

Improving Time Series Recognition and Prediction With Networks and Ensembles of Passive Photonic Reservoirs

Matthias Freiberger¹, Student Member, IEEE, Stijn Sackesyn, Chonghui Ma, Andrew Katumba², Student Member, IEEE, Peter Bienstman³, Member, IEEE, and Joni Dambre⁴

(Invited Paper)

Abstract—As the performance increase of traditional Von-Neumann computing attenuates, new approaches to computing need to be found. A promising approach for low-power computing at high bitrates is integrated photonic reservoir computing. In the past though, the feasible reservoir size and computational power of integrated photonic reservoirs have been limited by hardware constraints. An alternative solution to building larger reservoirs is the combination of several small reservoirs to match or exceed the performance of a single bigger one. This paper summarizes our efforts to increase the available computational power by combining multiple reservoirs into a single computing architecture. We investigate several possible combination techniques and evaluate their performance using the classic XOR and header recognition tasks as well as the well-known Santa Fe chaotic laser prediction task. Our findings suggest that a new paradigm of feeding a reservoir's output into the readout structure of the next one shows consistently good results for various tasks as well as for both electrical and optical readouts and coupling schemes.

Index Terms—Integrated photonic reservoir computing, deep reservoir computing, scalable reservoir computing, unconventional computing, neuro-inspired computing, neuromorphic computing.

I. INTRODUCTION

AS THE performance increase of traditional Von-Neumann computing attenuates with the apparent end of Moore's Law [1], new, unconventional computing approaches have been recently proposed. One of them, reservoir computing [2]–[4], originally proposed as an effective training method for recurrent neural networks, appears to be especially promising as it is

Manuscript received April 3, 2019; revised May 10, 2019 and July 13, 2019; accepted July 13, 2019. Date of publication July 25, 2019; date of current version August 9, 2019. This work was supported by the EU Horizon 2020 PHRESCO Project under Grant 688579. (Corresponding author: Matthias Freiberger.)

M. Freiberger and J. Dambre are with the Ghent University-imec IDLab, Department of Electronics and Information Systems, Technologiepark-Zwijnaarde 126, B-9052 Ghent, Belgium (e-mail: matthias.freiberger@ugent.be; joni.dambre@ugent.be).

S. Sackesyn, C. Ma, A. Katumba, and P. Bienstman are with Ghent University-imec Photonics Research Group, Department of Information Technology, Technologiepark-Zwijnaarde 126, B-9052 Ghent, Belgium (e-mail: stijn.sackesyn@ugent.be; chonghui.ma@ugent.be; andrew.katumba@ugent.be; peter.bienstman@ugent.be).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTQE.2019.2929699

well-fit to exploit the natural dynamics of physical systems [5]–[7]. In recent years many optical computing approaches have applied reservoir computing [8]–[16]. Among those, integrated passive photonic reservoir computing or silicon reservoir computing [17] is an interesting candidate for low-power computation at very high baudrates (>10 Gbdps). Nevertheless, the underlying technology is still plagued by a number of limitations. In the past, the feasible reservoir size and computational power of passive photonic reservoirs have been limited by a number of factors. Among these are high optical losses, a limited choice of suitable optical nonlinearities, and high hardware and wiring effort. Recently though, several constraints could be relaxed [18], [19]. Nevertheless, whereas new architectures and hardware enable the construction of larger reservoirs, an alternative route to explore is the combination of several small reservoirs to match or exceed the performance of a single bigger one. Moreover, it is well known from neural network literature [20] that performing subsequent nonlinear transformations on the input data by stacking a large number of non-linear neural network layers is highly beneficial in terms of performance on a wide variety of tasks. These insights have led to the paradigm of deep learning. In this context, the cascading of passive photonic reservoirs seems worth exploring, since it holds the promise of immense computational power under the assumption that combining several reservoirs yields similar improvements in performance as have been experienced when implementing neural networks in software. In this work we summarize our recent efforts to increase the available computational power by combining multiple reservoirs into a single computing device. As a first step, we discuss our approach and introduce the different investigated architectures in Section II. Thereafter, we describe our methodology and experimental setup in Section III. The following sections discuss our conducted experiments, where we group all investigated architectures by their mode of coupling. In Section IV, we investigate architectures making use of an electrical readout such that training as well as coupling of individual reservoirs is conducted in the electrical domain. Thereafter, in Section V, we proceed to fully optically trained and coupled reservoirs. Finally, we sum up our findings and conclusions and allude towards future directions of research in Section VI.

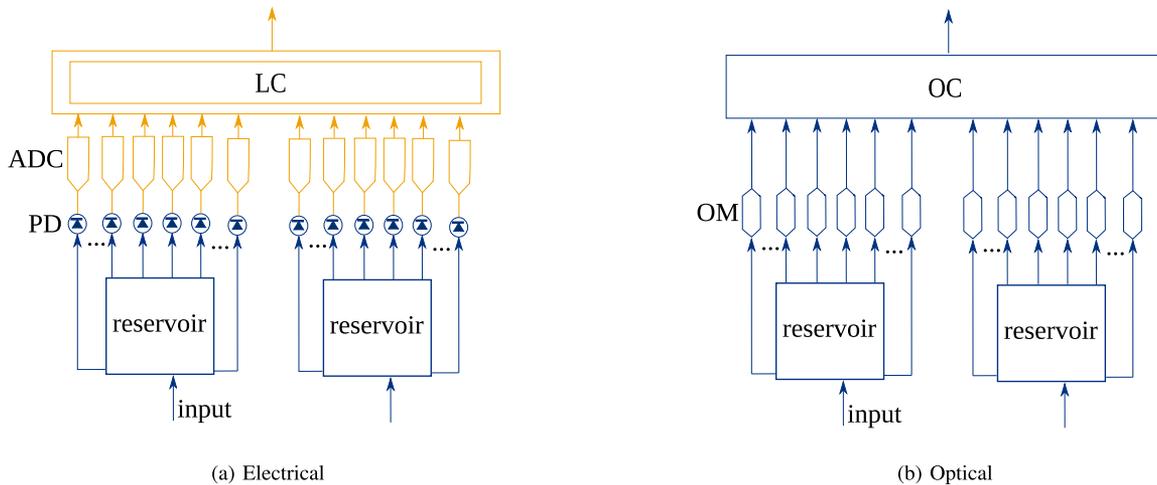


Fig. 1. Examples of an ensemble of photonic reservoirs connected to a single readout for electrical and optical coupling in Fig. 1a and 1b respectively. The weights of all reservoirs are trained jointly, i.e. as a single weight vector. PD: photodiode, ADC: analog-digital converter, LC: linear classifier, OM: optical modulator, OC: optical combiner.

II. INVESTIGATED RESERVOIR ARCHITECTURES

In the context of feasibility for mass production in combination with inter-chip fabrication variabilities, using fast, closed-form solution algorithms for training is mandatory. The strongest available candidate for such an algorithm to date is complex-valued ridge regression [21]. Therefore, an essential criterion for any cascaded setup of reservoirs is that the individual reservoirs still need to be trainable using ridge regression, rather than more powerful but slower iterative training algorithms. For this reason, our efforts to date are limited to setups which can be instantly trained using ridge regression. The methods investigated will be introduced in this section. Specifically, we investigated the applicability of the classical combination techniques ensembling and boosting [22] to photonic reservoir computing. Furthermore, we have also evaluated the paradigm of stacking, which has already been applied in the context of reservoir computing [23], [24]. Furthermore, we introduce a new combination technique inspired by stacking approaches as well as [25], which we refer to as chaining.

