Research Article | Vol. 32, No. 14 / 1 Jul 2024 / Optics Express 24328
Optics EXPRESS

Combining a passive spatial photonic reservoir computer with a semiconductor laser increases its nonlinear computational capacity

IAN BAUWENS,[1,2,*] KRISHAN HARKHOE,[1] EMMANUEL GOOSKENS,[2] PETER BIENSTMAN,[2] GUY VERSCHAFFELT,[1] AND GUY VAN DER SANDE[1]

[1]Applied Physics Research Group, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
[2]Photonics Research Group, Department of Information Technology, Ghent University-IMEC, Technologiepark Zwijnaarde 126, 9052 Ghent, Belgium
[*]ian.bauwens@vub.be

Abstract: Photonic reservoir computing has been used to efficiently solve difficult and time-consuming problems. The physical implementations of such reservoirs offer low power consumption and fast processing speed due to their photonic nature. In this paper, we investigate the computational capacity of a passive spatially distributed reservoir computing system. It consists of a network of waveguides connected via optical splitters and combiners. A limitation of its reservoir is that it is fully linear and that the nonlinearity – which is often required for solving computing tasks – is only introduced in the output layer. To address this issue, we investigate the incorporation of an additional active nonlinear component into the system. Our approach involves the integration of a single semiconductor laser in an external optical delay line within the architecture. Based on numerical simulations, we show that the architecture with this semiconductor laser has a nonlinear computational capacity that is significantly increased as compared to the original passive architecture, which can be beneficial to solving difficult computational tasks.

1. Introduction

Currently, there is a large and growing interest in machine learning (ML) and neuromorphic computing in our society. This was recently demonstrated by the large success of ChatGPT, an artificial intelligence large language model developed by OpenAI [1]. The training of such artificial neural networks is typically performed by digital computers. However, most machine learning models contain many parameters which need to be optimized. ChatGPT, for example, contains billions of trainable parameters, and its successor GPT-4 contains even more [2]. This means that training such models is computationally difficult and requires a lot of memory, time and energy. As a result, many different platforms are being researched as potential alternatives to the current standard of digital machine learning. Photonic implementations of artificial neural networks can provide such an alternative. This is motivated by their potential for increased computing speed, high power efficiency and the possibility of high parallelism inherent to photonics [3–7].

An example of such a photonic neuromorphic framework is photonic reservoir computing (RC). Reservoir computing systems are based on recurrent neural networks (RNNs) and consist of three different layers: an input layer, a reservoir and an output layer (also called the readout layer). The input layer is used to inject data into the system and the output layer is where predictions are made using the input data. The reservoir is a network layer that contains a large number of randomly interconnected nodes which functions as a dynamical nonlinear system. The connection weights of the reservoir are fixed and remain unaltered. The only trainable

#518654 https://doi.org/10.1364/OE.518654
Journal © 2024 Received 12 Jan 2024; revised 22 Mar 2024; accepted 2 May 2024; published 20 Jun 2024

Optics EXPRESS

weights are in the output layer, and therefore the total training time of RC is significantly reduced compared to traditional RNNs [8]. A large variety of (photonic) RC systems have already been successfully used to solve complex tasks, such as speech recognition and pattern recognition [7,9–11], time-series predictions [12–14] and nonlinear channel equalization tasks [15,16]. Various integrated implementations using photonics have already been experimentally realized, such as delay-based RC using a semiconductor with optical feedback [17] and spatially distributed RC using passive silicon photonics [8,18].

We focus on the computational capacities of a spatially distributed photonic network, which can also be implemented using an integrated approach. Its nodes are physically implemented using passive optical components. In this work, we use the passive spatially distributed network introduced in Ref. [16], referred to as the four-port architecture (FPA), and which we describe in detail in Section 2.1. Although the reservoir of this network is entirely passive, some nonlinearity is introduced in this RC system through the input layer (e.g. from the nonlinear transfer function of a Mach-Zehnder modulator) and/or through the readout layer (e.g. because the readout signal is measured using photodetectors). These networks have resulted in good performance on a range of tasks [8,16,19–21]. However, the nonlinearities present in such networks can, for certain difficult tasks, be too weak and result in a sub-optimal performance [22].

There have already been various attempts to introduce nonlinearities in this reservoir. In one of the early works of spatially distributed on-chip silicon photonic networks by Vandoorne et al. [19], the nonlinearities of the nodes were implemented by introducing semiconductor optical amplifiers (SOAs) in the network to replicate a hyperbolic tangent activation function. Although simulations of such networks resulted in good performance on benchmark tasks [20,21], such an approach resulted in a large power consumption and heat production as each node contained an active SOA.

We therefore want to investigate the effect of placing a single nonlinear component inside the reservoir. We propose to add extra nonlinearity to this network by introducing a single semiconductor laser (SL). This differs to previous attempts to introduce nonlinearity within the passive architecture, where SOAs were placed at every node [19–21]. In this study we use an SL as active component, as opposed to a SOA. This choice is motivated by the fact that SLs have a highly nonlinear and fast dynamical response, which results in a much faster operation as compared to SOAs (which have response times on the order of nanoseconds [23]). Furthermore, using SOAs usually implies that a spectral bandpass filter has to be incorporated in the setup to filter out spontaneous emission noise. This makes semiconductor lasers a great candidate for introducing nonlinearity to the FPA. Additionally, this active component can be helpful to provide extra power to counter the losses which occur within this network due to its topology.

The paper is organised as follows: in Section 2, we give a short introduction to the FPA. This architecture is then modified by introducing an SL in an external optical delay line, which we refer to as the FPA+SL. In Section 3, we investigate - based on numerical simulations - the performance of the FPA+SL on different benchmark tasks, together with the performance of the original passive FPA. We also investigate the linear and nonlinear memory capacity of these RC systems. In Section 4, we give the conclusions of this paper.

2. Numerical implementation of the investigated RC systems

2.1. Four-port architecture

In this work, we focus on the passive reservoir computing architecture first introduced in Ref. [8] and improved later on in Ref. [16], which is referred to as the four-port architecture (FPA). It is a linear photonic network which functions as a multipath interferometer and consists of 16 nodes, as shown in Fig. 1. These nodes are arranged in a 4×4 grid and connected by multiple waveguides. Each node consists of a multi-mode interference (MMI) coupler. The blue circles represent the nodes of the architecture (i.e. the MMI couplers) which split their input signals

Optics EXPRESS

equally into three output ports. Every node receives input from two neighbouring nodes and feeds its signal to two other neighbouring nodes. The two remaining ports (one input and one output port) of the nodes are used for data injection and detection and are indicated in Fig. 1 by "input ports" on the left-hand side and "output ports" on the right-hand side of the architecture for every corresponding node. Note that the numbering of the nodes starts at 0 in this work. The optical power at the computing nodes, $node_i$, is measured at all the output ports using photodetectors and is used for calculating weights w_i, which we discuss in more detail further in this paper.

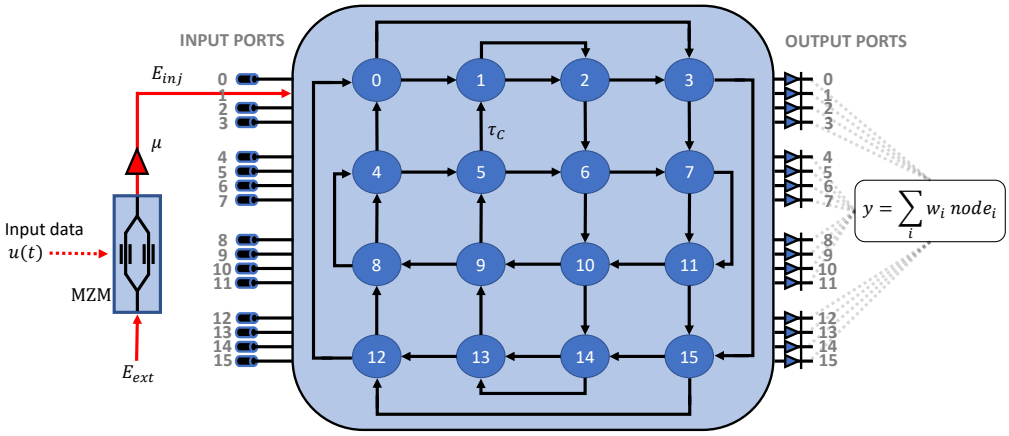


Fig. 1. Illustration of the FPA consisting of 16 3×3 multi-mode interference (MMI) couplers, shown as blue circles, which represent the nodes and which are arranged in a 4×4 grid. Each node is connected to 4 ports of neighbouring nodes and contains an input and output port, used for data injection and detection. The input data $u(t)$ is optically modulated and injected into the input port by a Mach-Zehnder modulator (MZM) at an injection rate μ. The optical power at the output ports of the 16 nodes ($node_i$) is measured using photodetectors in the readout layer and is used to train linear weights (w_i) corresponding to a target output (y). The internal delay time between nodes is represented by τ_C. In our simulations, port 1 is always used as input port.

