Experimental Demonstration of 4-Port Photonic Reservoir Computing for Equalization of 4 and 16 QAM Signals

Sarah Masaad, Stijn Sackesyn, Stylianos Sygletos, and Peter Bienstman

***Abstract*— We experimentally demonstrate the application of a passive 16-node photonic reservoir for analogue, hardware-based equalization of coherently modulated signals at 28 Gbaud. This integrated photonic network, termed the 4-port reservoir, replaces computationally expensive digital signal processing (DSP) procedures for both fiber impairment equalization, including chromatic dispersion, as well as for transceiver imbalance equalization. For full mitigation of transmission impairments, our photonic solution can seamlessly integrate with DSP blocks for frequency offset compensation and blind phase search, achieving bit error rates on-par with the legacy DSP blocks it replaces. The same reservoir is shown to successfully equalize both 4 and 16 QAM signals in a range of linear and nonlinear transmissions.**

***Index Terms*— Photonic Reservoir, Signal Equalization, Coherent Receiver, 4 QAM, 16 QAM, Analogue Computing, Photonic Computing, Dispersion Compensation**

I. INTRODUCTION

Coherent communication has been a key enabler for increasing transmission throughputs in long-haul communications without taxing the limited bandwidths of electronic interfaces, since it allows multilevel modulation over two orthogonal quadratures. However, the expensive hardware of a coherent receiver, as well as the power-hungry digital signal processing (DSP) procedures that are required post-detection, remain hindering for high-baudrate implementations and/or for mass deployment of coherent transponders in short- and mid-reach systems [1]. Such systems include inter- and intra-data center communications, where stringent power and cost requirements exist due to the large number of transceivers required. This has for long imposed the use of intensity modulation and direct detection (IM/DD) systems, which are cheaper to deploy and operate. More recently, research advances supported the upgrade from standard IM/DD to 4-level pulse amplitude modulation (PAM 4), thus increasing the spectral efficiency while maintaining compatibility with IM/DD technologies [2]. However, supporting the ever-increasing data rate cannot indefinitely be sustained through modulating on a single dimension [3]. As such, coherent transceivers that exploit more degrees of freedom can be suitable solutions for future short-reach systems when technologies reducing both transceiver costs and power consumption are mature [4]. To this end, common research directions investigate reduction in receiver complexity through alternative architectures, as well as more efficient methods for the required signal processing procedures post detection. Efforts pertaining to the former theme include self-coherent receiver architectures like the Kramers-Kronig receiver [5] and the Gerchberg-Saxton receiver [6], as well as homodyne coherent receivers [7]. For the latter, alternative solutions to standard DSP methods like analogue-based solutions can circumvent the need for analog-to-digital converters (ADCs) and operate at lower bandwidths than their digital counterparts. Both electronic [8], [9] and optical analogue solutions [10]–[13] can be utilized.

In this work, we experimentally demonstrate the use of optical signal processing to relax electronic processing needs of coherent transceivers deploying 4 and 16 quadrature amplitude modulation (QAM) formats. This not only allows harnessing the large bandwidths of optics for processing signals, but also naturally supports one-shot processing of both quadratures. The processor used is a photonic integrated network based on a machine learning paradigm known as reservoir computing. Our reservoir architecture can perform equalization tasks and therefore replaces digitally implemented equalizers for chromatic dispersion and transceiver impairments. Moving these steps, which normally operate at twice the baud rate, into the optical domain considerably reduces the computational complexity of the pipeline and the bandwidth of the electronics. In these experiments we pair our network with standard DSP blocks performing other necessary procedures including frequency offset compensation and blind phase search, thus showcasing that our network can seamlessly integrate with existing DSP pipelines. This shows that gradual migration into an optical co-processing solution is supported, before moving to fully optical or optical-analogue hybrids.

The rest of the paper is organized as follows: Section II introduces reservoir computing and describes the architecture

Parts of this work were performed in the context of the European projects Postdigital (GA860360), Nebula (GA871658), Neoteric (GA871330), Prometheus (GA101070195), and the Belgian FWO (G006020N) and EOS (40007536) projects.

Sarah Masaad, Stijn Sackesyn, and Peter Bienstman are with Photonics Research Group, INTEC-department, Ghent University-imec, Belgium sarah.masaad@ugent.be

Stylianos Sygletos is with Aston Institute for Photonic Technologies, Aston University, United Kingdom

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

of the photonic reservoir used in this experiment. Next, Section III details the experimental setup and section IV discusses the processing of the collected data. Section V presents and discusses the achieved results and the paper is finally concluded in Section VI.

II. PHOTONIC RESERVOIR COMPUTING

Integrated photonics offers the capability of miniaturizing and integrating various photonic components onto a compact chip. This facilitates the generation, modulation, processing, and detection of light with minimal footprint [14]. By utilizing this technology, various approaches, including those based on machine learning, are currently explored in order to enable light-based computing in compact circuits. This allows using light's large bandwidths, low-loss channels, and multiplexing capabilities for performing computational tasks.

To this end, a prominent framework, known as reservoir computing (RC), has been successfully used for computing with light in a plethora of machine learning and signal processing applications [15], [16]. An RC network consists of a number of nodes linked with weighted interconnects, similar to a recurrent neural network. However, the network's inner connections are random and untrained. Only the output weights, in what is known as the readout, are trainable. This offers an efficient way to construct neural networks in physical hardware where manipulating internal weights is either expensive or inherently infeasible.

Our implementation of an integrated photonic reservoir, which we term the 4-port architecture [17], is based on a silicon nitride (SiN) platform and utilizes 3x3 multimode-interferometers (MMIs) as nodes. As seen in Figure 1, two of a node's input ports come from other reservoir nodes and two of its output ports connect to other nodes. The third input port allows injecting an external signal into the reservoir, but can also be left unconnected such that selected nodes can be chosen as inputs. The third output port leads to the readout where complex-valued trainable weights are implemented.

The fixed weights inside the reservoir originate from the waveguide interconnects, which induce phase and (to a lesser extent) amplitude modulation on the propagating signals. Due to inherent fabrication imperfections, such as sidewall roughness, each fabricated reservoir has a unique, yet fixed, set of internal weights. Furthermore, these waveguides are designed as spiraled connections such that they induce meaningful delays to the optical signals before they are mixed in the nodes. This increases the memory of the system and allows solving memory-intensive tasks like dispersion compensation. From simulations, these waveguides were shown to have optimal performance when they introduce time delays corresponding to half the symbol period. As such, most of the connections were designed with this criterion.

