatially extended nonlinear optical systems. In 2007, he was a visiting scientist at the Institute for Cross-Disciplinary Physics and Complex Systems (IFISC) in the Universitat de les Illes Balears, Palma de Mallorca, Spain (under supervision of Prof. Dr. Claudio R. Mirasso and invited by Prof. Maxi San Miguel), where he worked on nonlinear dynamics of large networks of delay-coupled nonlinear oscillators. In 2013, he was granted a Research Professor fellowship at the Vrije Universiteit Brussel. Since 2012, he provides Physics and Photonics education to bachelor and master students in Sciences and Engineering. His current research interests include modeling and nonlinear dynamics of semiconductor lasers, synchronization phenomena and bio-inspired information processing.