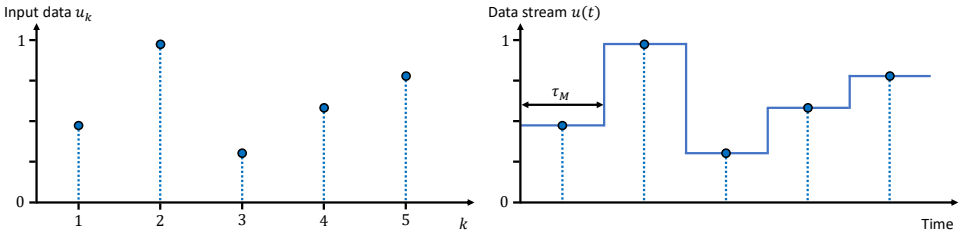
To numerically implement the FPA, we use the PhotonTorch module to code it as a photonic element. PhotonTorch [24] is a Python model which allows for highly parallel numerical simulations of large photonic circuits on graphical processing units (GPUs) and permits easy access to individual component parameters for optimization. The parameters we use for simulating the FPA are given in Table 1. The most important parameter for this architecture is the length of the waveguide between two MMI couplers, which causes an internal delay time τ_C between nodes in the architecture and a phase due to propagation. In practice, the internal delay time τ_C can be controlled accurately in an integrated setup, while the phase can fluctuate significantly due to fabrication imperfections leading to geometry variations within the waveguide, and therefore also to variations in the effective index of refraction. This is because only 50nm of optical path length difference can result in a phase difference of 0.1π, while an optical path length of 50μm is needed for a significant time delay of approximately 1 ps [8,25]. In our simulations, we define the waveguide length between nodes as $\frac{c\tau_C}{n_g}$, with n_g the group index of refraction and c the speed of light. The internal delay time τ_C is a parameter we vary, between 2.5 ps and 60 ps. The phase due to propagation between the nodes is modelled as a random variable drawn from the uniform distribution $[0, 2\pi]$ for every waveguide, and is fixed at the initialization of the simulations. Note that in our simulations no extra delay is modelled for the MMI couplers, as the effect of this would result in a general delay for all MMI couplers which could be incorporated in the internal delay time τ_C, as all MMI couplers are assumed to be identical.

Optics EXPRESS

Table 1. Parameters and their values used in the simulations (unless stated otherwise).

Parameter	Symbol	Standard value
Simulation timestep	dt	0.5 ps
Group refractive index	n_g	4.564
Waveguide losses		1.50 dB/cm
Wavelength	λ	1550 nm
Injection rate	μ	100 s^{-1} or $100\sqrt{10}$ s^{-1}
Internal delay time between couplers	τ_C	Scanned over [2.5,60] ps
Input segment duration	τ_M	30 ps
External delay time (to SL)	τ_D	5 ps
Feedback rate (to SL)	η	$10\sqrt{2}$ns^{-1}
Linewidth enhancement factor	α	3
Threshold gain	g	1 ps^{-1}
Differential gain	ξ	$5\times10^{-9}ps^{-1}$
Spontaneous emission noise factor	β	10^{-6}
Carrier lifetime	τ_e	1 ns
Normalized threshold pump current rate	J_{thr}	$10^{17}s^{-}1$
Normalized excess pump current rate (for FPA+SL)	ΔJ	$2J_{thr}$
Modulation amplitude of MZM	A_{MZM}	$\frac{\pi}{2}$
Bias of MZM	Φ_{MZM}	$\frac{\pi}{4}$

The injection of discrete input data samples u_k is done by a sample-and-hold procedure resulting in an input data stream $u(t)$, where this input data stream is piecewise constant and where every segment of $u(t)$ is stretched with a constant input segment duration τ_M. An illustration of this procedure is shown in Fig. 2. The ratio τ_C/τ_M is an important parameter for the FPA.



(a) (b)

Fig. 2. Illustration of discrete input data samples u_k (a) and the input data stream $u(t)$ consisting of piecewise constant segments, with input segment duration τ_M, resulting from the sample-and-hold procedure (b).

If $\tau_C/\tau_M > 1$, the duration of a segment of the input data stream is shorter than the time required for the sample to travel between two adjacent nodes. This means that a segment of the input data stream has not reached these nodes before the next sampling event, where we measure the optical power of the nodes at the readout layer and which occurs every τ_M. If $\tau_C/\tau_M < 1$, the segment of the input data stream has reached at least one adjacent node before the next sampling event occurs. By making this ratio lower, the segment of the input data stream will be able to reach more nodes within the network before being sampled. However, if τ_C/τ_M is too small, the segment of the input data stream will traverse the network multiple times before the next

Optics EXPRESS

sampling event occurs. This ultimately leads to an almost constant output power from each node as the optical power is spread over the whole network, resulting in a poor performance as the nodes' states are now (almost) independent of the input data.

In Ref. [8], it was found that a ratio of $\tau_C/\tau_M = 0.4$ was optimal for the 2-bit XOR task for bit rates between 0.125 and 12.5 Gbs^{-1} for a spatially distributed RC system similar to the one used here. The injection of the input data stream $u(t)$ is done using one input port, as shown in Fig. 1. In this work, we have arbitrarily chosen input port 1, while the other input ports are not being used. Because of the symmetry in the FPA, we do not expect this choice of input port to be crucial. This is supported by some tests we have done using some other input ports, showing similar results. We inject the data optically by using an unbalanced Mach-Zehnder modulator (MZM), where the output of the MZM is given by

$$E_{MZM} = E_{ext}\left(1 + e^{i(A_{MZM}u(t)+\Phi_{MZM})}\right)e^{i\omega_0 t}. \tag{1}$$

$E_{ext} = 100$ is the amplitude of the injected external field, ω_0 is the frequency of this 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 set at $A_{MZM} = \pi/2$, and bias of the MZM is set at $\Phi = \pi/4$. The resulting electric field E_{MZM} is subsequently injected into the network via a 2×1 MMI coupler, at an injection rate μ, so that the injected signal $E_{inj} = \mu E_{MZM}$. Note that the input is modulated on both the phase and amplitude of the injected signal.

The readout layer is constructed by measuring the optical power of every output port of the FPA at every sampling event (i.e. every τ_M), where we use the output of all 16 nodes, indicated by $node_i$ in Fig. 1. These node states are used for calculating linear weights, w_i, during training on labeled data, and for testing on unseen data.

2.2. Four-port architecture with a semiconductor laser in the external delay line

We now combine the previously mentioned FPA [16] with a single-mode SL in an external optical delay line. This network is shown in Fig. 3 and we refer to this architecture as the FPA+SL. The injection of data into this architecture is identical as the procedure described in Section 2.1, i.e. using the sample-and-hold procedure. The standard input port is again arbitrary chosen to be port 1. All the other input ports of the architecture are again not being used. For the output ports, one output port of the architecture is connected with an external optical delay line. This results in the output of one node not being available for the readout layer, which we refer to as the feedback port. All other 15 remaining nodes are connected to photodetectors. The signals from these photodetectors, indicated by $node_i$ in Fig. 3, are used to form the readout layer and are used for calculating linear weights, w_i, during training and for testing on unseen data. The optical signal from the feedback port is injected into a single-mode SL using an external delay line. The field emitted by the SL is then coupled back to the 2×1 MMI coupler, that in its turn is connected to input port 1 of the FPA, thus concluding the feedback loop. The value for the external delay time τ_D of the delay line from the FPA to the laser is variable in this paper, with values scanned between 5 ps and 60 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 set $\tau_M = 30$ ps, unless stated otherwise, as this timescale results in the highest memory capacity [26] for delay-based reservoir computing when using an SL [27].