This reservoir's architecture is passive, as it does not require external power to drive the nodes or connections. Furthermore, non-volatile optical weights can also be integrated on the chip. While these weights require initial power to be programmed, once set they hardly consume any electrical power.

Note that it is possible to use the reservoir architecture with or without the integrated optical weights. This is depicted in Figure 1, which shows two methods for implementing the readout. In the parallel readout, shown in red, the readout signals from the reservoir are directed to the integrated and optically implemented weights, followed by an integrated summation tree. Finally, the summation tree output is led to a receiver. Here, optical weights are iteratively adjusted during the training phase until the weighted and summed output detected by the receiver closely resembles the target signal. Then the trained weights are used in the inference phase.

The optical weights themselves are commonly implemented using controllable phase shifters and multi-mode interferometers. Through shifting the phase of one signal with respect to the other, constructive or destructive summation occurs, which allows controlling the amplitude of the signal. Another phase shifter can also be used to control the resulting signal's phase, thus allowing the implementation of complex weights. This controllable optical phase shift effect can be implemented in several ways, including tungsten heaters, micro-electro mechanical systems (MEMS), or phase change material like barium-titanate (BTO) [18].

If a parallel readout cannot be implemented, a serial readout, shown in blue, is an alternative for proof-of-concept lab experiments. Note that the entire reservoir would follow only a single readout scheme, however they are both depicted on the same figure for conciseness. In the serial readout, each readout channel is measured separately and sequentially and is then stored offline. Through offline processing, appropriate weights can be computed and then tested on the collected data. However, a serial readout remains different from a delay-based reservoir, since sampling is done over multiple spatially separate nodes as opposed to the single node used in delay-based reservoirs [19].

A serial readout is not typically preferred since measuring nodes sequentially adds processing complexity (as will be seen later) and prohibits online utilization of the reservoir. However, it can be used as a temporary fallback to bypass issues

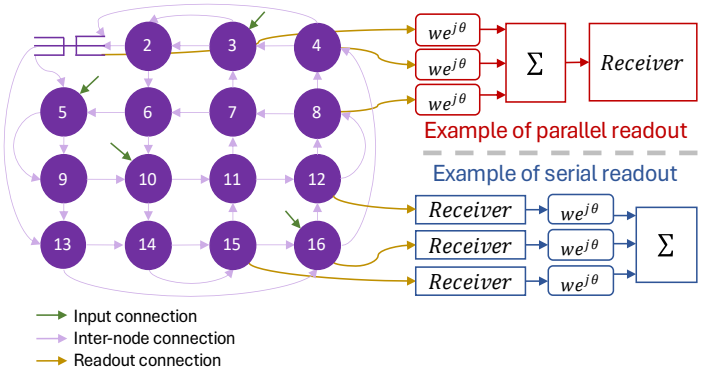

Fig. 1 16-node reservoir with 2 possible readout schemes. Nodes are shown as numbered vertices, with the exception of node 1, which is shown as an abstracted 3x3 MMI. Input, internode, and readout connections are shown as colored arrows. A parallel readout scheme is shown in red which includes on-chip weights and one detected signal. A serial readout scheme is shown in blue, where electronic off-chip weights are implemented after individual detection of readout states.

associated with excessive optical losses. This was the case with the generation of the 4-port reservoir used in this experiment, where around 8 dB losses per coupler were incurred when coupling in and out of the reservoir chip. Additional component losses were also incurred, including insertion losses in the MMIs and waveguide bend losses. These accumulated losses were prohibitive in this experiment to utilize the parallel readout scheme because utilizing the integrated readout would introduce additional optical combining losses on top of those already incurred. This would place the detected signals below the noise floor of the receiver. In the absence of excessive losses, the integrated readout becomes viable and is indeed investigated in a newer chip generation where the coupling mechanism is changed for a more efficient one.

Extensive simulations and experiments have demonstrated the versatility of this architecture in tackling applications in fiber-optic communication systems. For example, experimental equalization was demonstrated in [20] for intensity modulated and direct detected signals. In [21], the same set of readout weights was experimentally shown to equalize signals of multiple wavelengths, being a first step towards the applicability of this reservoir in wavelength division multiplexed systems. In this paper, we experimentally show that the 4-port reservoir can successfully equalize 4 and 16 QAM signals in the presence of chromatic dispersion and transceiver impairments, matching the performance of legacy DSP.

III. EXPERIMENTAL SETUP

The experimental setup is shown in Figure 2 and the system parameters are summarized in Table 1. On the transmitter side, a dual-polarization IQ modulator was used to modulate a continuous-wave (CW) laser emitting at 1550 nm by a randomly generated data stream. The data stream was composed of either 4 or 16 QAM symbols pulse shaped using a root-raised cosine filter with a roll-off of 0.2 and transmitted at 28 Gbaud. The output of the transmitter was amplified by Amplifier 1 to 23 dBm, and then attenuated by a variable optical attenuator that controlled the desired input power into the fiber. This setup allowed sweeping the fiber's input power without changing the noise profile of the amplifier. The signals then traversed 20 km of standard single-mode fiber.

Table 1 System parameters

Parameter	Value
Carrier Wavelength	1550 nm
Modulation Format	4 / 16 QAM
Pulse Shape	Root Raised Cosine
Roll-Off Factor	0.2
Fiber Input Power	2 – 15 dBm
Fiber Length	20 km
Receiver Input Power	0 dBm
Sampling Rate	64 Gsamples/sec

As discussed in the previous section, significant losses were incurred in this 4-port reservoir generation, ranging between 35 dB and 50 dB per node, and thus further amplification before detection was needed. This amplification was divided between Amplifier 2 (before the reservoir) and Amplifier 3 (after the reservoir) such that the input power to the receiver was maintained at 0 dBm. All amplifiers were erbium-doped fiber amplifiers.

The receiver used was a dual-polarization coherent receiver with transimpedance amplifiers. The detected signals were then sampled by a real-time oscilloscope (RTO) at 64 Gsamples/sec and stored on a computer for offline processing.