A. Ensembling

In ensembling [22], several classifiers are trained for the same task and combined by taking a combination of the individual classifier predictions. In the simplest case, classifier predictions are averaged which is commonly referred to as bagging. More advanced approaches train a classifier to take a weighted sum of the predictions of all former classifiers in an attempt to combine the strengths of all previously trained classifiers and average out their weaknesses. An ensemble of passive photonic reservoirs is straightforward to implement by adding an additional weighting layer on top of several photonic reservoirs working in parallel. A second, even simpler method would be to simply connect the nodes of several reservoirs to a single readout (see Fig. 1). This is the approach we followed within this work, since bagging consistently exhibited poor performance

in preliminary experiments. While ensembles can deliver great improvements at moderate implementation effort, in order to work well, the models in an ensemble must be different, i.e. the mistakes they make must be as uncorrelated as possible. Any two passive photonic reservoirs are different by construction due to the silicon photonics manufacturing process, which affects the effective indices of their waveguides. This translates to strong phase variations in the reservoir connections affecting the individual signals of their corresponding reservoir states and makes them ideal candidates for reservoir ensembling approaches in hardware.

B. Boosting

Boosting [22], [26], [27] is a technique used successfully in the past to combine several weak classifiers into a stronger classifier. While there are several different forms of boosting, gradient boosting is the most attractive to our purposes since it can be implemented in a straightforward way in hardware as seen in Fig. 2. In order to combine several reservoirs into a gradient boosting approach, only the first reservoir is trained on the desired output. Its resulting prediction is subtracted from the original desired signal to form the desired signal that is used for training the second reservoir. That way, the second reservoir is actively trained to correct the first reservoir's errors when the two output predictions are subsequently added in hardware. This procedure can be repeated as often as necessary, adding more reservoirs in the process to correct the remaining error.

C. Stacking

When stacking classifiers [22]–[24], one presents the prediction signal of a given classifier as input signal to a subsequent classifier. The underlying idea is that later classifiers will be able to correct the errors of previous classifier stages. Despite receiving different input signals, all classifiers are trained using the same desired output signal. Fig. 3 illustrates two classifiers

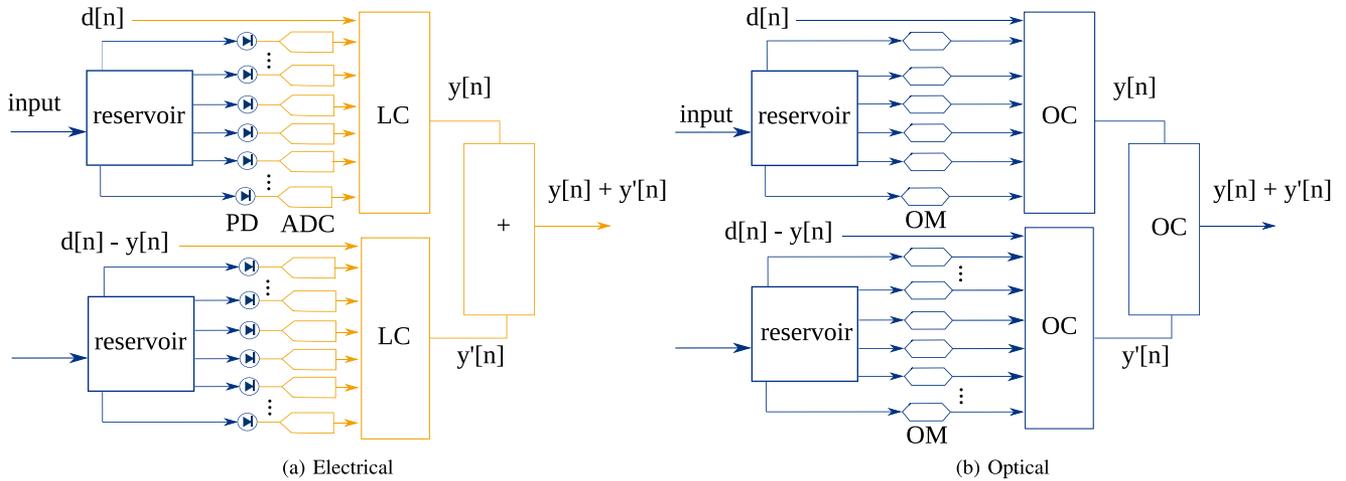


Fig. 2. Example of gradient boosting with two photonic reservoirs using electrical and optical coupling in Fig. 2a and 2b respectively. Added reservoirs are trained on the difference between the desired signal $d[n]$ and the actual summed output of all previous reservoirs $y[n]$. That way each added reservoir attempts to add a prediction signal complementary to the current sum of prediction signals such that the sum of all signals predicts the desired signal as well as possible. PD: photodiode, ADC: analog-digital converter, LC: linear classifier, OM: optical modulator, OC: optical combiner.

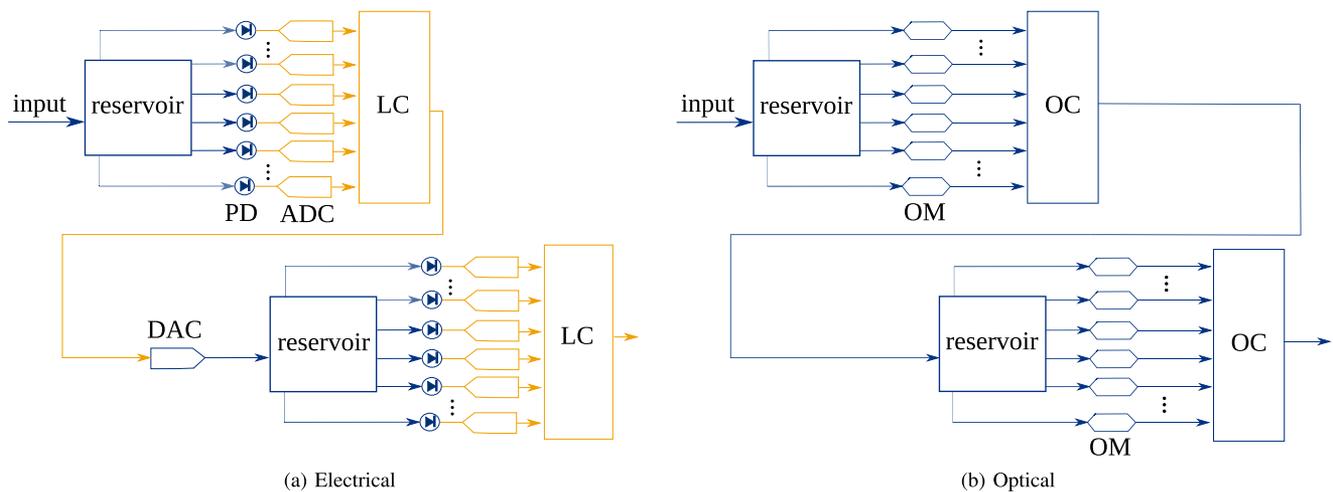


Fig. 3. Example of two photonic reservoirs with integrated optical readout stacked upon each other using electrical and optical coupling in Fig. 3a and 3b respectively. Subsequent reservoirs receive the predicted output of previous stages as an input and attempt to improve upon it. All readouts are trained on the same desired signal. PD: photodiode, ADC: analog-digital converter, DAC: digital-analog converter, LC: linear classifier, OM: optical modulator, OC: optical combiner.

stacked as described above, where training and coupling between reservoirs can be realized in the electrical or the optical domain.