The rate equations of the SL [28] for the complex amplitude of the slowly-varying envelope of the electric field $E(t)$ around a center frequency and the number of carriers in the active medium available for lasing $N(t)$ (both dimensionless quantities), are given in Eqs. (2,3).

$$\frac{dE(t)}{dt} = \frac{1}{2}(1 + i\alpha)\xi N(t)\,E(t) + \eta E_{FB}(t) + \tilde{F}_\beta(t) \tag{2}$$

Optics EXPRESS

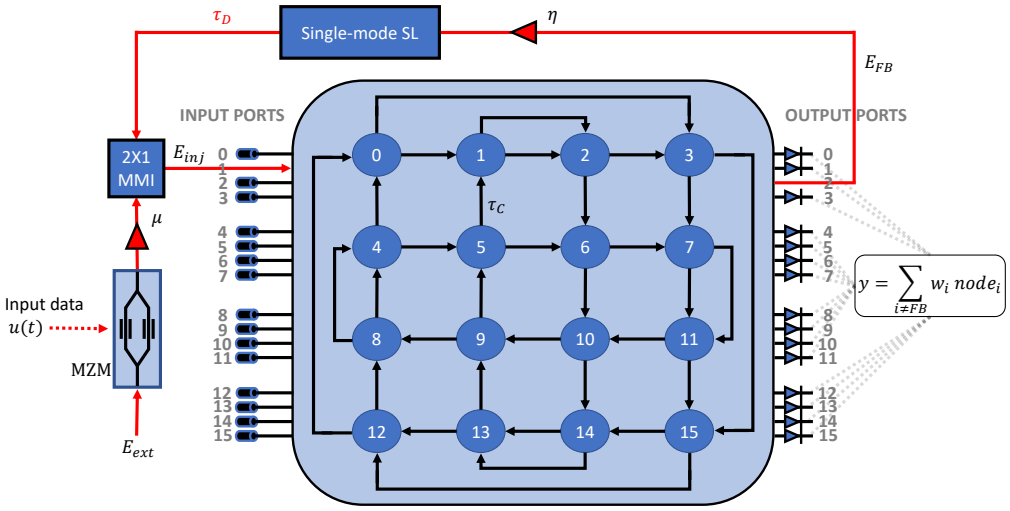


Fig. 3. Illustration of the FPA+SL. The input data $u(t)$ is optically modulated and injected into the input port by a Mach-Zehnder modulator (MZM) at an injection rate μ and a 2×1 multi-mode interference (MMI) coupler. In our simulations, we use input port 1 to inject data and use one of the output ports as feedback (FB) port. This feedback 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 output ports of the other 15 nodes ($node_i$) is measured using photodetectors in the readout layer and is used to train linear weights (w_i) corresponding to a target output (y). The internal delay time between nodes is represented by τ_C. In our simulations, port 1 is always used as input port, while the feedback port is variable.

$$\frac{dN(t)}{dt} = \Delta J - \frac{N(t)}{\tau_e} - (g + \xi N(t))\,|E(t)|^2\,, \tag{3}$$

where α is the linewidth enhancement factor, ξ and g are the differential gain and linear threshold gain. η represents the fixed feedback rate. ΔJ is the excess pump current divided by the elementary charge, so that $\Delta J = J - J_{thr}$, with J the normalized pump current rate, and J_{thr} the normalized threshold pump current rate of the SL. τ_e is defined as the carrier lifetime. $\tilde{F}_\beta(t)$ represents complex Gaussian white noise to simulate the spontaneous emission noise. $\tilde{F}_\beta(t)$ has a zero mean $\langle \tilde{F}_\beta(t) \rangle = 0$ and autocorrelation $\langle \tilde{F}_\beta(t) \tilde{F}_\beta(t')^* \rangle = \frac{\beta N(t)}{\tau_e} \delta(t - t')$, where β controls the spontaneous emission noise strength. The $E_{FB}(t)$ term represents the slowly-varying envelope amplitude of the feedback term, originating from the feedback port of the FPA and which is injected into the single-mode SL. Remark that we use the same frequency ω_0 for the injection laser and for the center frequency of $E(t)$ in Eq. (2) (i.e. the detuning between E_{ext} and $E(t)$ is set to zero).

3. Numerical results

3.1. Santa Fe prediction task

In this section, we compare the performance of the FPA and the FPA+SL on the Santa Fe time-series prediction task. Specifically, the goal of this benchmark task is to make a one-step ahead prediction using the previous input samples. The Santa Fe dataset contains just over 9000 data points sampled from a far-IR laser in a chaotic regime [29]. To train the weights of the readout layer, we take 3010 data samples from the beginning of this time-series. The next 1010 data samples are used to perform the testing. The 10 first data samples of both sets are discarded from the training and testing procedures to remove the effects of any transients when initializing the architectures. As a measure for the performance, we use the normalized mean squared error

Optics EXPRESS

(NMSE) on the test set, where the NMSE is defined as

$$\mathrm{NMSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\langle (\mathbf{y} - \hat{\mathbf{y}})^2 \rangle}{\langle (\mathbf{y} - \langle \mathbf{y} \rangle)^2 \rangle}, \tag{4}$$

with $\mathbf{y}$ the expected output data and $\hat{\mathbf{y}}$ the predicted output data. The best performance corresponds to the lowest NMSE, with reported best NMSE values for simulated RC systems on the Santa Fe one-step ahead prediction task being around 10^{-2} [27,30,31].

In the next two sections, we investigate the influence of the injected power, which is controlled by the injection rate μ, on the performance of both architectures. We consider two different situations. The first situation is where we inject data, coming from the MZM, into the architecture at an injection rate $\mu = 100$ s^{-1} ps, and which we refer to as the weak data injection regime. The second situation is where the injection rate is higher, $\mu = 100\sqrt{10}$ s^{-1}. This corresponds to a 10-fold increase in the injected power compared to the previous situation. We refer to this as the strong data injection regime.

3.2. Performance on Santa Fe task in weak data injection regime

To find the optimal combination for the internal delay time τ_C between nodes and the external delay time τ_D of the network, we perform a two dimensional scan of these parameters and investigate their effect on the performance of the Santa Fe benchmark task. As mentioned previously, we have fixed the input segment duration to $\tau_M = 30$ ps. As we initially do not know whether or not the choice of the feedback port would have a large impact on the results, we perform simulations using subsequently a different output port as feedback port. The result of the two dimensional sweep of τ_C and τ_D is shown in Fig. 4 for the FPA+SL in the weak data injection regime for the 16 different choices of feedback port configuration.

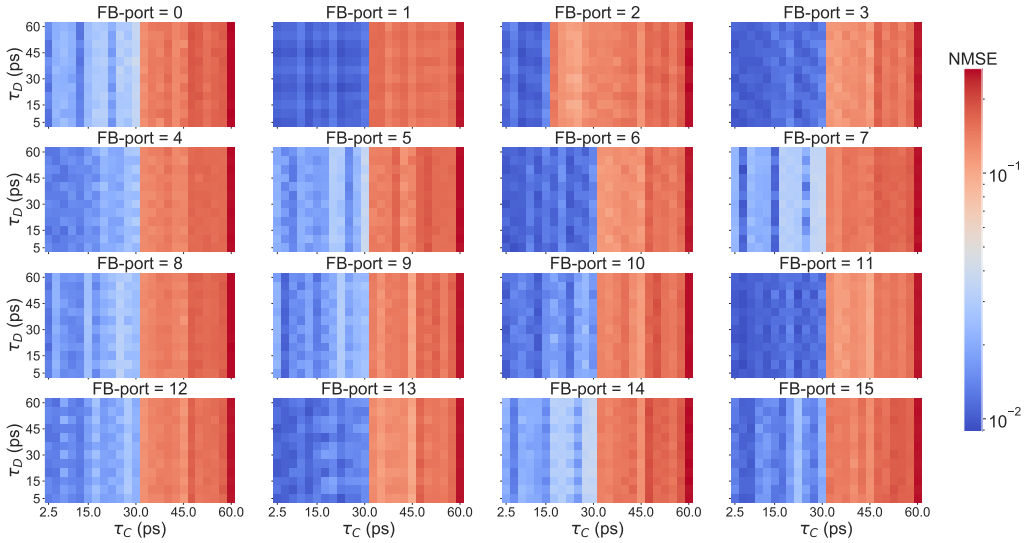


Fig. 4. Performance of the FPA+SL in the external delay line on the Santa Fe task for a two dimensional sweep of τ_C and τ_D, for different feedback port configurations shown as subplots. The input segment duration $\tau_M = 30$ ps is held constant for all subplots and the injection rate $\mu = 100$ s^{-1}.