Despite the dual-polarization setup, only one polarization was used for transmission. This is due to the polarization selectivity of the SiN chip where the grating couplers and waveguides on chip are highly lossy for TM waves. Therefore, the data stream was loaded with zeros on the IQ arms of the Y (or TM) polarization. The Y polarization arm was still biased at the null point to supress the carrier, thus leaving the output of the Y polarization arm on the transmitter's side negligible.

Nonetheless, polarization rotations that occur throughout the system cause the signal to arrive at the chip's input with a random state of polarization (SOP). As such, a polarization controller was used to align the input signal with the TE polarization by maximizing the output power from the reservoir. If the signal was not polarization rotated before coupling into the chip, the power losses from the loss of the TM

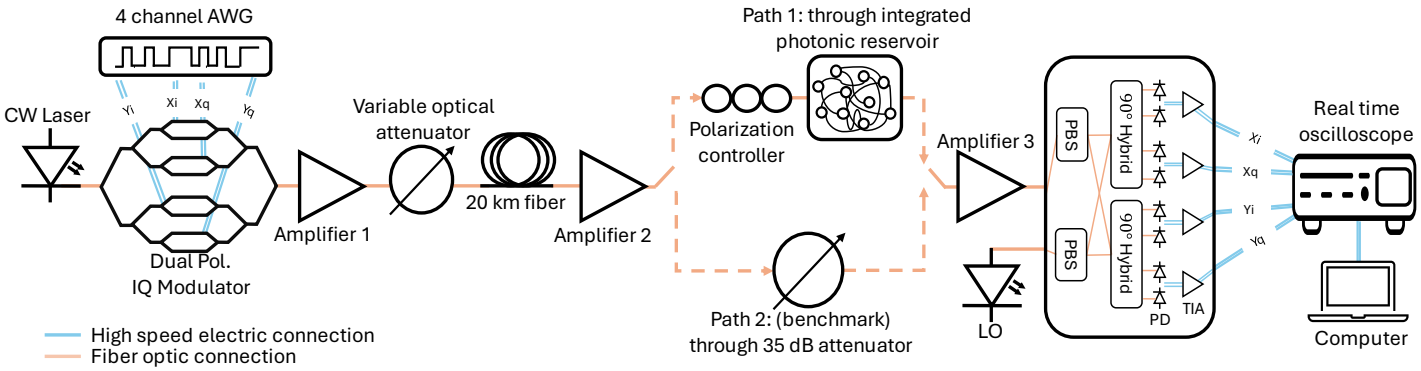

Fig. 2 Experimental setup with two possible paths: path 1 goes through the reservoir and a polarization controller, which is used to collect readout states on a node-by-node basis. Path 2 is a benchmark which replaces the reservoir with a 35 dB attenuator. All other components shown are common to both paths. CW: continuous-wave laser, AWG: arbitrary waveform generator, LO: local oscillator, PBS: polarizing beam splitter, PD: photodiode, TIA: transimpedance amplifier

polarization may have been prohibitive. At the receiver's end, the signal also arrived with a random SOP due to rotations, and was detected on both polarization branches of the receiver. Signals from the output of each reservoir node were sequentially measured and stored for offline processing.

To assess the equalization of the reservoir, its performance was compared to a benchmark system that relied solely on a DSP pipeline. This benchmark setup used a 35 dB attenuator instead of the reservoir such that the same amplifier signal-to-noise ratio (SNR) behaviour was maintained for both systems. As shown in Figure 2, measurements on the reservoir and the benchmark were done consecutively, by connecting either the attenuator or the reservoir to the setup. The benchmark data collected was also processed offline. The details of these processing steps as well as those for the RC system are discussed in the next section.

IV. OFFLINE PROCESSING

A. Processing Benchmark Data

The DSP pipeline used for the benchmark system is shown in Figure 3a [22]. Steps 1 through 7 constitute the main pipeline, with an alternative path through steps 6* and 7*. A discussion of the details and relevance of this alternative path is deferred to subsection III.D.

In the main DSP pipeline, two equalizers are deployed. The first is an ideal chromatic dispersion compensating (CDC) filter used to compensate for the incurred dispersion. The second is an active equalizer that has a 2x2 multiple-input multiple-output (MIMO) architecture used to equalize residual linear impairments, including transceiver imperfections, as well as polarization demultiplexing. Furthermore, in long-haul transmission, this equalizer is used for polarization mode dispersion compensation. For its implementation four 17-tap finite-impulse response (FIR) filters are used, two for each polarization. The two MIMO outputs obtained are given by [23]

$$y1[k] = \boldsymbol{w}'_{1X}[k]\boldsymbol{X}_X[k] + \boldsymbol{w}'_{1Y}[k]\boldsymbol{X}_Y[k] \quad (1)$$

$$y2[k] = \boldsymbol{w}'_{2X}[k]\boldsymbol{X}_X[k] + \boldsymbol{w}'_{2Y}[k]\boldsymbol{X}_Y[k] \quad (2)$$

where $\boldsymbol{X}_X[k]$ and $\boldsymbol{X}_Y[k]$ are matrices sized $17 \times N$ that originate from the FIR filters, with N being the number of samples. The subscripts X and Y denote which polarization gave rise to the respective matrix. Four 17-element weight vectors multiply these matrices, namely $\boldsymbol{w}_{1X}$, $\boldsymbol{w}_{2X}$, $\boldsymbol{w}_{1Y}$, and $\boldsymbol{w}_{2Y}$. The subscripts 1 and 2 indicate which output this weight vector will yield, while the subscripts X and Y indicate which polarization matrix it will multiply. Outputs $y1[k]$ and $y2[k]$ are the two polarization demultiplexed and linearly compensated signals.

Since both $\boldsymbol{X}_X$ and $\boldsymbol{X}_Y$ carry data from a single transmitted polarization, with proper initialization, the polarization demultiplexing results in weights $\boldsymbol{w}_{2X}$ and $\boldsymbol{w}_{2Y}$ converging to yield an output $y2[k]$ that is 0. For output $y1[k]$, $\boldsymbol{w}_{1X}$ and $\boldsymbol{w}_{1Y}$ converge to yield a linearly compensated signal.