D. Chaining

Chaining is a connection scheme inspired by similar techniques [23]–[25], in which the predicted output of a given reservoir is fed as an extra weighted input to the readout of a subsequent reservoir (see Fig. 4 for illustration). That way, an additional reservoir is trained directly to improve an already trained prediction signal. Again, this connection step can be repeated a number of times adding more reservoirs in the process. Just like in the previous techniques, each additional reservoir should be as different as possible from the previous one(s).

Note that the chaining and stacking approaches are similar to the recent approach followed in [25], but not identical. In this work, each reservoir module is driven with all the states of its predecessor (with untrained weights). The readout is trained on the aggregated states of all reservoirs in the ensemble. We expect the architecture in [25] could perform better than the one studied here because, as the information flows from reservoir to reservoir, each subsequent reservoir in [25] has memory that reaches further into the past. Translated to integrated photonics technology, it would have to be simplified, e.g., by projecting a random combination of each reservoirs states back into the next reservoir and training the readout on all states in the electrical domain. As our simulation studies have been conducted in parallel with the development of an actual chip prototype, and shall be compared to the performance of the actual prototypes in the

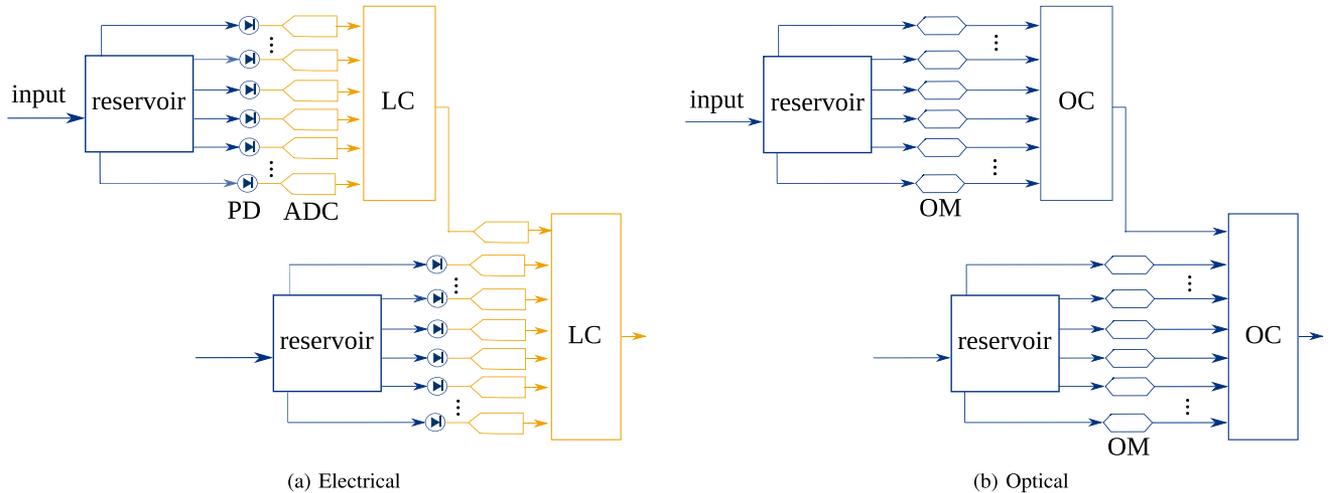


Fig. 4. Example of chaining with two photonic reservoirs using electrical and optical coupling in Fig. 4a and 4b respectively. The classifier stage of a subsequent reservoir obtains the prediction of the previous reservoir as an input. The aim is to improve upon the previous prediction using the reservoir states of the subsequent reservoir as an input. PD: photodiode, ADC: analog-digital converter, LC: linear classifier, OM: optical modulator, OC: optical combiner.

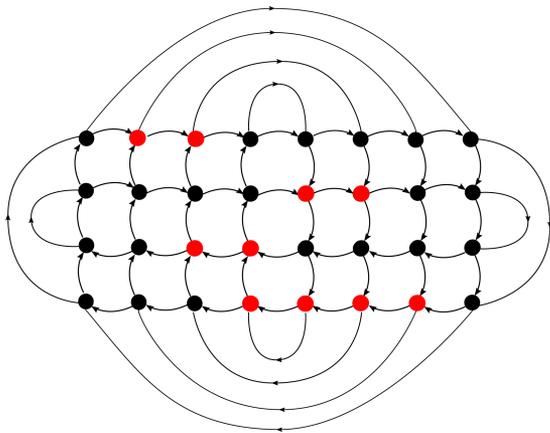


Fig. 5. Illustration of the modified swirl architecture of the simulated reservoir as proposed by Sackesyn *et al.* [19]. In addition to the inner delay lines of the nodes, delay lines are added outside of the swirl node grid. Input is injected through the marked nodes.

future, we have constrained ourselves to designs that fit within the implementation constraints of this prototype. Nevertheless developing integrated photonic reservoir designs closer to the DeepESN architecture appears to be an appealing direction for future research.

III. METHODOLOGY

In our simulation studies, we use an updated version of the classical swirl architecture [17]: in addition to the inner delay lines of the nodes, delay lines are added outside of the swirl node grid, as illustrated in Fig. 5. A simulated 1550 nm passive photonic reservoir utilizing this updated 4×8 version of the swirl architecture as proposed by Sackesyn *et al.* [19] is used as the essential building block in our simulations. This reservoir is modeled as an optical circuit

comprising multimode interferometers as nodes connected by shallow-etched waveguides as delay lines between nodes. These delay lines are modeled to exhibit an average loss of 1 dB/cm and are 2.14 mm long, to match the timescales of the reservoir to 32 Gbits. However, for some node pairs that need to be connected, the euclidean distance would be larger than 2.14 mm. Therefore, we choose the delay lines between these nodes to be twice the length of the other delay lines, namely 4.28 mm long. Note that an integer multiple of the original delay is chosen here, since preliminary experiments have shown that for a passive photonic reservoir with delay lines of mixed lengths, all lengths should be integer multiples of the shortest delay line length.

As a baseline reservoir to compare our cascaded architectures, we use a single simulated reservoir of identical architecture and technology, but with 4 times the number of nodes.

A. Simulation Setup

We simulate the response of our optical integrated circuit to the upsampled, intensity-modulated input signal using Caphe [28], [29] as described in Appendix A. We obtain a sampled complex output signal of each reservoir node as result, denoting amplitude and phase of the optical signal at that node at a certain instant in time. Now we proceed differently for each of the coupling states used for the investigated architecture. For electrical coupling, we convert the complex-valued optical signal at each of the 32 nodes of the reservoir from the optical into the electrical domain using a photodetector model which is described at length in Appendix B. The resulting samples of 32 simulated electrical signals are then arranged in a time series of real-valued reservoir state vectors from which a weighted linear combination (the classifier) is taken to obtain an output signal solving the problem at hand. The obtained output signal can be treated as the final answer of the system, or reconverted into the optical domain to be passed on to a following optical stage which is simulated in an identical way.

For optically cascaded reservoirs, which make use of an integrated optical readout we follow the modeling and training process of an integrated optical readout as described in [21]. The complex-valued optical response signal is obtained as described for the electrical approach. However, instead of converting this signal to the electrical domain, the sampled complex optical signals at each reservoir node are arranged into a time series of complex state-node vectors $X \in \mathcal{C}$. The integrated optical readout is simulated by computing an inner product between each of these vectors and a complex weight vector w , which represents the complex optical weights. The resulting complex-valued signal is either fed into the next stage of a given reservoir architecture, which can be simulated in an identical way, or passed to the photodetector model to obtain the final electrical output signal. In both the electrical and optical cases, the final electrical output signal is downsampled in the middle of the bit period. After downsampling, the signal is thresholded for binary tasks, such as the XOR and the header recognition tasks, in order to obtain a clean binary output bit sequence.