In Fig. 4, we observe a distinct region of good performance which depends on τ_C. For all feedback ports, except feedback port 2, we find that this region of good performance corresponds

Optics EXPRESS

to $\tau_C \in [2.5, 30]$ ps (remark that our scan only starts at $\tau_C = 2.5$ps). This shows that when considering this architecture, the internal delay time τ_C should not surpass the input segment duration $\tau_M = 30$ ps to avoid a poor performance, which was already known from previous studies [26]. When the internal delay time τ_C is longer than the input segment duration 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. Therefore, when we inject input sample n of the Santa Fe time-series, the readout layer only contains input samples $n-1$ or older while the benchmark task still remains the same, i.e. predicting input sample $n+1$. Thus, the reservoir is now implicitly performing a multi-step ahead prediction, which is more difficult than the one-step ahead prediction. Note also that the value of the external delay time τ_D does not seem to be critical to achieve good performance, as the results do not vary with τ_D within the scanned region.

We remark that the performance when using feedback port 2 is different compared to all other feedback ports. Namely, its performance deteriorates at $\tau_C > /\tau_M 2 = 15$ ps, while for all the other feedback ports this is at $\tau_C > \tau_M = 30$ ps. When we use feedback port 2, this output port is not accessible for the readout layer. As we inject data in port 1, we thus now have to wait for $2\tau_C$ before a segment of the input data stream first percolates in the readout layer, and therefore τ_M needs to be at least equal to $2\tau_C$ for good performance.

To compare the performance of the FPA+SL with the FPA, we again fix $\tau_M = 30$ ps and scan the internal delay time τ_C where we repeat the simulation run 10 times with different random phases between nodes for both architectures. For the FPA+SL, we fix the external delay time τ_D to 5 ps. The reason for this specific value for τ_D is because, according to Fig. 4, it results in a good performance and because we want to minimize the external delay as much as possible, to allow for compact future implementations. Figure 5 shows the results of the performance of the FPA and the FPA+SL. The results for both architectures are shown with their median and interquartile range (IQR). The reason for using these metrics instead of the mean and standard deviation is because of their resilience to outliers.

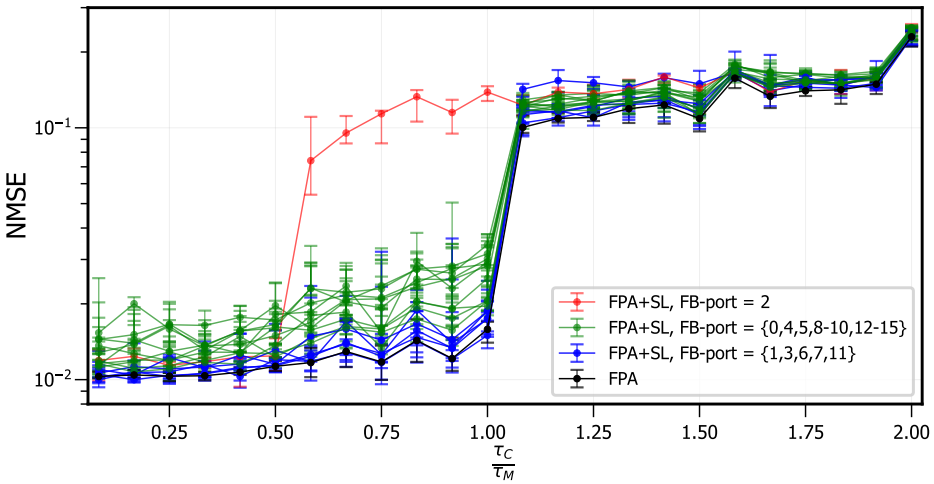


Fig. 5. Median and interquartile range of the NMSE as a function of τ_C / τ_M of the FPA and the FPA+SL, over 10 simulation runs with different phases between nodes. The external delay is held constant at $\tau_D = 5$ ps, the input segment duration at $\tau_M = 30$ ps and an injection rate $\mu = 100$ S^{-1} for both architectures. The performance of the FPA+SL with feedback from port 2 is shown in red and with feedback from the other ports in blue or green.

Optics EXPRESS

The performance of the FPA is shown as a black curve in Fig. 5. The best performance of the FPA is found at $\tau_C/\tau_M \leq 1$. In Fig. 5, we also show the performance of the FPA+SL, for the 16 possible feedback port configurations. Due to the different performances of the feedback port configurations, we classify them into three separate groups based on their performance as a function of τ_C/τ_M. The performance corresponding to feedback port 2 is shown in red, as it is the only configuration which results in a poor performance for τ_C>15 ps for the FPA+SL. The two other groups containing the other feedback port configurations are again split based on their performance, in this case using an arbitrarily chosen threshold NMSE of 2×10^{-2} for $\tau_C \leq \tau_M$. The first group consists of feedback port configurations (ports 0, 4, 5, 8 to 10, 12 to 15) which are considered bad, with median NMSE values higher than 2×10^{-2} for $\tau_C \leq \tau_M$, shown in green. The other group consists of feedback port configurations (ports 1, 3, 6, 7 and 11) which are considered good, with median NMSE values lower than 2×10^{-2} for $\tau_C \leq \tau_M$, shown in blue. We do, however, have to mention that even though the medians of this group correspond to a lower NMSE compared to the first group, their interquartile ranges still overlap.

The fact that we observe different performances for different feedback port configurations can be heuristically explained by the optical power present at the different output ports. This is because of attenuation due to loss within the waveguides and because of the topology of the architecture, where the optical power is distributed throughout the architecture. As we introduce the signal via input port 1, we can calculate the path that the light will travel within the FPA. We can do this by considering the architecture as a directed graph. The shortest distance between the origin, i.e. node 1, and the other nodes can be calculated via Dijkstra's algorithm [32], and the results are shown in Fig. 6(b). The number of occurrences of said shortest distance is also calculated and given in Fig. 6(c). The numbering of the nodes of the architecture are shown in Fig. 6(a) for clarity.

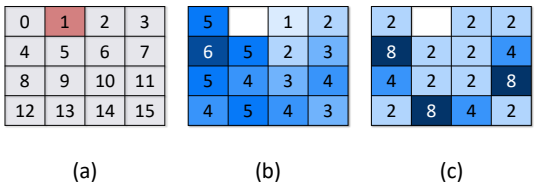


Fig. 6. Numbering of the nodes of the FPA, with the standard choice of the input port as node 1 highlighted in red (a). Shortest path expressed as multiples of τ_C between node 1 and all the other nodes of the FPA (b) and the number of occurrences of this shortest path (c).

Figure 6(b) shows that some nodes are visited within a short distance from the input port at node 1, while for some nodes the shortest distance to the input node 1 is $6\tau_C$. Also, Fig. 6(c) shows that some ports are visited often within this shortest path, due to the topology of the FPA.

The results from Fig. 6(b) explain why some feedback ports such as at the output ports of nodes 3 and 6 result in good performance. This is because the signal only has to travel 2 waveguides from the input port at node 1 to those nodes. Therefore, the input signal only passes through 3 MMI couplers before reaching nodes 3 and 6. The power which arrives at these nodes will therefore be less attenuated due to splitting and waveguide losses when compared to the power in the nodes which are farther away from the input node.

The fact that nodes 3 and 6 receive significantly more optical power than the other nodes implies that the external SL can more easily achieve injection locking when it receives its injection from either of these two nodes. The reason why the feedback port configuration for node 11 shows a good performance, despite its large distance from the input node, is the fact that the shortest path from node 1 to node 11 occurs multiple times in the FPA, as is shown in 6(c): for node 11, the shortest path occurs 8 times, thus still allowing for a high power in this node. The

Optics EXPRESS

same argument holds for node 7, which has a distance of only 3 waveguides from node 1, but which also occurs often. Note that these discussions are based on only the shortest distances. Different paths taken may add extra optical power to the nodes, but these paths will be longer, and thus result in more attenuation. We conclude that some feedback port configurations perform better compared to others on the Santa Fe task due to a higher power available in the feedback signal that is injected in the SL. In the next section, we further investigate the importance of injection locking of the laser on the performance for the FPA+SL. We use stronger data injection into the architecture so that a stronger field is being injected into the single-mode SL. Instead of the previously injection rate $\mu = 100\ \mathrm{s}^{-1}$, we now use $\mu = 100\sqrt{10}\ \mathrm{s}^{-1}$. We expect this to more easily lead to injection locking, and hence we expect improved performance.