To find these weight vectors, a Constant Modulus Algorithm (CMA) is used for 4-QAM. For 16-QAM, CMA is first applied in a pre-convergence stage, followed by a Radius Directed Equalization (RDE) algorithm [24]. These are iterative and blind optimization algorithms that attempt to minimize the radial distance between the distorted signal and the most likely radius from a 4- or 16- QAM alphabet.

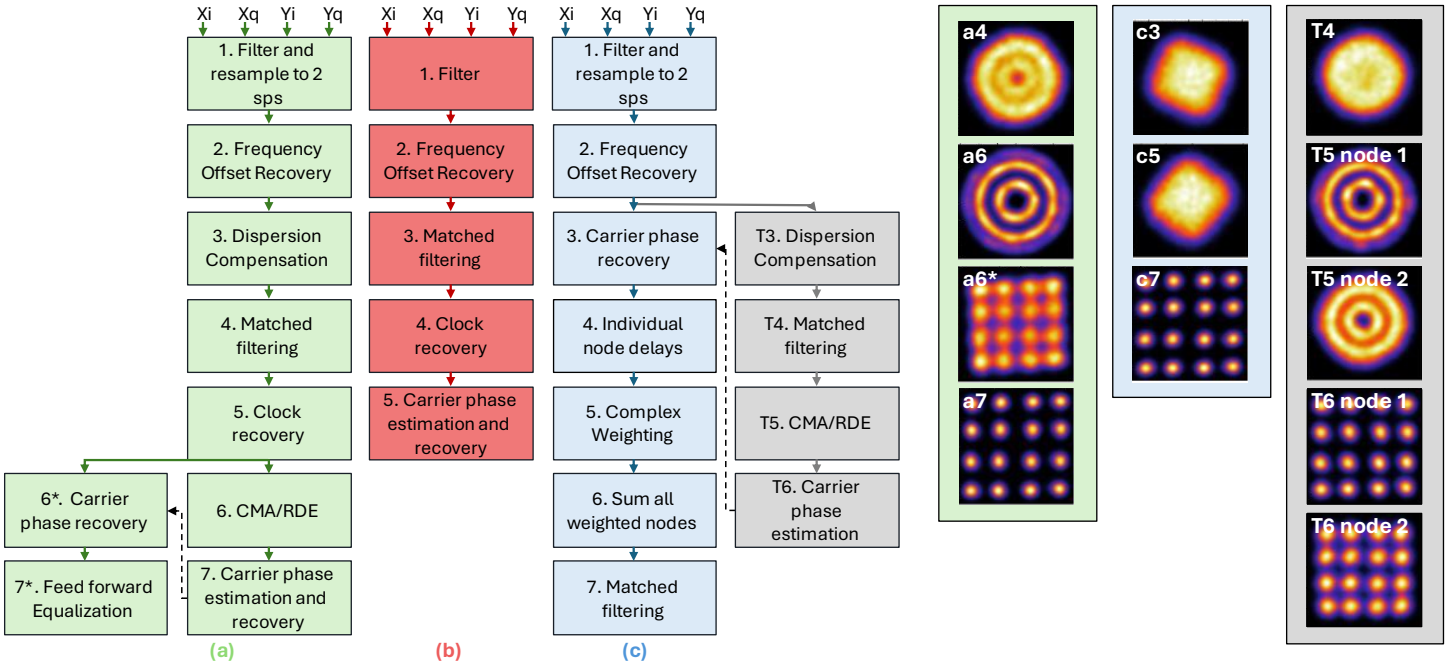

Fig. 3 Signal processing pipelines of [a] benchmark system, [b] RC with parallel readout, [c] RC with serial readout. On the right, constellation diagrams after key steps are shown, where the first letter refers to the pipeline and the number refers to the step. Blocks around constellation diagrams are also color coded to match the pipeline they originated from.

The rest of the pipeline contains DSP procedures used to track and compensate for time-varying issues. Namely, the frequency offset between the continuous wave (CW) carrier and local oscillator (LO) is found through raising the signal spectrum to the fourth power [22]. The signal is then frequency shifted based on the generated frequency tone. Furthermore, the phase and frequency offsets of the real-time oscilloscope's (RTO) sampling clock are compensated in a clock recovery algorithm, where a Gardner timing error detector is used for estimation and an interpolator for compensation. Finally, the signals are compensated for the phase shifts of the carrier through a blind phase search algorithm (BPS) [25].

B. Processing Reservoir Data

The photonic reservoir supports the migration of expensive and optically implementable equalization blocks to the photonic realm. As such both the CDC and active equalization procedures are replaced by the photonic RC. This migration offers considerable speed and power advantages since both these operations happen in one shot, at the speed of transmission, and bypass the need for complex-valued digital operations at twice the baud rate.

On the other hand, DSP procedures correcting for the time variable errors mentioned above are still necessary. This results in an opto-electronic cointegrated solution with low DSP requirements. The pipeline in red in Figure 3b shows the remaining DSP procedures for a photonic reservoir with a parallel readout.

For a reservoir with serial readout, the sequential measurement of the readout nodes requires special handling as shown in Figure 3c. In this case, all time-varying issues need to be compensated for each node individually, since each node would have incurred different values for these issues. However, DSP procedures for these compensations are devised for input signals of certain criteria. For example, the Gardner timing error detection algorithm requires clear transitions between symbols for proper operation, and as such is typically implemented after CDC. Naturally, with readout signals that are formed from dispersed signals that interfere within the reservoir, clock recovery is very challenging. As such, this step is omitted from the pipeline, and its implication will be discussed later as part of the timing ambiguities discussion in subsection III.C.

Similarly, the carrier phase estimation by BPS requires an equalized signal before the algorithm is applied and as such is normally the last block in the chain. However, with a serial RC readout, each node would have different time-varying phase rotations and as such this must be compensated before the nodes are summed. Since these phase rotations cannot be extracted immediately from the uncompensated node data, a temporary chain of DSP blocks, greyed in Figure 3c, is implemented, where the nodes are dispersion compensated and then equalized through the same MIMO equalizer discussed in the previous subsection. This equalizer can perform linear equalization and as such it is able to undo the reservoir mixing. This allows retrieving – to some extent – a clean constellation. After the phase offset array is obtained using BPS, all the temporary DSP signals are discarded. This phase offset array is then used to compensate the unprocessed node data. We stress again that the entire pipeline of Figure 3c is only needed for the temporary fallback serial readout scheme, and not for the ultimate parallel scheme that should be facilitated by our next chip generation, which will utilize pipeline 3b.