To gain conclusive results on the performance of our cascaded reservoir systems, we sweep the symbol rate between 20 and 40 Gbdps (with the exception of the 3 bit XOR task where the bit rate is swept between 30 and 50 Gbps since Vandoorne *et al.* [17] have shown that for higher delay XOR tasks, larger delays between nodes are necessary). For a reservoir with fixed delay line lengths, changing the symbol rate changes the number of symbols transmitted between nodes in a given time interval and thus the amount of time steps (symbols) of the previous signal that still affect the internal state of the reservoir. Therefore performing a sweep over the bitrate enables us to investigate our architecture's performance when varying the reservoir's memory while still relating to a single optical circuit. Therefore the symbol rate has a significant influence on the performance of our investigated systems. While one could simply keep the symbol rate fixed and perform sweeps over the delay line length of the reservoir instead, this approach is favored in order to be able to compare our obtained simulation results with the results on future actual implementations of the simulated circuits.

As we sweep the symbol rate, we always sample 5 times during every symbol period of the signal, which results in a sampling rate of 160 GHz at a symbol rate of 32 Gbps, with higher or lower sampling rates, respectively, for other bitrates. The number of samples per symbol period is kept fixed for varying bitrates in order to supply our machine learning algorithms with always the same amount of training data. This prevents degrading or improving performance due to large or small amounts of training data. All obtained results are averaged over 10 different architecture instances, in which the reservoirs have been initialized with different random phases for their input and connection waveguides, in order to account for the high manufacturing variability of integrated photonic reservoirs and combinations thereof.

B. Tasks

As tasks to evaluate the performance of our reservoirs we use the delayed XOR task with 3 bits delay, the 1 sample ahead

prediction Santa Fe task [30], and the 5 bit header recognition task. For this last task, the pattern 10101 needs to be found in the input bit stream. We chose this pattern since we have found it to be among the more difficult bit patterns to detect compared to easy patterns such as 11111. As we perform the detection of a 5 bit header, this implies that we need sufficient training data to train our classifiers. Since the probability of a single 5 bit header occurring in a 10000 bit uniform randomly drawn sequence is rather low, this implies a heavy class-imbalance in our training data with only very little positive samples. Using longer sequences of bits would yield in significantly longer-running simulations and does not solve this problem since this imbalance of data is independent of sequence length. Therefore we choose a different path and generate our bit sequences in the following way. We generate a random bit sequence, where we randomly draw bits from a uniform distribution, but insert additional instances of the desired header pattern based on a Poisson random variable. The detailed process used to generate the random input bit sequences is described at length in Appendix C.

We train all our readouts using ridge regression, where we perform 5-fold cross-validation to find the optimal regularization parameter for each reservoir at each bitrate. We validate our algorithms on a separate test set. For the XOR and header recognition tasks, the training set and test set both consist of 10000 randomly generated bits. As an error measure for these tasks, we use the bit error rate (BER), which is defined as

$$e_{\text{bit}} = \frac{1}{N} \sum_{n=1}^N \mathbb{1}[y_T[n] \neq d[n]], \quad (1)$$

where $y_T[n]$ is the subsampled, thresholded output signal of the reservoir and $d[n]$ is the desired signal. Given that we generate 10000 bits of test data, the minimal bit error rate, which can be estimated with a confidence level of $\approx 90\%$, is 10^{-3} [31]. For the Santa Fe Laser prediction task, we use the original training set of 1000 samples as well as all remaining available data for testing which results in a test set of 9093 samples. For the Santa Fe task, as an error measure, we use the normalized mean square error (NMSE) defined as

$$e_{\text{NMSE}} = \frac{1}{N} \sum_{n=1}^N \frac{(y[n] - d[n])^2}{\sigma_d^2}. \quad (2)$$

$y[n]$ is the subsampled output signal of the reservoir and σ_d^2 is the variance of the desired signal $d[n]$.

C. Pre- and Postprocessing

We preprocess all input signals to the reservoir by upsampling them to 5 samples per symbol/bit, applying a low-pass filter and power-encoding them such that the overall maximum input power to the reservoir is 0.1 W, where we set a fixed bias of 0.02 W. This leaves 0.08 W for the signal amplitude. Note that this power is distributed over all input nodes, which yields a maximum input power of $0.1/10 = 0.01\text{W}$ for each one of the 10 input nodes. A detailed description of the input node configuration can be found in Appendix D.

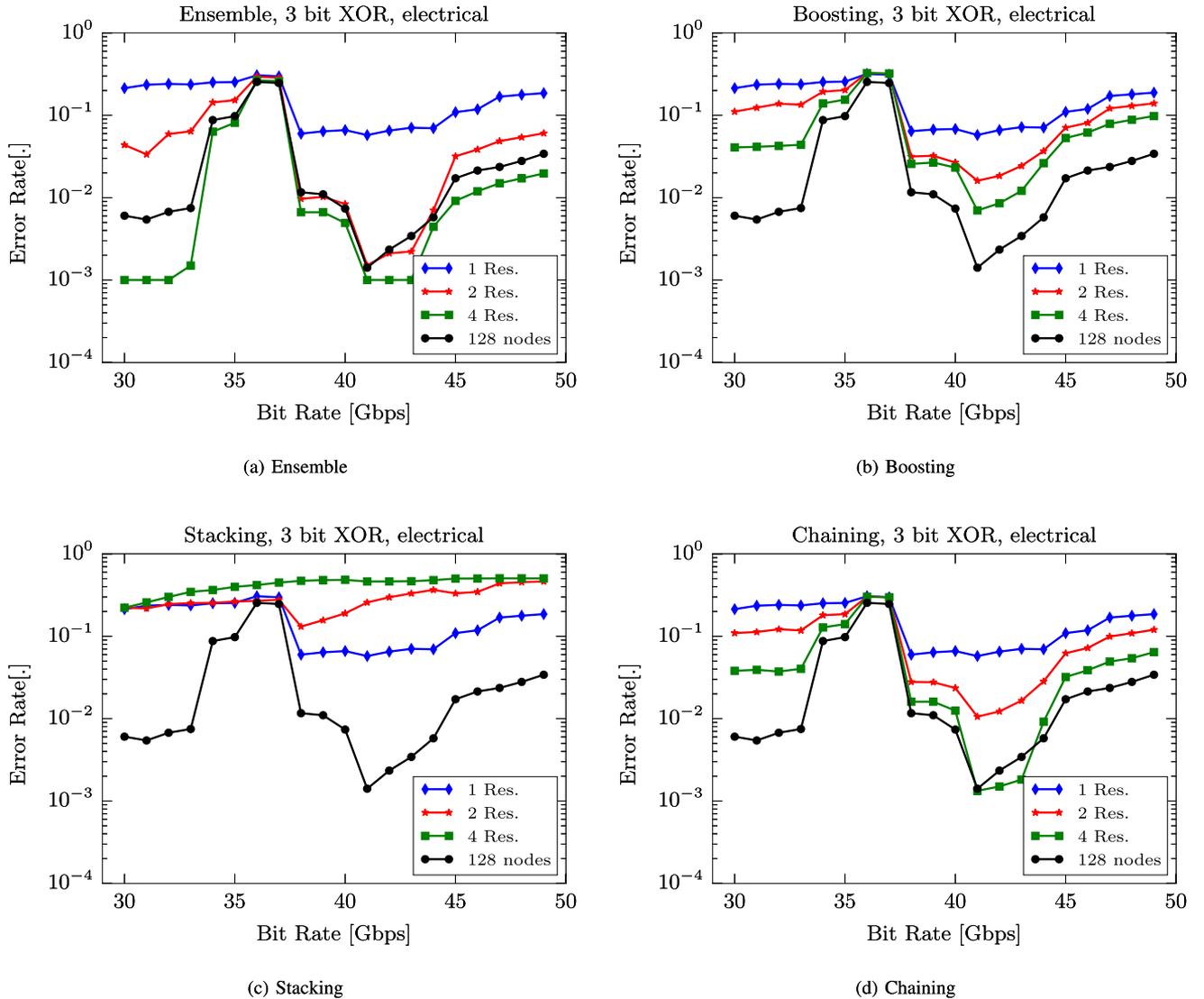


Fig. 6. Bit error rate of simulated 1550 nm prototype on 3 bit XOR as a function of bitrate for 1, 2, and 4 reservoirs combined using ensembling, boosting, stacking and chaining in the electrical domain. Comparison with 8×16 node baseline. The minimum detectable error rate is 10^{-3} .