3.3. Performance on Santa Fe task in strong data injection regime

In Fig. 7, we repeat the two dimensional parameter scan of τ_C and τ_D, at fixed $\tau_M = 30$ ps, but now for a larger injection rate of $\mu = 100\sqrt{10}\ \mathrm{s}^{-1}$. Compared to Fig. 4, we notice similar results with two distinct regions of good and bad performance. However, we observe that the contrast between the region of good and bad performance is increased, with more points in the good region having a very good performance. The fact that we observe a large region of good performance shows that we do not require to carefully fine-tune the hyperparameters τ_C and τ_D of the FPA+SL for this task. Moreover, as the performance does not change much with τ_C and τ_D, we are able to keep the delay times very small, resulting potentially in a small footprint of the FPA+SL. The timing of the architecture appears to be unchanged, with all the edges between good and bad performance being unchanged compared to Fig. 4 for all the feedback port configurations, i.e. at $\tau_C = \tau_M = 30$ ps for all feedback ports except for port 2.

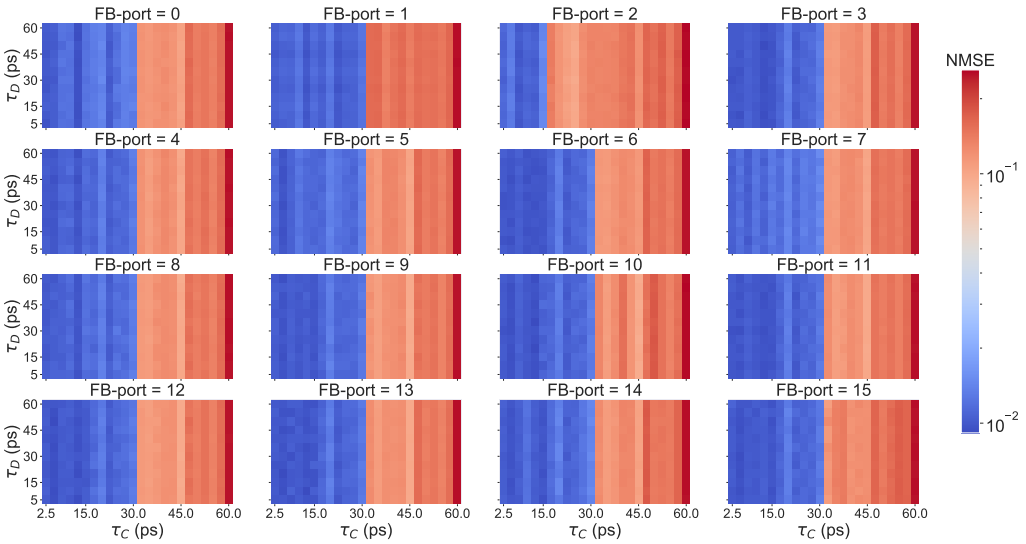


Fig. 7. Performance of the FPA+SL on the Santa Fe task for a two dimensional sweep of τ_C and τ_D, for different feedback port configurations shown as subplots. The input segment duration $\tau_M = 30$ ps is held constant for all subplots and the injection rate $\mu = 100\sqrt{10}\ \mathrm{s}^{-1}$.

We now compare the performance of the FPA+SL, with a stronger injection compared to the previous section, with the FPA. We again scan the internal delay time τ_C and repeat the simulations 10 times with different random phases between nodes, at fixed $\tau_M = 30$ ps. We again fix the external delay time τ_D to 5 ps for the FPA+SL, limiting the length of the external delay line. The results of the simulations, indicated by the median NMSE and the NMSE's interquartile

Optics EXPRESS

range, as a function of τ_C are shown in Fig. 8. We find that the NMSE of the FPA barely differs from the results found in Fig. 5, where a weaker injection was used. However, we now observe that all feedback port configurations for the FPA+SL, shown in blue, have similar performance, as opposed to the weak injection regime. This means that the choice of the feedback port becomes less relevant for high injection strengths. This is most likely due to the stronger injection, so that it is easier to achieve injection locking in the laser for all feedback ports, which allows for a more consistent response as compared to the weak injection regime shown in Fig. 5 [33]. Additionally, all feedback port configurations, apart from port 2, show similar performance to the FPA for all values of τ_C/τ_M. We observe that the performance remains very good for $\tau_C \leq \tau_M$, with an NMSE $\approx 10^{-2}$. Note also that an NMSE of 10^{-2} is already among the best performance values reported in reservoir computing literature on the Santa Fe task, so that we expect that an even lower NMSE is very difficult to achieve. As with Fig. 5, we observe that when $\tau_C > \tau_M$, the performance of the FPA+SL starts to deteriorate drastically for all feedback ports except for port 2. For feedback port 2, shown in red, we again observe a deteriorating NMSE at $\tau_C > \tau_M/2$, as was present in Fig. 5.

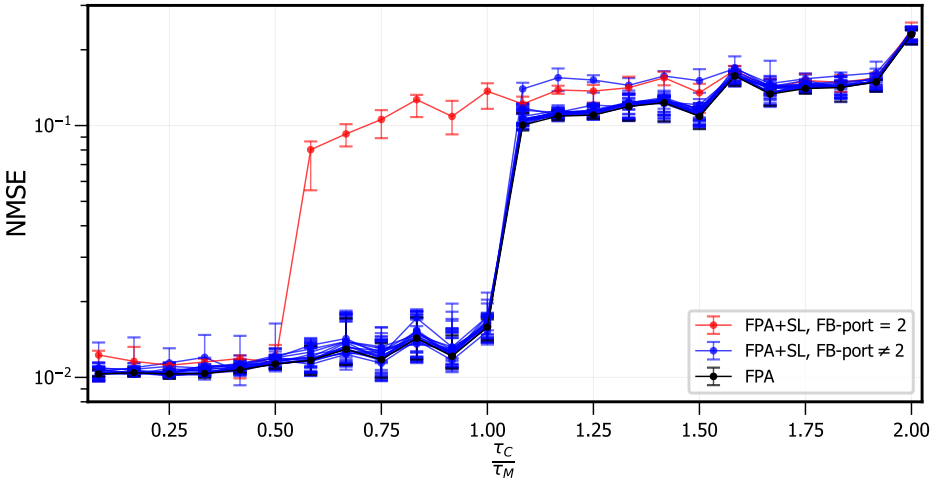


Fig. 8. Median and interquartile range of the NMSE as a function of τ_C/τ_M of the FPA and the FPA+SL, over 10 simulations runs with different phases between nodes. The external delay is held constant at $\tau_D = 5$ ps, the input segment duration at $\tau_M = 30$ ps and an injection rate $\mu = 100\sqrt{10}$ s^{-1} for both architectures. The performance of the FPA+SL with feedback from port 2 is shown in red and with feedback from the other ports in blue.

The similar performance of the FPA and the FPA+SL seems to indicate that for the Santa Fe prediction task, the additional nonlinearity of the SL in the feedback loop does not lead to better performance on this specific task, and that the nonlinearity of the MZM and photodetectors is sufficient. However, with our architecture, we are able to outperform other architectures on the Santa Fe task with a lower error while using fewer nodes. For example, if we compare our architecture to the parallel and deep reservoir computing system [11], we are able to achieve a similar performance, but with fewer nodes (15 versus 800 nodes). Likewise, we are able to achieve a similar performance as the on-chip photonic neural field [18] (15 versus 1300 nodes). Remark that for other tasks, this will likely be different. Furthermore, it is not straightforward to compare these results with other works - e.g. where they have incorporated SOAs instead of SLs as nonlinearities into passive architectures [19–21] - as their benchmarks differ from the Santa Fe task we use as benchmark. Instead of focusing on these other benchmark tasks, we will

Optics EXPRESS

quantify the performance of the FPA and FPA+SL using computational capacities which can be generalized to the performance on any other task.