To further clarify the role of the DSP blocks, Figure 3 also shows several 16- QAM constellation insets after key steps for the different pipelines, where the first letter refers to the pipeline and the number refers to the step. Blocks around constellation diagrams are also color coded to match the pipeline they originated from.

Insets [a4] and [T4] show signals after the matched filtering step for the standard pipeline 3a and the temporary pipeline. Constellation [a4] is to be expected at this stage of the DSP, since we can note distinct power levels, seen as data points distributed on 2 noisy radii (the third is obscured in the noise). The constellation still shows linear distortion and phase rotation but those are compensated respectively through CMA/RDE as seen in [a6] and BPS as seen in [a7]. (inset [a6*] is related to an alternative DSP pipeline that will be discussed later.)

On the other hand, constellation [T4] does not show a similar radial behaviour, which is a result of the signal mixing in the reservoir. Since the mixing is linear in nature, it can be partially undone by linear equalization using the CMA/RDE algorithms. This is seen in insets [T5] which show two different nodes after equalization. While the first node seems to have been entirely compensated, the latter still exhibits distortions indicating that the linear unmixing of the CMA/RDE algorithm is not fully successful. Nonetheless, both can then be sent to the BPS algorithm which extracts the time varying phase rotation arrays to yield constellations in [T6]. These steps are then discarded and only the resulting phase rotation information is used to compensate phase rotations in 16 node measurements. An example of the node data after phase rotation compensation is shown in inset [c3]. The data is still dispersed and linearly mixed in the reservoir, hence the distorted form is seen. However, in the absence of phase rotations, the overall shape of the constellation is not circular. The weights for the readout states are then calculated, and an example of weighted node data is shown in inset [c5]. Finally, when all weighted nodes are summed, the result is passed to a matched filter that yields the constellations in [c7].

C. Resolving Timing Ambiguities in Reservoir Data

The sequential measurement of reservoir nodes meant that several timing ambiguities were introduced. These ambiguities and their solutions are addressed in this subsection.

First, the captured data streams for each node will vary in their symbol-based starting point. While traditionally cross correlation can perform this symbol timing recovery, with node data that is dispersed and distorted this method does not work. To tackle this, we synthetically attach a long header of zeros to the data which acts as a flag for the start of the symbol stream. By aligning the node data on this header, the symbol timing ambiguity can be resolved.

The second ambiguity then arises, which concerns the alignment of the nodes with respect to one another. Since the nodes are all aligned to a common header, we lose their alignment with respect to each other. These alignments must be reintroduced before the nodes are summed together. To resolve this issue, we rely on extracting timing information from a

simulation model of the reservoir network. An impulse signal is inputted to the network, and the timing delay corresponding to when the impulse is first seen at the output of each node is captured. An example of this is shown in Figure 4, where the impulse response of 3 nodes is seen. The impulse is first seen at t = 1, 0.5, and 0 respectively. In this manner, the timing information is extracted for all nodes and then used to shift them with respect to each other.

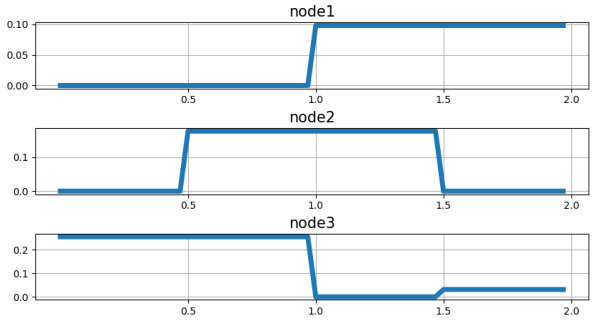

Fig. 4 Simulated impulse response of reservoir showing time alignment of the first three nodes with respect to each other. Timing information extracted from this simulation is used to resolve the relative timing ambiguity arising from sequential node measurements.

Finally, we address the timing ambiguity arising from the absence of a clock recovery algorithm. Normally, a DSP-based clock recovery procedure allows correcting for the sampling clock's frequency mismatch, phase mismatch, and jitters by tracking and correcting early or late sampling instances on a symbol-by-symbol basis. However, as mentioned earlier, this is challenging to perform for uncompensated node data. To mitigate the impact of this, we attempt to resolve the phase mismatch issue on the whole node stream by finding one optimal sampling phase per measured node. This method attempts to correct for sampling that is one sample too late or one sample too early. While this may not be ideal for every symbol in the whole data stream, the optimal sampling point is one where the least number of errors arises. Furthermore, since each node was sampled individually, the necessary offsets may be different for each node. This becomes a combinatorial problem that is unfeasible to fully explore and therefore optimization algorithms are useful.

We utilize a genetic algorithm known as Covariance matrix adaptation evolution strategy (CMA-ES) to explore this clock phase error through sample-wise delays for all 16 nodes. This is done by choosing a set of sample-wise delays whose combination minimizes the training MSE. Naturally, it may also be that no correction is needed and that the sampling phase was indeed accurate. Thus, for each node the optimization algorithm may choose a sample delay that is equal to 1, -1, or 0. In each iteration of the optimization algorithm, a set of sample shifts is tried on the nodes and then they are fitted using linear regression to compute node weights. The resulting training error is recorded and the algorithm then suggests a new set of sample-wise delays. This process is repeated for 200 iterations to find the best set.

D. Computing Weights

Finally, we address the weight computation method used to find appropriate weights for the node data. The experiment was done based on a training and testing scheme, where we transmit 95,000 symbols and partition 25,000 for training and 70,000 for testing.

An input signal matrix X was formed from the training portion of the node data. Recall that the dual polarization receiver setup would yield two signals that pertain to the same transmitted polarization. Depending on the SOP at the receiver, either of these two signals would have a better SNR. As such, they were both used in the input signal matrix, to yield a 32 by 25,000 matrix. Note that these two signals are treated exactly the same when performing the sampling frequency correction discussed in the previous section. As such, they practically only vary in their SNR, but we choose to keep both since manual selection of the polarization with SNR for each node is cumbersome. Furthermore, utilizing both polarizations matches the behaviour of the CMA/RDE algorithm, which similarly assigns different weights to each of the polarizations. The data labels for the training set, which are the symbol alphabets, form the signal target y. The weight vector W can then be calculated using a closed-form solution form

$$W = (X^H X + \alpha I)^{-1} X^H y \quad (3)$$

where α is a regularization parameter and I is the identity matrix. The superscript H refers to the conjugate-inverse of the matrix, which is needed for complex-valued data.