We postprocess all reservoir prediction signals by down-sampling them to 1 sample per symbol/bit, where the signals are sampled in the middle of each bit/symbol period. For our digital tasks, in addition to the sampling, we threshold the signal in postprocessing, where the threshold is determined on the training prediction of the reservoir. We found here that for header recognition using half the difference between minimum and maximum value works best, while the XOR task performs slightly better when using half the difference between the 80th and 20th percentile instead.

IV. COMBINING RESERVOIRS WITH ELECTRICAL READOUTS

As a first step, we evaluate the cascading strategies introduced in Section II when performing both training and coupling in the electrical domain. More specifically, we detect the power on all nodes equipped with an electrical readout. Thereafter, we

normalize all node channels to be zero-mean with unit variance and train real-valued weight vectors based on these normalized reservoir states using ridge regression. As already mentioned, the coupling between reservoirs happens in the electrical domain as well, meaning that intermediate results and target signals are evaluated in the electrical domain after which they are transferred back into the optical domain for further processing in the next stages using amplitude encoding. To evaluate the performance of our cascaded architectures, we have used the delayed XOR tasks with 3 bit delay, as well as the Santa Fe time series prediction task where we aim to predict the next sample in a recorded time-series generated by a laser operating in a chaotic regime.

Fig. 6 shows the results for the 3 bit delayed XOR task: the ensemble performs best and shows significant improvement on a wide range of bit rates as more reservoirs are added to the ensemble. It is notable that even 2 smaller reservoirs trained

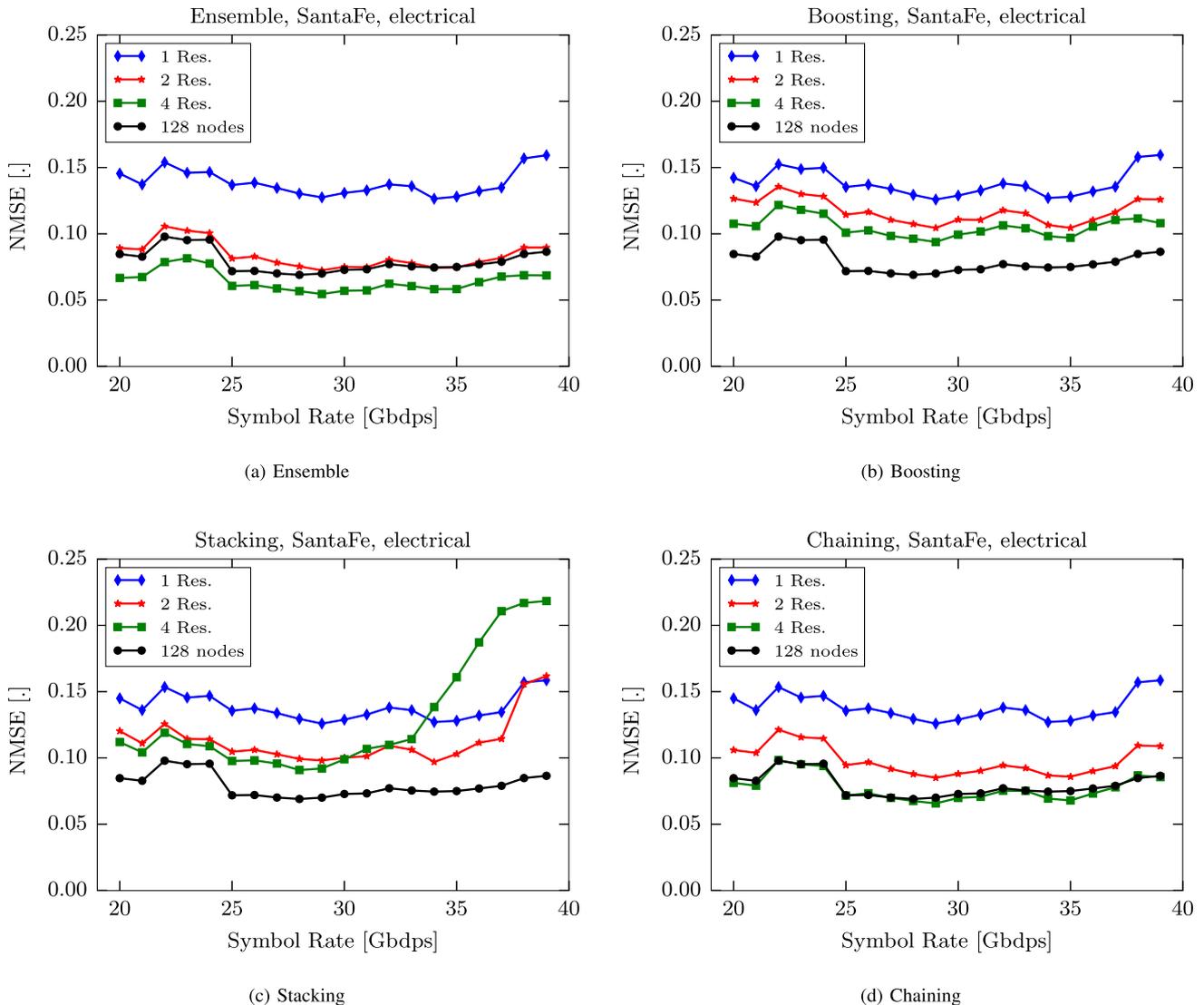


Fig. 7. Bit error rate of simulated 1550 nm prototype on the Santa Fe time series prediction task as a function of bitrate for 1, 2, and 4 reservoirs combined using ensembling, boosting, stacking and chaining in the electrical domain. Comparison with 8×16 node baseline. The minimum detectable error rate is 10^{-3} .

as an ensemble at times manage to outperform the 128 node baseline reservoir which contains twice the number of nodes. A possible explanation for this effect is that several small reservoirs introduce more richness and variation in the resulting combined reservoir states than would be possible for a single larger reservoir. A further observation we made in preliminary experiments, i.e. that cascaded reservoirs need to vary between themselves in order to improve the error rate, also supports this conclusion. The gradient boosting approach is outperformed by the baseline, ensembling and chaining but still seems to show moderate improvement as more reservoirs are added to correct classification errors. The approach of stacking reservoirs on the other hand seems to be unsuitable for our current setup: results for stacked reservoirs turn out worse than the original single prototype for high bitrates. Finally, our chaining approach outperforms the baseline on a few bitrates, but is slightly outperformed on most bitrates.