3.4. Memory capacity of the architectures

In the previous section, we have investigated the performance of the FPA and the FPA+SL on the one-step ahead Santa Fe time-series prediction task. However, the Santa Fe task does not quantify the amount of nonlinearity of both systems. The added nonlinearity from the SL could be beneficial to more complex benchmark tasks, such as e.g. the NARMA10 task. Instead of selecting specific benchmark tasks that might benefit from this added nonlinearity, we have chosen to study a task-independent measure of the computational capacity which can quantify this increased nonlinearity. We quantify - based on numerical simulations - the linear and nonlinear memory capacity of the FPA and the FPA+SL. We scan the internal delay time τ_C and the input segment duration τ_M. We again fix the external delay for the FPA+SL at $\tau_D = 5$ ps, i.e. the same value as in the previous section. As done in the previous sections, we use input port 1 to inject the input data. For the choice of the feedback port, we use output port 1. This allows us to introduce the nonlinearity early in the architecture, and which reduces the losses due to propagation. A different choice for the feedback port can have a different result on the calculated memory capacity, but this is outside the scope of this work.

We calculate the memory capacity using the techniques from Ref. [34]. We inject an input data stream, originating from a sample-and-hold procedure of discrete input samples u_k- drawn from a uniform distribution between $[-1, +1]$. Next, we train the readout layer to reconstruct products consisting of normalized Legendre polynomials of l-delayed previous input samples u_{k-l}. We limit this delayed input to $l \leq 10$. In total, we inject 500 010 input samples from the uniform distribution, and again discard the first 10 samples which are used for initialization of the architecture. To have strong injection, we use the parameter values from Section 3.3.

The target data y_k is constructed by the product of normalized Legendre polynomials $P_\delta(\cdot)$, of a given degree δ, from previous inputs. Note that there are multiple combinations of Legendre polynomials when considering combined polynomial degrees larger than 1. For example, if the combined degree $d = 3$, we need to take into account all the product combinations of Legendre polynomials with degrees: $\delta \in \{(1, 1, 1), (1, 2) \text{ and } (3)\}$. To illustrate this, the combination $\delta = (1, 2)$ for the third degree can be written as a product of Legendre polynomials with degrees 1 and 2:

$$y_{k,(\delta=(1,2),(l_1,l_2,l_3))} = P_1(u_{k-l_1})\, P_2(u_{k-l_2}, u_{k-l_3}), \tag{5}$$

where (l_1, l_2, l_3) are the set of indices of past input samples, and which we have limited to $l_i \leq 10$ here. Therefore, there also exist many different possible combinations for these sets of indices, adding to the complexity when calculating the memory capacity. The mean squared error (MSE) between the expected signal y_k, such as for the example shown in Eq. (5), and the predicted signal $\hat{y}_k$ can be calculated for all input samples. The MSE, for a specific product combination δ and a set of specific indices of past inputs $(l_1, \dots)$, is used for calculating the memory capacity,

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

where $C_{\delta,(l_1,\dots)} \in [0, 1]$ and with the average taken over all input samples. If we then sum over all possible combinations of the delayed elements $(l_1, \dots)$, for a specific combination of Legendre polynomials δ, and sum over all possible Legendre combinations which result in the combined degree of d, we have the memory capacity of degree d,

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

Optics EXPRESS

If we sum over all memory capacities over all degrees, we can find the total computational memory capacity of the system, MC.

The theoretical upper limit for the total memory capacity MC for both RC systems is given by the total number of nodes within the architecture, as proven in Ref. [34]. Note, however, that the total memory capacity does not provide a clear indication for the performance on different benchmark tasks. The individual memory capacities per degree provide a much better indication for this, as shown in Ref. [22]. Note also that we only use a finite number of input data samples, meaning that we could potentially overestimate the value of the memory capacity C_d. To avoid this situation, we implement a threshold capacity $C_{thr} \approx 2 \times 10^{-4}$, so that values for $C_{\delta,(l_1,\dots)}$ below C_{thr} are not considered when calculating the memory capacity, as discussed in Ref. [34].

In Fig. 9(a)-(e), we show the memory capacity for the first five degrees, and the cumulative (total) memory capacity up to the fifth degree for the FPA in Fig. 9(f). In Fig. 9(a), we observe that the linear memory capacity (i.e. degree 1, $C_{d=1}$), is the highest when the internal delay time τ_C is equal to the input segment duration τ_M, i.e. $\tau_C = \tau_M$. For the second degree, $C_{d=2}$, shown in Fig. 9(b), we observe the highest memory capacities when $\tau_C \lessapprox \tau_M$. This can also be seen for the third degree, $C_{d=3}$, shown in Fig. 9(c), although the region of high memory capacity is in that case reduced to $2\tau_C \lessapprox \tau_M$. For the fourth and fifth degrees, $C_{d=4}$ and $C_{d=5}$, shown in Fig. 9(d) and Fig. 9(e), we observe negligible memory capacities. The total memory capacity, shown in Fig. 9(f), shows a nearly uniform distribution, except for $\tau_C > \tau_M$. Note that we only calculate memory capacities up to degree $d = 5$, which means that we might somewhat underestimate the total memory capacity MC for this architecture. However, the memory capacity up to the fifth degree should already provide a good indication for the nonlinearities within the architecture.

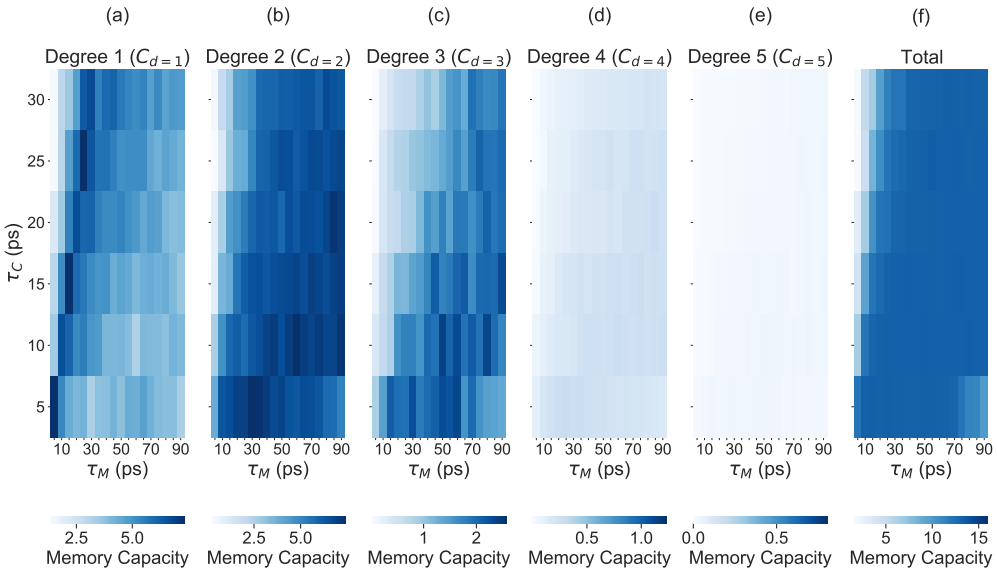


Fig. 9. Memory capacity per degree for the FPA (a)-(e), and the total memory capacity (f). The internal delay time τ_C and input segment duration τ_M are varied. The injection rate is fixed to $\mu = 100\sqrt{10}\ \mathrm{s}^{-1}$.

For the linear memory capacity of the FPA we observe the largest memory capacity when $\tau_C = \tau_M$. One potential reason for this can be as follows. The linear memory capacity of an architecture represents the ability of recalling past inputs that are unmixed with other inputs, as the target data is constructed using Legendre polynomials of degree 1. Reminding the reader we inject data in node 1, when $\tau_C = \tau_M$ the segment of the input data stream has reached its neighbouring node,

Optics EXPRESS

i.e. node 2, when the next sampling event occurs. This means that the readout layer contains information where no mixing with other inputs has taken place. When $\tau_C < \tau_M$, the segment of the input data stream is able to reach more nodes before a sampling event occurs. Therefore, the information available in the readout layer when a sampling event occurs, will be a mixture of various previous inputs which have been propagating longer in the architecture. This has a negative effect on the linear memory recall of a single input.

For the quadratic memory capacity, we find the highest memory capacities when $\tau_C \lessapprox \tau_M$. This could be attributed to the increased mixing of input samples at the nodes which occurs if $\tau_C \leq \tau_M$. At $\tau_M = 6\tau_C$, we now find the highest quadratic memory capacity. We make the observation that $6\tau_C$ is the shortest time it takes for a signal to loop around the architecture and return back to that same node. This is the shortest loop possible within the architecture, and thus with the lowest loss, and can be encountered often within the architecture. For example, an original input injected in node 1 loops around the FPA following the shortest loop and mixes with another input back at node 1 after $6\tau_C$. This path, corresponding to traveling 6 waveguides, is illustrated in green in Fig. 10(a).