Note that the reservoir's limited nonlinearity resulted in a relatively simple network structure. Furthermore, the intrinsic noise introduced by the EDFAs in the setup provided an implicit regularization effect. These aspects led to the reduced need of hyperparameter tuning, which we observed in the minimal effect that changing the regularization parameter had on the results. As such, no validation set was provisioned since we did not need to compare results of different regularizations or other hyperparameters.

Computing the reservoir weights through linear regression utilizes a data-aided training algorithm, where labelled target data is used to compute the weights. On the other hand, blind optimization algorithms like CMA/RDE do not utilize such labels, but rather find ways to iteratively adjust weights through probability of symbol occurrence. These algorithms are very helpful when random phase rotations obscure the use of a target signal or when data is not labelled. However. to level the comparison ground between the DSP and reservoir pipelines, a comparable, but not standardly used, DSP alternative is devised that utilizes a data-aided training algorithm. This is shown in Fig 3a in steps 6* and 7*. Here, the same phase noise array computed in step 7 is used to compensate the signal post clock recovery. This generates the constellation seen in inset [a6*] of Figure 3, which shows a clear 16-QAM constellation but requires further equalization. To achieve that, a Feed-Forward Equalizer (FFE) is used and is composed of two FIR filters, one for each polarization, each with 16 taps to match the reservoir's 16 nodes. The FFE output is again a 32 by 25,000 matrix for the training set and is trained using the same linear regression method to compute the appropriate equalizer weights.

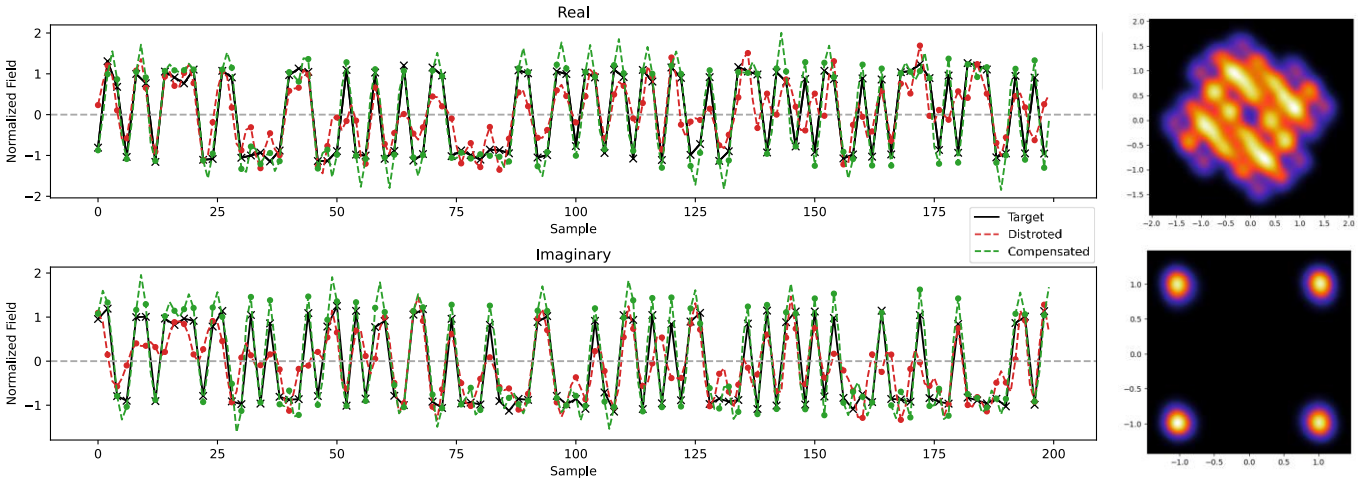

Fig. 5 Time-domain waveforms (left) and constellation diagrams (right) of a 4-QAM signal at 2 dBm input power before and after compensation by the 4-port reservoir. On the left, the top panel shows the real component while the bottom panel shows the imaginary component of the signals. Black, red, and green waveforms represent ideal, distorted, and compensated signals, respectively. The constellation diagrams on the right show the distorted signal (top) and the reservoir compensated (bottom)

V. RESULTS AND DISCUSSION

The role of the photonic reservoir is to equalize signals from fiber and transceiver impairments. To visualize the impact of this processing, Figure 5 shows a 4-QAM signal's time-domain waveforms, with the real and imaginary components plotted separately. The ideal, distorted, and compensated signals are represented by black, red, and green waveforms, respectively. To isolate the impact of the reservoir, the distorted signal was pre-compensated for frequency and phase offsets. Note that the distorted and compensated waveforms are plotted with 2 samples per symbol, and as such rigid transitions are seen. The effectiveness of the compensation is further visualized through constellation diagrams, comparing the distorted (top) and compensated (bottom) signals.

To benchmark the performance of the reservoir, Figure 6 shows the post-compensation results from the CMA/RDE DSP pipeline, the FFE pipeline, and the RC system. The BER is reported for the 16-QAM signals while the MSE is used as a metric for the 4-QAM signals since their BER is negligible. In the 16-QAM plot, the results show very similar performance for all three systems with a slight edge for the RC. Furthermore, a small advantage is seen at 13 and 14 dBm for the RC and FFE's data aided learning compared to the CMA/RDE's blind optimization. This advantage in the nonlinear regime is also clearly visible in the MSE plot for the 4-QAM signals. To further visualize this, Figure 7 shows constellation diagrams resulting from each of the three systems for a 4 QAM signal at a 15 dBm fiber input power. Since the CMA/RDE algorithm performs equalization by minimizing the distance between the radius of a distorted symbol and an ideal radius, this algorithm shows the least curbing of nonlinearities. On the other hand, both the FFE and RC minimize the distances to the symbol itself, therefore achieving better nonlinear equalization.