In order to also measure performance on an analog task in addition to the bit pattern task above, we used the Santa Fe chaotic laser prediction task as a benchmark for our systems. In this task we predict the next sampling value for a time series recorded from a far IR laser driven in a chaotic regime. Fig. 7 shows the results of our systems on this task.

As we can see, the results are mostly consistent with our observations on the XOR task: ensembling outperforms the baseline. Stacking seems to be better suited for analog tasks and performs comparably to boosting in the low and intermediate regions of the observed range of bitrates. Both boosting and stacking are nevertheless significantly outperformed by the baseline. For higher bitrates the performance of stacking gets significantly worse. Finally, the chaining approach is on par with the baseline. In order to compare our result with delayed feedback approaches, we refer to the work of Soriano *et al.* [11] who report an NMSE of 0.025 (using a train set of 3000 samples

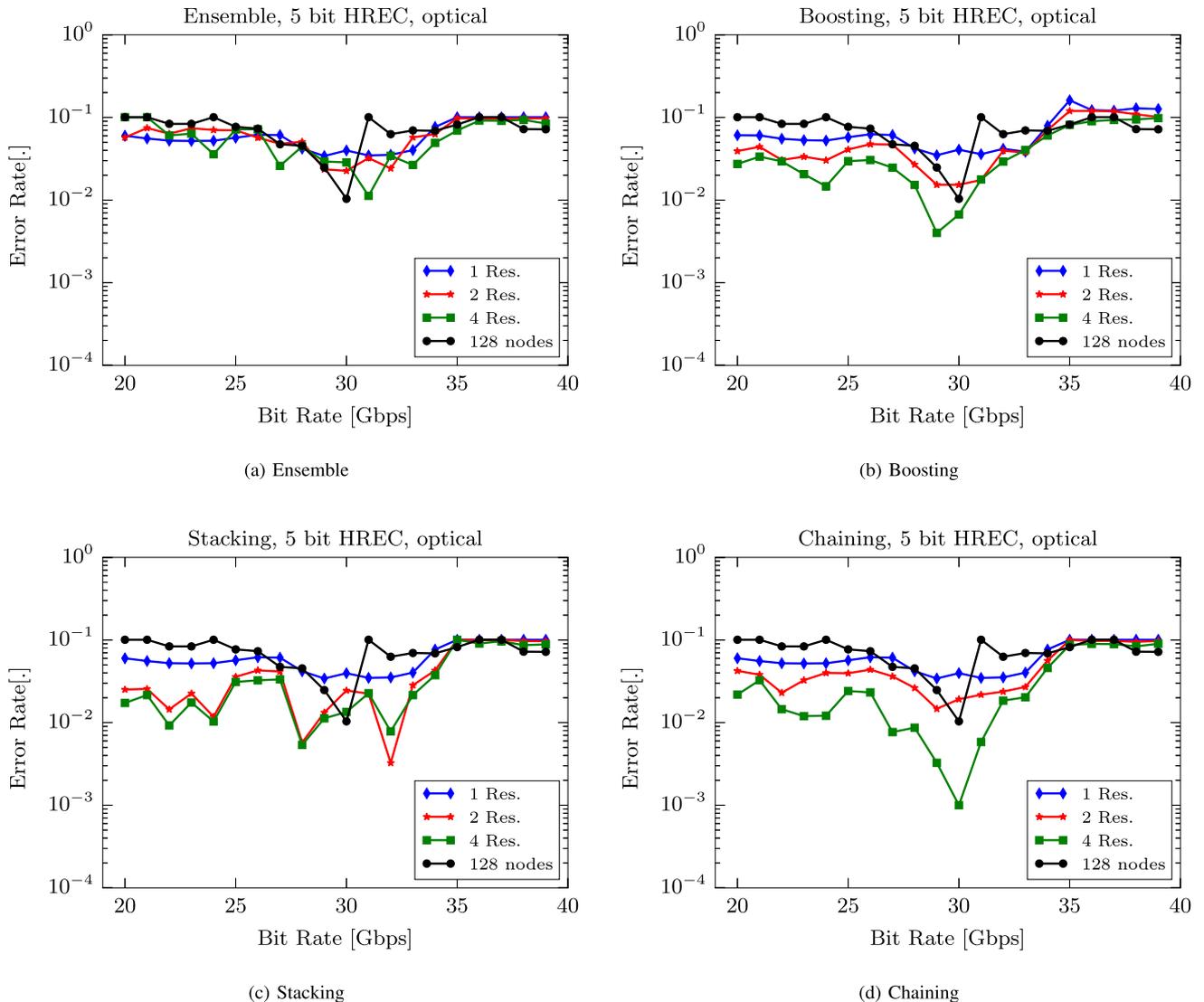


Fig. 8. Bit error rate of simulated 1550 nm prototype on 5 bit header recognition (pattern 10101) as a function of bitrate for 1, 2, and 4 reservoirs combined using ensembling, boosting, stacking and chaining in the optical domain. Comparison with 8×16 node baseline. The minimum detectable error rate is 10^{-3} .

and a test set of 1000 samples) on the Santa Fe dataset for a delayed feedback reservoir computer. This was however for a 500-node system, much larger than the 4 reservoirs with a total number of 128 nodes (32 nodes per reservoir), which have been used here.

V. COMBINING RESERVOIRS IN THE OPTICAL DOMAIN

As a second family of architectures, we discuss architectures with full optical coupling. Also, all intermediate signals are summed and processed in the optical domain. As we do not assume any optical nonlinearity in the system, the choice of applicable tasks is more limited for this setup than in the electrical case. We use the 5 bit header recognition task here to assess and compare the performance of cascaded systems to a larger baseline. Fig. 8 shows the results for our simulated all-optical architectures.

On the header recognition task we can spot several differences when comparing to the obtained results in the electrical domain. Both the previously best performing ensembling approach as well as the baseline now perform significantly worse than the chaining approach. A possible explanation here might be that the 5 bit HREC task requires significantly more memory than previous tasks, which can only be provided by approaches in which the signal is actually processed in several stages, receiving additional information about the input signal from previous reservoirs. This is the case for chaining, boosting and stacking, which show moderate improvement as more reservoirs are added. It seems noteworthy that, contrary to the electrical coupling case, the stacking approach seems to yield moderate improvements over the baseline at certain bitrates. The chaining approach seems to outperform the remaining approaches, but nevertheless manages to minimize the error rate only for a bitrate of 30 Gbps. We would like to emphasise that chaining reservoirs

TABLE I
RESULTS FOR ELECTRICAL TRAINING/COUPLING AT 30 Gbps

Task	Ensemb.	Boost.	Stack.	Chain.	Basel.
XOR 3 bit (BER)	0.001	0.041	0.222	0.038	<i>0.006</i>
Santa Fe (NMSE)	0.057	0.099	0.099	<i>0.070</i>	0.073

TABLE II
RESULTS FOR OPTICAL TRAINING/COUPLING AT 30 Gbps

Task	Ensemb.	Boost.	Stack	Chain.	Basel.
HREC 5 bit (BER)	0.029	<i>0.007</i>	0.013	0.001	0.010

does not increase the memory of the reservoirs themselves, contrary to approaches such as [25]. Instead, chaining appears to improve the memory capacity of a reservoir setup by enabling the training algorithm to utilize the memory of the individual reservoirs more efficiently. We suspect this effect to be related with the application of ridge regression in the complex domain which will be subject to future research.