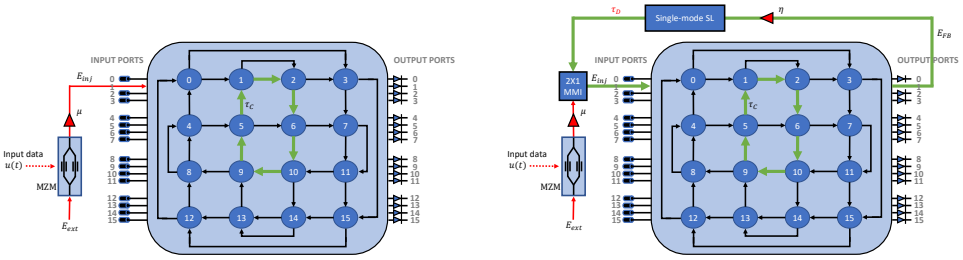


Fig. 10. Illustration of the spatially distributed FPA, with the shortest path which loops from input node 1 to itself depicted in green (a) and for the FPA+SL in the external delay line (b).

The trend of high memory capacities for internal delays τ_C smaller than the input segment duration τ_M continues also for the cubic memory capacity. However, the region of high memory capacities is slightly reduced to values $2\tau_C \lessapprox \tau_M$. This can be explained due to the fact that more signal mixing with other inputs is required for higher degrees of the memory capacity. However, this becomes difficult to fully explain in detail, as the calculation of the mixing of such signals becomes very cumbersome for all possible paths a signal can travel within the architecture. For the cubic memory capacity, we observe a high memory capacity around $\tau_M = 12\tau_C$. As we have mentioned previously, the shortest time it requires for an input sample to loop around the FPA and mix with another input sample is $6\tau_C$. This means that $12\tau_C$ corresponds to looping twice around the FPA, resulting in three input samples mixing with each other. If we consider $\tau_C \ll \tau_M$, the cubic memory capacity will decrease. It is possible that this is due to large amount of waveguides a signal has to travel through to mix with another input sample, leading to a high amount of power loss.

The values for the fourth and fifth degree of memory capacity are negligible for the FPA, indicating the limited amount of higher order nonlinearity present in this architecture.

The total memory capacity has lower values for $\tau_C > \tau_M$, which can be explained due to the fact that signals injected at $t = 0$ have not yet reached the next computing node at the next sampling event, when we sample at $t = \tau_M$. Note also that the maximum total memory capacity does not

Optics EXPRESS

reach its theoretical upper limit, given by the number of nodes (i.e. 16), which is most likely due to the amount of noise present in the system and due to correlations between nodes.

We now want to compare the memory capacity of the FPA with that of the FPA+SL. Therefore, we show in Fig. 11 the memory capacity up to the fifth degree for the FPA+SL. For the linear memory capacity, $C_{d=1}$, shown in Fig. 11(a), we again observe that the largest memory capacity is achieved when the internal delay time τ_C is equal to the input segment duration τ_M, i.e. $\tau_C = \tau_M$. For the second degree, $C_{d=2}$, shown in Fig. 11(b), we observe the highest memory capacities when $\tau_C \lessapprox \tau_M$. We also observe a region where the quadratic memory capacity is decreased, namely at $\tau_C = 5$ ps for $\tau_M \gtrapprox 40$ ps. The region of good cubic memory capacity, $C_{d=3}$, shown in Fig. 11(c), can be found around $2\tau_C \lessapprox \tau_M$, as is the case for the FPA. For the fourth and fifth memory capacity degrees, $C_{d=4}$ and $C_{d=5}$, shown in Fig. 11(d) and 11(e), we observe a region where these higher order memory capacities are not negligible, as opposed to the FPA. The total memory capacity, shown in Fig. 11(f), shows again a nearly uniform distribution over all τ_C and τ_M ranges, except for $\tau_C > \tau_M$.

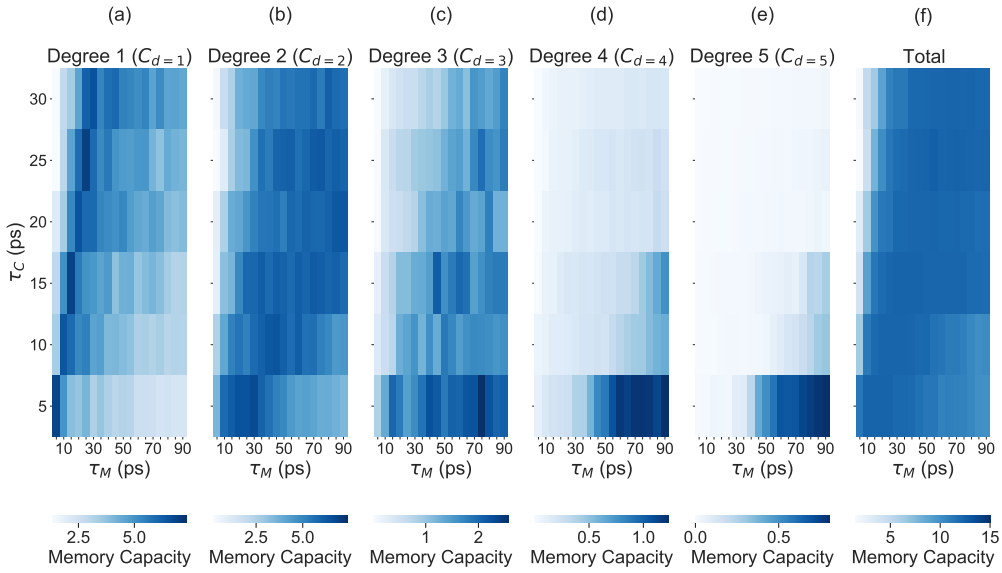


Fig. 11. Memory capacity per degree for the FPA+SL (a)-(e), and the total memory capacity (f). The internal delay time τ_C and input segment duration τ_M are varied, with the external delay time fixed to $\tau_D = 5$ ps. The injection rate is fixed to $\mu = 100\sqrt{10}$ s^{-1}.

If we compare the memory capacity of the FPA (shown in Fig. 9) with that of the FPA+SL (shown in Fig. 11), we observe that the linear and the total *MC* is very similar in both architectures. The FPA+SL, however, does offer a significant increase in the memory capacity of the fourth and fifth degree when compared to the FPA. As the total memory capacity *MC* is unchanged, this increase goes together with a corresponding decrease in the quadratic memory capacity of the FPA+SL [34,35]. We observe this shift of the nonlinear memory capacity to higher degrees not for all values of τ_C and τ_M, but only when τ_M is much larger than τ_C. For example, if $\tau_C = 5$ ps, we see a strong increase in the memory capacity of the fourth and fifth degree for $\tau_M \gtrapprox 40$ ps. This timescale coincides with the time needed for a segment of the input data stream to travel through the smallest loop in the FPA plus the time needed to travel through the external feedback loop. This path is indicated in green in Fig. 10(b), and the time needed to follow this path is $7\tau_C + \tau_D = 40$ ps.

Optics EXPRESS

The fact that we observe increased nonlinear memory capacities for higher degrees at low internal delay times τ_C, i.e. at $\tau_C = 5$ ps, has several advantages. By reducing τ_C, and therefore also shortening the coupling lengths between MMIs on-chip, we can reduce the fabrication cost and relax the demands on the optical power budgets as waveguide losses decrease with decreasing τ_C. Note, however, that the newly introduced SL in the external delay line comes at the cost of losing a physical computing node compared to the FPA. Nonetheless, this is a good compromise if we want to increase the nonlinearity in the reservoir of the RC system. To utilise this additional higher nonlinearity, we have to remark that an increased input segment duration τ_M needs to be used, which reduces the computation bandwidth of the combined system.

The similar performance of both systems on the Santa Fe task, shown in Figs. 5 and 8, can be understood heuristically by the results of the memory capacity. It seems that the Santa Fe task requires mainly a memory recall, and some low-order nonlinearity, such that we obtain good performance (i.e. low NMSE) when the linear and quadratic memory capacities are high. As these two memory capacities are similar for both systems, we can now understand the similar (good) performances on the Santa Fe task for $\tau_C \leq \tau_M$.