The reservoir, however, seems to perform worse than the benchmark pipelines in the linear regime of the 4-QAM signals. This can be attributed to the higher losses of most reservoir nodes, where many were as low as 50 dB. Recall that the benchmark setup was set to match the loss of the highest output power node and as such would have a better overall SNR

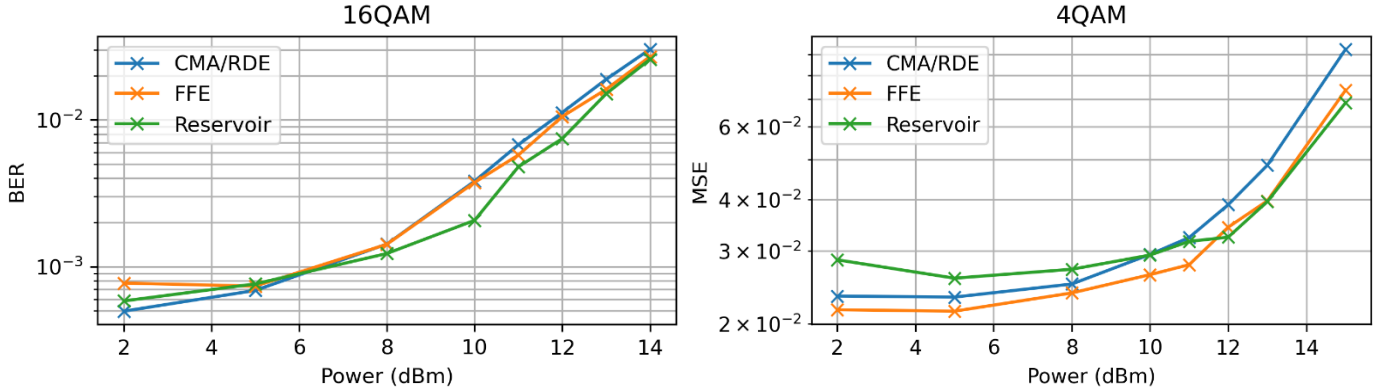

Fig. 6 Equalization results over a range of transmission powers from RC system (green) benchmarked against full DSP pipelines equalized using CMA/RDE (blue) and FFE (orange). BER results are reported for 16-QAM (left) and MSE results are reported for 4-QAM (right)

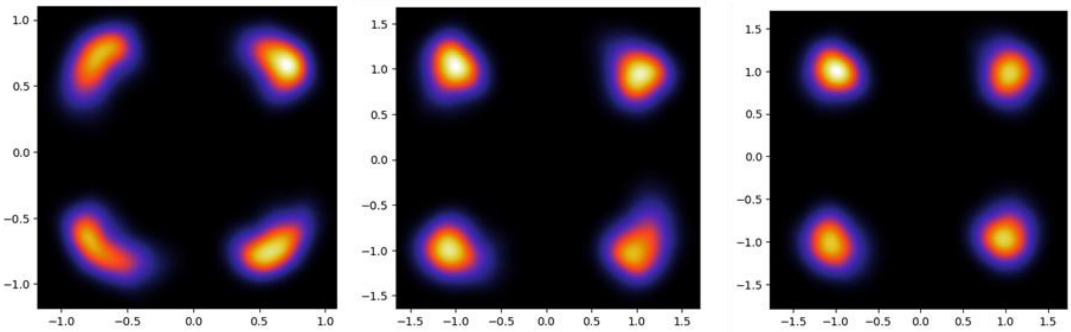

Fig. 7 4-QAM constellation diagrams for 15 dBm fiber input power after equalization in the CMA/RDE pipeline (left), FFE pipeline (center), and RC system (right)

performance. This penalty is only visible in low error scenarios, like those for the low power 4-QAM signals, where SNR becomes the main contributor. Furthermore, since MSE is the metric used, even slight noise penalties are readily observable. To verify this, Figure 8 shows the reservoir errors for the 4-QAM signal compared to the benchmark pipeline when losses of 35 dB, 45 dB, and 50 dB are introduced. For conciseness, only the FFE benchmark pipeline is shown, which is the better performing of the two DSP benchmarks. As seen, the reservoir outperforms the benchmark at 50 dB and closely matches that of the 45 dB loss.

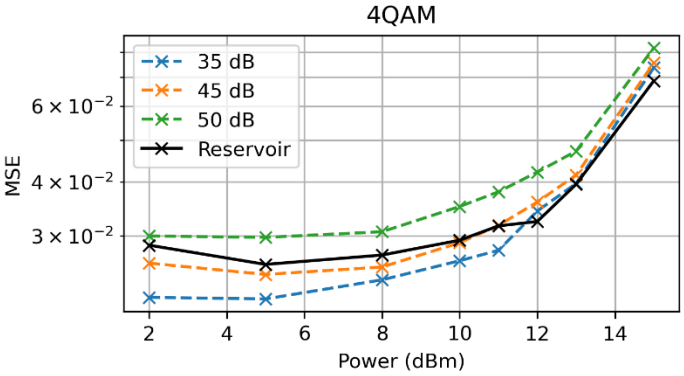

Fig. 8 Reservoir performance (black) compared to FFE benchmark pipeline when benchmark losses were 35 dB (blue), 45 dB (orange), and 50 dB (green)

Finally, we note that the choice of fiber length is common for the targeted short- and mid- reach applications, but experimental expansion to longer lengths was restricted by the equipment available at the time of the experiment. However, independent simulations showed that the reservoir can compensate for up to around 100 km of dispersion at the 28 Gbaud transmission rates. This limitation arises from the amount of memory in the reservoir, which can be expanded by adding more nodes. Other hyper-parameters, including the length of the inter-node delay lines and the addition of delay lines before the summation tree, can also be optimized to increase the memory of the network.

VI. CONCLUSION

We experimentally demonstrated the use of an integrated photonic reservoir for the equalization of 4- and 16-QAM signals at 28 Gbaud. This photonic system replaces the dispersion compensation and linear equalization blocks in the DSP pipeline, with results reported showing that it can achieve BER values on par with its DSP counterparts. Furthermore, the results show that the photonic reservoir has inherent parallelism on both quadratures and one generic chip has the programmability to address errors of different modulation formats at various linear and nonlinear transmission powers. This approach surpasses conventional computationally expensive equalization procedures, since it operates directly at the transmission speed, eliminating the need for computationally demanding high-baud-rate operations in the DSP.

ACKNOWLEDGMENTS

The authors would like to thank Mohammed Patel from AIPT for his assistance with experimental resources.