VI. CONCLUSION

Tables I and II summarize the error rates obtained for all performed tasks at 30 Gbps in the electrical and optical domain respectively. Error rates printed in bold face indicate the best performing approach per task, error rates in italic the second best. From the tables above as well as Figs. 6, 7, and 8 one can see that our proposed chaining approach performs consistently well regardless of coupling type or task, and is only outperformed by ensembling on more nonlinear tasks which require low memory in the electrical domain. This stems from the fact that chaining is a simple architecture which can be trained in a straightforward way but nevertheless introduces additional richness to the setup when compared to single-reservoir architectures with the same total number of nodes. Ensembling on the other hand has shown excellent results on high nonlinearity/low-memory tasks. This is likely due to the ensemble of reservoirs exhibiting a richer reservoir state matrix in comparison to the state matrix of a baseline reservoir with an identical number of nodes (i.e. 4 times the number of nodes of a single reservoir in an ensemble of 4 reservoirs). Therefore ensembling is a simple robust method which outperforms chaining, stacking and boosting on short-memory tasks. Boosting and stacking have shown only small improvements on the nonlinear tasks, but still both manage to outperform the baseline on memory-intensive tasks. Nevertheless we consider boosting more valuable than stacking since it manages to deliver consistent improvements over all tasks/coupling modes, while stacking has shown to perform at times worse as more reservoirs are added. Based on these observations, for future hardware implementations, we recommend the application of a chaining architecture in order to leverage the performance of fully optical passive photonic reservoir computing systems. For reservoir systems trained in the electrical domain, ensembling is a very efficient, cheap alternative. As a concluding remark we would like to emphasise that while outperforming single large

reservoirs by efficiently combining a number of smaller ones is desirable, merely matching the performance of those larger reservoirs is of value as well: in the technologies we are currently investigating [32] a combination of smaller reservoirs is easier to construct than a single larger reservoir. For future work we will focus on transferring the computational power and insights of deep neural networks into the optical hardware domain in order to fully exploit the potential of cascaded photonics reservoir computing on chip.

APPENDIX A OPTICAL CIRCUIT SIMULATION

We simulate the coherent optical circuit of the reservoir as a graph in the complex domain where waveguide and combiner losses are modelled along the graph's edges and the multimode interferometers constitute the nodes of the graph. In order to take the strong variability of passive photonic reservoir circuits into account, we model the phase modulation of each waveguide connection to be random following a uniform distribution in $[0, 2\pi]$. This graph is represented as a complex-valued adjacency matrix. While the equivalence of the graph adjacency matrix to the reservoir connection matrix is obvious, all reservoir state signals are evaluated in continuous time. This is done by transferring the graph into a set of ordinary differential equations as described at length in [29]. This set of ordinary differential equations is then solved using numerical methods in order to obtain all signals occurring at the reservoir nodes in the modelled circuit.

APPENDIX B DETAILS ON PHOTODETECTOR MODEL

To simulate an integrated photodetector for the electrical and optical readouts, we apply the photodetector model proposed in [18]: we compute the electric current of a sampled complex-valued signal \mathbf{x} as

$$\mathbf{i}(\mathbf{x}) = R|\mathbf{x}|^2, \quad (3)$$

where R is the photodetector's responsivity. Thereafter a zero-mean Gaussian noise vector \mathbf{n} with a variance σ_n^2 is added to $\mathbf{i}(\mathbf{x})$. The variance σ_n^2 is computed as

$$\sigma_n^2 = 2qB(\langle I \rangle + \langle I_d \rangle) + 4k_B T B / R_L \quad (4)$$

where q is the elementary particle charge, B is the bandwidth of the photodetector, $\langle I \rangle = \frac{1}{N} \sum_n i[n]$ is the photocurrent, $\langle I_d \rangle$ is the dark current, k_B is the Boltzmann constant, T is the temperature and R_L is the load impedance of the photodetector. We set $R = 0.5 \frac{\text{A}}{\text{W}}$, $B = 70 \text{ GHz}$, $\langle I_d \rangle = 0.1 \text{ nA}$, $T = 300 \text{ K}$ and $R_L = 1 \text{ M}\Omega$ in all our simulations. Finally, to model the limited bandwidth B of the integrated optical detector, a fourth-order Butterworth low-pass filter is applied to the resulting output signal.

APPENDIX C

GENERATION ON RANDOM BIT SEQUENCES FOR HEADER RECOGNITION

As we perform the detection of a 5 bit header, this implies that we need sufficient data to train our classifiers. Since the probability of a single 5 bit header occurring in a 10000 bit long sample is rather low, this implies a heavy class-imbalance in our training data with only very little positive samples. Using longer sequences of bits improves this problem only by a small degree since the data are still equally unbalanced. Therefore we choose a different path and generate our bit sequences in the following way: we generate a random bit sequence, where we randomly draw bits from a uniform distribution, but insert additional instances of the desired header pattern h_{desired} (for instance: 10101) based on a Poisson random variable. Every time before we generate a random bit, we note the number M_{actual} of occurrences of the desired header pattern h_{desired} in the bit sequence generated so far. Further, we compute M_{desired} , the number of desired header patterns assuming that 5% of all bits end in such a desired pattern and that they are evenly distributed over the sequence as

$$M_{\text{desired}} = N_{\text{current}} \cdot 0.05 \quad (5)$$

where N_{current} is the current length of the already generated bit sequence. Based on M_{actual} and M_{desired} , we compute the probability p that instead of a random bit, the desired header pattern h_{desired} will be injected into the bit sequence. p is computed as

$$p = 1 - P(x \leq M_{\text{actual}}, \lambda = M_{\text{desired}}), \quad (6)$$

where $P(x, \lambda)$ is the cumulative distribution function of the Poisson distribution. Thus, the smaller the probability of the generated sequence to contain M_{actual} or less instances of h_{desired} , the larger the probability p that instead of a single, uniformly random bit will be drawn, the desired sequence will be inserted into the bit sequence. Finally, a test on the probability p is performed to determine whether a new random bit or the desired header will be appended at the end of the bit sequence. This procedure is repeated until the bit sequence contains the predetermined amount (in this work: 10000) bits. This procedure allows us to control the number of desired patterns in a header while still generating random bit sequences.

APPENDIX D

INPUT AND READOUT NODE CONFIGURATION FOR 4×8 AND 8×16 RESERVOIRS

For the 4×8 building block reservoir, we inject inputs into nodes number 1, 2, 12, 13, 18, 19, 27, 28, 29 and 30, where nodes are numbered row by row and left to right. For the 8×16 baseline reservoir, we inject inputs into the nodes with the same input indices as well as nodes with the input indices that correspond to the original nodes multiplied with 2, 3, and 4. This results in nodes 1, 2, 3, 4, 6, 8, 12, 13, 18, 19, 24, 26, 27, 28, 29, 30, 36, 38, 39, 48, 52, 54, 56, 57, 58, 60, 72, 76, 81, 84, 87, 90, 108, 112, 116 and 120 being input nodes to the baseline reservoir.