We also double-checked whether the SL is essential to the architecture for achieving the high nonlinear memory capacities, or if simply using an external delay is sufficient for this. This was done by looking into a simpler architecture without the SL in the external delay line. For this version of the architecture (not shown here), we have not observed any improvement in the fourth and fifth degrees of the memory capacity when compared to the FPA. This indicates that it is indeed the nonlinear behavior of the SL which increases these higher degrees of memory capacities and not the limited change to the architecture topology.

Future work aimed at further increasing the nonlinearity within the reservoir of this system could be fruitful. This can be done via various changes to the architecture, as we only have the requirement of using a delay line with a nonlinearity. For example, one could compare the effects of multiple delay lines coupled with different nodes in the architecture or investigate the effect of distributing a single feedback to different input nodes instead of one node, as done here. We expect that some improvements can be achieved with respect to increasing the nonlinearities, due to increased mixing of signals inside the architecture. Ultimately, this could result in a redistribution of the (non)linear memory capacity. Another interesting approach would be to use multiple input ports to inject data into the architecture, instead of using only a single input port.

4. Conclusion

In this paper, we have numerically investigated the effect of incorporating an additional nonlinearity in a passive photonic spatially distributed reservoir computing architecture, referred to as the FPA. This FPA lacks nonlinearity inside its reservoir. We have added a single nonlinear component to this FPA in the form of a single-mode SL in an external delay line. We find a comparable performance of both architectures on the Santa Fe one-step ahead time-series prediction task. Additionally, we observe that the delay in the feedback loop does not need to be optimized as the performance is largely independent of it. As a result, we can take a small external delay time so that the overall footprint of the system does not increase greatly. The time delay between the nodes of the spatially distributed system plays an important role in the performance. Still, precise tuning of this time delay is not needed on this task as long as it remains shorter than the time during which an input sample is injected. Moreover, we show that the added nonlinearity can be beneficial for more complex benchmark tasks, compared to the Santa Fe prediction task, reflected by the higher memory capacities for higher degrees when we combine the FPA with the SL. This increased nonlinearity of the reservoir can thus be achieved by adding only a single nonlinear element in the network. The higher nonlinearity indicates that the FPA+SL will enhance the task solving capability and result in better computational performance when a more complex nonlinearity is required for a more complex benchmark task.

Optics EXPRESS

Funding. Fonds De La Recherche Scientifique - FNRS (Excellence of Science, EOS number 40007536); Fonds Wetenschappelijk Onderzoek (Excellence of Science, EOS number 40007536, G006020N, G028618N, G029519N).

Disclosures. The authors declare no conflicts of interest.

Data Availability. Data underlying the results presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

References

1. T. Eloundou, S. Manning, P. Mishkin, *et al.*, "GPTs are GPTs: an early look at the labor market impact potential of large language models," (2023).
2. GPT-4 Technical Report, 2023.
3. G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," Nanophotonics **6**(3), 561–576 (2017).
4. T. F. De Lima, B. J. Shastri, A. N. Tait, *et al.*, "Progress in neuromorphic photonics," Nanophotonics **6**(3), 577–599 (2017).
5. G. Tanaka, T. Yamane, J. B. Héroux, *et al.*, "Recent advances in physical reservoir computing: A review," Neural Networks **115**, 100–123 (2019).
6. M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," Computer Science Review **3**(3), 127–149 (2009).
7. D. Verstraeten, B. Schrauwen, M. d'Haene, *et al.*, "An experimental unification of reservoir computing methods," Neural networks **20**(3), 391–403 (2007).
8. K. Vandoorne, P. Mechet, T. Van Vaerenbergh, *et al.*, "Experimental demonstration of reservoir computing on a silicon photonics chip," Nat. Commun. **5**(1), 3541 (2014).
9. M. R. Salehi, E. Abiri, and L. Dehyadegari, "An analytical approach to photonic reservoir computing–a network of SOA's–for noisy speech recognition," Opt. Commun. **306**, 135–139 (2013).
10. 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.
11. H. Hasegawa, K. Kanno, and A. Uchida, "Parallel and deep reservoir computing using semiconductor lasers with optical feedback," Nanophotonics **12**(5), 869–881 (2023).
12. H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," Science **304**(5667), 78–80 (2004).
13. E. S. Skibinsky-Gitlin, M. L. Alomar, E. Isern, *et al.*, "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.
14. D. Canaday, A. Griffith, and D. J. Gauthier, "Rapid time series prediction with a hardware-based reservoir computer," Chaos: An Interdiscip. J. Nonlinear Sci. **28**(12), 123119 (2018).
15. Y. Paquot, F. Duport, A. Smerieri, *et al.*, "Optoelectronic reservoir computing," Sci. Rep. **2**(1), 287 (2012).
16. S. Sackesyn, C. Ma, A. Katumba, *et al.*, "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.
17. K. Harkhoe, G. Verschaffelt, A. Katumba, *et al.*, "Demonstrating delay-based reservoir computing using a compact photonic integrated chip," Opt. Express **28**(3), 3086–3096 (2020).
18. S. Sunada and A. Uchida, "Photonic neural field on a silicon chip: large-scale, high-speed neuro-inspired computing and sensing," Optica **8**(11), 1388–1396 (2021).
19. K. Vandoorne, W. Dierckx, B. Schrauwen, *et al.*, "Toward optical signal processing using photonic reservoir computing," Opt. Express **16**(15), 11182–11192 (2008).
20. K. Vandoorne, J. Dambre, D. Verstraeten, *et al.*, "Parallel reservoir computing using optical amplifiers," IEEE Trans. Neural Netw. **22**(9), 1469–1481 (2011).
21. M. R. Salehi and L. Dehyadegari, "Optical signal processing using photonic reservoir computing," J. Mod. Opt. **61**(17), 1442–1451 (2014).
22. T. Hülser, F. Köster, K. Lüdge, *et al.*, "Deriving task specific performance from the information processing capacity of a reservoir computer," Nanophotonics **12**(5), 937–947 (2023).
23. F. Duport, B. Schneider, A. Smerieri, *et al.*, "All-optical reservoir computing," Opt. Express **20**(20), 22783–22795 (2012).
24. 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," Sci. Rep. **9**(1), 5918 (2019).
25. M. A. A. Fiers, T. Van Vaerenbergh, F. Wyffels, *et al.*, "Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns," IEEE Trans. Neural Netw. Learning Syst. **25**(2), 344–355 (2014).
26. K. Harkhoe and G. Van der Sande, "Task-independent computational abilities of semiconductor lasers with delayed optical feedback for reservoir computing," in *Photonics*, vol.6 (Multidisciplinary Digital Publishing Institute, 2019), p. 124.

Optics EXPRESS

27. I. Bauwens, K. Harkhoe, P. Bienstman, *et al.*, "Influence of the input signal's phase modulation on the performance of optical delay-based reservoir computing using semiconductor lasers," Opt. Express **30**(8), 13434–13446 (2022).
28. D. Lenstra and M. Yousefi, "Rate-equation model for multi-mode semiconductor lasers with spatial hole burning," Opt. Express **22**(7), 8143–8149 (2014).
29. A. Weigend and N. Gershenfeld, The Santa Fe Time Series Competition Data (1991).
30. R. M. Nguimdo, G. Verschaffelt, J. Danckaert, *et al.*, "Fast photonic information processing using semiconductor lasers with delayed optical feedback: Role of phase dynamics," Opt. Express **22**(7), 8672–8686 (2014).
31. M. C. Soriano, S. Ortín, D. Brunner, *et al.*, "Optoelectronic reservoir computing: tackling noise-induced performance degradation," Opt. Express **21**(1), 12–20 (2013).
32. E. W. Dijkstra, "A note on two problems in connexion with graphs," Numer. Math. **1**(1), 269–271 (1959).
33. J. Bueno, D. Brunner, M. C. Soriano, *et al.*, "Conditions for reservoir computing performance using semiconductor lasers with delayed optical feedback," Opt. Express **25**(3), 2401–2412 (2017).
34. J. Dambre, D. Verstraeten, B. Schrauwen, *et al.*, "Information processing capacity of dynamical systems," Sci. Rep. **2**(1), 514 (2012).
35. M. Inubushi and K. Yoshimura, "Reservoir computing beyond memory-nonlinearity trade-off," Sci. Rep. **7**(1), 10199 (2017).