REFRENCES

[1] P. J. Winzer, D. T. Neilson, and A. R. Chraplyvy, “Fiber-optic transmission and networking: the previous 20 and the next 20 years [Invited],” *Opt. Express*, vol. 26, no. 18, p. 24190, 2018, doi: 10.1364/oe.26.024190.

[2] X. Zhou, R. Urata, and H. Liu, “Beyond 1Tb/s datacenter interconnect technology: Challenges and solutions,” *Opt. InfoBase Conf. Pap.*, vol. Part F160-, pp. 11–13, 2019.

[3] C. Xie and J. Cheng, “Coherent Optics for Data Center Networks,” *2020 IEEE Photonics Soc. Summer Top. Meet. Ser. SUM 2020 - Proc.*, pp. 10–11, 2020, doi: 10.1109/SUM48678.2020.9161052.

[4] J. K. Perin, A. Shastri, and J. M. Kahn, “Coherent Data Center Links,” *J. Light. Technol.*, vol. 39, no. 3, pp. 730–741, 2021, doi: 10.1109/JLT.2020.3043951.

[5] A. Mecozzi, C. Antonelli, and M. Shtaif, “Kramers–Kronig receivers,” *Adv. Opt. Photonics*, vol. 11, no. 3, p. 480, 2019, doi: 10.1364/aop.11.000480.

[6] H. Chen, N. K. Fontaine, J. M. Gene, R. Ryf, D. T. Neilson, and G. Raybon, “Dual Polarization Full-Field Signal Waveform Reconstruction Using Intensity only Measurements for Coherent Communications,” *J. Light. Technol.*, vol. 38, no. 9, pp. 2587–2597, 2020, doi: 10.1109/JLT.2020.2978052.

[7] T. Gui, X. Wang, M. Tang, Y. Yu, Y. Lu, and L. Li, “Real-Time Demonstration of Homodyne Coherent Bidirectional Transmission for Next-Generation Data Center Interconnects,” *J. Light. Technol.*, vol. 39, no. 4, pp. 1231–1238, 2021, doi: 10.1109/JLT.2021.3052826.

[8] J. K. Perin, A. Shastri, and J. M. Kahn, “DSP-free coherent receivers for data center links,” *2018 Opt. Fiber Commun. Conf. Expo. OFC 2018 - Proc.*, pp. 1–3, 2018.

[9] T. Hirokawa *et al.*, "Analog Coherent Detection for Energy Efficient Intra-Data Center Links at 200 Gbps per Wavelength," *J. Light. Technol.*, vol. 39, no. 2, pp. 520–531, 2021, doi: 10.1109/JLT.2020.3029788.

[10] K. Vandoorne *et al.*, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nat. Commun.*, vol. 5, 2014, doi: 10.1038/ncomms4541.

[11] S. M. Ranzini, F. Da Ros, H. Bülow, and D. Zibar, "Tunable optoelectronic chromatic dispersion compensation based on machine learning for short-reach transmission," *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204332.

[12] C. Mesaritakis, K. Sozos, D. Dermanis, and A. Bogris, "Spatial Photonic Reservoir Computing based on Non-Linear Phase-to-Amplitude Conversion in Micro-Ring Resonators," 2021, doi: 10.1364/OFC.2021.Tu1H.2.

[13] S. Masaad, S. Sackesyn, S. Sygletos, and P. Bienstman, "Experimental Demonstration of 4-Port Photonic Reservoir Computing for Equalization of 4 and16 QAM signals," in *Submitted ECOC*, 2024, no. 1, pp. 2–5.

[14] L. Thylén and L. Wosinski, "Integrated photonics in the 21st century," *Photonic Res.*, vol. 2, no. 2, pp. 75–81, 2014.

[15] A. Lugnan *et al.*, "Photonic neuromorphic information processing and reservoir computing," *APL Photonics*, vol. 5, no. 2, 2020, doi: 10.1063/1.5129762.

[16] C. Huang *et al.*, "Prospects and applications of photonic neural networks," *Adv. Phys. X*, vol. 7, no. 1, 2022, doi: 10.1080/23746149.2021.1981155.

[17] S. Sackesyn, C. Ma, A. Katumba, J. Dambre, and P. Bienstman, "A Power-Efficient Architecture for On-Chip Reservoir Computing," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11731 LNCS, pp. 161–164, 2019, doi: 10.1007/978-3-030-30493-5_16.

[18] S. Shekhar *et al.*, "Roadmapping the next generation of silicon photonics," *Nat. Commun.*, vol. 15, no. 1, 2024, doi: 10.1038/s41467-024-44750-0.

[19] G. Van Der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017, doi: 10.1515/nanoph-2016-0132.

[20] S. Sackesyn, C. Ma, J. Dambre, and P. Bienstman, "Experimental realization of integrated photonic reservoir computing for nonlinear fiber distortion compensation," *Opt. Express*, vol. 29, no. 20, p. 30991, 2021, doi: 10.1364/oe.435013.

[21] E. Gooskens, S. Sackesyn, J. Dambre, and P. Bienstman, "Experimental results on nonlinear distortion compensation using photonic reservoir computing with a single set of weights for different wavelengths," *Sci. Rep.*, no. 0123456789, pp. 1–7, 2023, doi: 10.1038/s41598-023-48816-9.

[22] M. S. Faruk and S. J. Savory, "Digital Signal Processing for Coherent Transceivers Employing Multilevel Formats," *J. Light. Technol.*, vol. 35, no. 5, pp. 1125–1141, 2017, doi: 10.1109/JLT.2017.2662319.

[23] D. A. de Arruda Mello and F. A. Barbosa, *Digital Coherent Optical Systems*. 2021.

[24] P. J. Winzer, A. H. Gnauck, C. R. Doerr, M. Magarini, and L. L. Buhl, "Spectrally efficient long-haul optical networking using 112-Gb/s polarization-multiplexed 16-QAM," *J. Light. Technol.*, vol. 28, no. 4, pp. 547–556, 2010, doi: 10.1109/JLT.2009.2031922.

[25] T. Pfau, S. Hoffmann, and R. Noé, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M-QAM constellations," *J. Light. Technol.*, vol. 27, no. 8, pp. 989–999, 2009, doi: 10.1109/JLT.2008.2010511.