REFERENCES

- [1] W. Haensch *et al.*, "Silicon CMOS devices beyond scaling," *IBM J. Res. Develop.*, vol. 50, no. 4.5, pp. 339–361, 2006.
- [2] W. Maas, T. Natschlaeger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [3] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Sci.*, vol. 304, no. 5667, pp. 78–80, 2004.
- [4] D. Verstraeten, B. Schrauwen, and D. S. M. D'Haene, "An experimental unification of reservoir computing methods," *Neural Netw.*, vol. 20, no. 3, pp. 391–403, 2007.
- [5] J. Bürger and C. Teuscher, "Variation-tolerant computing with memristive reservoirs," in *Proc. IEEE/ACM Int. Symp. Nanoscale Architectures*, 2013, pp. 1–6.
- [6] E. Demis *et al.*, "Atomic switch networks-nanoarchitectonic design of a complex system for natural computing," *Nanotechnol.*, vol. 26, no. 20, 2015, Art. no. 204003.
- [7] M. Hermans, M. Soriano, J. Dambre, P. Bienstman, and I. Fischer, "Photonic delay systems as machine learning implementations," *J. Mach. Learn. Res.*, vol. 16, pp. 2081–2097, 2015.
- [8] G. V. der Sande, D. Brunner, and M. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.
- [9] L. Larger *et al.*, "Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing," *Opt. Express*, vol. 20, no. 3, pp. 3241–3249, 2012.
- [10] A. Smerieri *et al.*, "Towards fully analog hardware reservoir computing for speech recognition," in *Proc. AIP Conf. Proc.*, 2012, pp. 1892–1895.
- [11] M. C. Soriano *et al.*, "Optoelectronic reservoir computing: Tackling noise-induced performance degradation," *Opt. Express*, vol. 21, no. 1, pp. 12–20, Jan 2013.
- [12] M. Fiers *et al.*, "Nanophotonic reservoir computing with photonic crystal cavities to generate periodic patterns," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 344–355, Feb. 2014.
- [13] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, "Parallel reservoir computing using optical amplifiers," *IEEE Trans. Neural Netw.*, vol. 22, no. 9, pp. 1469–1481, Sep. 2011.
- [14] J. Qin, Q. Zhao, H. Yin, Y. Jin, and C. Liu, "Numerical simulation and experiment on optical packet header recognition utilizing reservoir computing based on optoelectronic feedback," *IEEE Photon. J.*, vol. 9, no. 1, pp. 1–11, Feb. 2017.
- [15] F. Laporte, A. Katumba, J. Dambre, and P. Bienstman, "Numerical demonstration of neuromorphic computing with photonic crystal cavities," *Opt. Express*, vol. 26, no. 7, pp. 7955–7964, 2018.
- [16] A. Lugnan, J. Dambre, and P. Bienstman, "Integrated pillar scatterers for speeding up classification of cell holograms," *Opt. Express*, vol. 25, no. 24, pp. 30 526–30 538, 2017.
- [17] K. Vandoorne *et al.*, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature Commun.*, vol. 5, 2014, Art. no. 3541.
- [18] A. Katumba, M. Freiberger, P. Bienstman, and J. Dambre, "A multiple-input strategy to efficient integrated photonic reservoir computing," *Cogn. Comput.*, vol. 9, no. 3, pp. 307–314, 2017.
- [19] S. Sackesyn, C. Ma, J. Dambre, and P. Bienstman, "An enhanced architecture for silicon photonic reservoir computing," in *Proc. Cogn. Comput. 2018: Merg. Concepts with Hardware*, 2018.
- [20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [21] M. Freiberger, A. Katumba, P. Bienstman, and J. Dambre, "Training passive photonic reservoirs with integrated optical readout," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1943–1953, Jul. 2019.
- [22] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*, vol. 1, no. 10. New York, NY, USA: Springer, 2001.
- [23] L. Keuninckx, "Electronic systems as an experimental testbed to study nonlinear delay dynamics," Ph.D. dissertation, Vrije Universiteit, Brussel, Belgium, 2016.
- [24] S. Nichele and A. Molund, "Deep reservoir computing using cellular automata," 2017, *arXiv:1703.02806*.
- [25] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.
- [26] L. Mason, J. Baxter, P. L. Bartlett, and M. R. Frean, "Boosting algorithms as gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 512–518.

- [27] S. Basterrech, "An empirical study of the 12-boost technique with echo state networks," 2015, *arXiv:1501.00503*.
- [28] M. Fiers, T. V. Vaerenbergh, K. Caluwaerts, J. Dambre, and P. Bienstman, "Caphe: Time-domain and frequency-domain modeling of nonlinear optical components," presented at the Adv. Photon. Congr., 2012, Paper IM2B.3.
- [29] M. Fiers *et al.*, "Time-domain and frequency-domain modeling of nonlinear optical components at the circuit-level using a node-based approach," *J. Opt. Soc. Am. B*, vol. 29, no. 5, pp. 896–900, May 2012. [Online]. Available: <http://josab.osa.org/abstract.cfm?URI=josab-29-5-896>
- [30] A. S. Weigend and N. A. Gershenfeld, "Results of the time series prediction competition at the santa fe institute," in *Proc. IEEE Int. Conf. Neural Netw.*, 1993, pp. 1786–1793.
- [31] M. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE J. Sel. Areas Commun.*, vol. SAC-2, no. 1, pp. 153–170, 1984.
- [32] A. Katumba *et al.*, "Neuromorphic computing based on silicon photonics and reservoir computing," *IEEE J. Sel. Topics Quantum Electron.*, vol. 24, no. 6, Dec. 2018, Art. no. 8300310.

Matthias Freiburger (S'17) was born in Graz, Austria, in 1983. He received the M.Sc. degree in information and computer engineering from the Graz University of Technology, Graz, Austria, in 2016. He is currently working toward the Ph.D. degree with IDLab, Department of Electronics and Information Systems, Ghent University-imec, Ghent, Belgium. His current research focuses on scaling up neuromorphic systems. His research interests include recurrent neural networks on chip and deep learning.

Stijn Sackesyn was born in Ghent, Belgium, in 1994. He received the M.Sc. degree in photonics from Ghent University, Ghent, Belgium and the Free University of Brussels, Brussels, Belgium, in 2017. He is currently working toward the Ph.D. degree in photonics engineering with the Photonics Research Group, Ghent University-imec, Ghent, Belgium. His research interests include photonic neuromorphic signal processing for optical communications networks.

Chonghuai Ma was born in Shenyang, China, in 1991. He received the B.Sc. degree in physics from Nankai University, Tianjin, China, in 2013, and the M.Sc. degree in photonics from Friedrich Schiller University Jena, Jena, Germany, in 2016. He is currently working toward the Ph.D. degree with the Photonics Research Group, Department of Electronics and Information Systems at Ghent University-imec, Ghent, Belgium. His research focuses on integrated photonic neuromorphic computing systems. His current research interests include architectures and training algorithms for neuromorphic devices, photonics integrated circuits design, and quantization in machine learning.

Andrew Katumba (S'14) was born in Masaka, Uganda, in 1985. He received the M.Sc. degree in optics and photonics from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2013, and the Ph.D. degree in photonics engineering from Photonics Research Group, Ghent University-imec, Ghent, Belgium, in 2019. His research interests include photonic neuromorphic architectures for high speed optical telecommunications systems. He is a Student Member of IEEE Photonics Society and the International Society for Optics and Photonics.

Peter Bienstman (M'97) was born in Ghent, Belgium, in 1974. He received a degree in electrical engineering from Ghent University, Ghent, Belgium, in 1997, and the Ph.D. degree in 2001 from the Department of Information Technology (INTEC), Ghent University, Ghent, Belgium, where he is currently a Full Professor. His research interests include several applications of nanophotonics (biosensors, photonic information processing, etc.) as well as nanophotonics modeling. He has authored/coauthored more than 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.

Joni Dambre was born in Ghent, Belgium, in 1973. She received the M.Sc. degree in electronics engineering and the Ph.D. degree in computer science engineering from Ghent University, Ghent, Belgium, in 1996 and 2003, respectively. She is currently a Full Professor with IDLab, Department of Electronics and Information Systems, Ghent University-imec. She performs research on machine learning and neural networks, with applications in sensor processing and robotics, as well as innovative unconventional and neuromorphic computing hardware that exploit the computational power of physical dynamical systems.