

Research Article

Egor Manuylovich*, Dmitrii Stoliarov, David Saad and Sergei K. Turitsyn

Optical neuromorphic computing via temporal up-sampling and trainable encoding on a telecom device platform

<https://doi.org/10.1515/nanoph-2024-0614>

Received November 7, 2024; accepted May 28, 2025;
published online June 19, 2025

Abstract: Mapping input signals to a high-dimensional space is a critical concept in various neuromorphic computing paradigms, including models such as reservoir computing (RC) and extreme learning machines (ELM). We propose using commercially available telecom devices and technologies developed for high-speed optical data transmission to implement these models through nonlinear mapping of optical signals into a high-dimensional space where linear processing can be applied. We manipulate the output feature dimension by applying temporal up-sampling (at the speed of commercially available telecom devices) of input signals and a well-established wave-division-multiplexing (WDM). Our up-sampling approach utilizes a trainable encoding mask, where each input symbol is replaced with a structured sequence of masked symbols, effectively increasing the representational capacity of the feature space. This gives remarkable flexibility in the *dynamical phase masking* of the input signal. We demonstrate this approach in the context of RC and ELM, employing readily available photonic devices, including a semiconductor optical amplifier and nonlinear Mach–Zehnder interferometer (MZI). We investigate how nonlinear mapping provided by these devices

can be characterized in terms of the increased controlled separability and predictability of the output state.

Keywords: nonlinear mapping; reservoir computing; extreme learning machine nonlinear optical loop mirror; optical computing

1 Introduction

There is a growing interest in unconventional approaches to computing, as traditional digital computing is reaching its fundamental limitations [1], [2], particularly due to the unsustainable power consumption of machine learning approaches [3]. Reservoir computing (RC) [4], [5] and extreme learning machine (ELM) [6] are two popular unconventional (non-digital) computing concepts implicitly based on nonlinear mapping of the input into a high dimensional output where it can be processed using simple and efficient linear algorithms. For practical implementation purposes, operating only with a linear readout layer is a substantial advantage, due to the straightforward processing and low computational complexity required. The main difference between the two methods is that the RC architectures exploit recurrent connections, creating memory in the system, while ELM is a feed-forward approach that does not use any memory.

The idea of mapping input signals into a high-dimensional output for processing is rooted in the foundations of information theory, but also features in machine learning approaches, such as, support vector machines (SVM) [7] (with predetermined nonlinear mapping) and reservoir computing, which uses a recurrent neural network as an uncontrolled nonlinear mapping. Difficult computing tasks are made easier by transforming them nonlinearly to a higher dimensional space where linear processing can be applied. In this approach computing is treated as structuring the nonlinear mapping, instead of processing of structures as in traditional computing. Note that creating high-dimensional space available for the mapping is

*Corresponding author: Egor Manuylovich, Aston Institute of Photonic Technologies, Aston University, Birmingham, UK,
E-mail: e.manuylovich@aston.ac.uk

<https://orcid.org/0000-0002-5722-0122>

Dmitrii Stoliarov and Sergei K. Turitsyn, Aston Institute of Photonic Technologies, Aston University, Birmingham, UK,
E-mail: d.stoliarov@aston.ac.uk (D. Stoliarov), s.k.turitsyn@aston.ac.uk (S.K. Turitsyn). <https://orcid.org/0000-0001-8635-2346> (D. Stoliarov).
<https://orcid.org/0000-0003-0101-3834> (S.K. Turitsyn)

David Saad, Aston Institute of Photonic Technologies, Aston University, Birmingham, UK; and Aston Centre for Artificial Intelligence Research and Application, Aston University, Birmingham, UK,
E-mail: d.saad@aston.ac.uk. <https://orcid.org/0000-0001-9821-2623>

not sufficient in itself, since the output signal should be effectively spread across this feature space, without making many dimensions redundant. Therefore, the important challenge is to ensure a high effective dimensionality of the output, that is quantified by a set of linearly independent variables.

Manipulating the effective dimensionality of the feature space by a nonlinear transformation of the streamed temporal signal paves the way for a range of non-conventional computing methods. In particular, it is well suited for manipulating temporal continuous (analog) signals that are naturally generated in sensing, imaging and a number of other applications. Analog information processing is one of the key pillars of unconventional computing. Analog computing which has a long history, is experiencing a resurgence due to its superior power efficiency and capability of parallel processing [8]. Analog computing is well suited for tasks that require continuous inputs and outputs. The effective dimensionality can be controlled in different ways (e.g. temporal sampling or frequency filtering) projecting the infinite-dimensional space of the analog signal after nonlinear transformation onto a well-separated set of features. Additionally, effective dimensionality can be controlled by the parameters and characteristics of the nonlinear transformation.

Analog information processing can be implemented in a variety of physical systems trained to learn complex features. In more general terms, natural, human-engineered physical, chemical, and biological systems can be used as substrates to realize computing algorithms (see, e.g. [9]–[13] and references therein).

Ultra-fast nonlinear photonic systems, in particular, are attractive for the implementation of unconventional computing approaches due to their relatively low power dissipation and capability of parallel signal processing. Recently, there has been a great deal of interest in the development of photonic-based ELMs and RCs (see, e.g., [14]–[23] and references therein). Several notable works have demonstrated the potential of photonic reservoir computing in leveraging existing telecom technologies. A 16-node square mesh reservoir on a silicon photonics chip was implemented, capable of executing Boolean operations and header recognition tasks [24]. A parallel photonic reservoir computing approach using semiconductor optical amplifiers was also investigated, demonstrating competitive performance in speech recognition tasks [25]. A unified framework for reservoir computing and extreme learning machines was developed using a single nonlinear neuron with delayed feedback, realizable in optoelectronic and all-optical implementations [26]. Another study experimentally implemented

reservoir computing with a nonlinear optoelectronic oscillator, achieving high performance on spoken digit recognition and time series prediction tasks [27]. A high-speed photonic reservoir computing system based on InGaAsP microring resonators was demonstrated for efficient all-optical pattern recognition in dispersive Fourier imaging [28].

Light possesses a rich set of degrees of freedom that can carry information. In optical communications, parameters such as amplitude, phase, wavelength, and polarization are routinely used to encode signals, with spatial division multiplexing emerging as an important technique for increasing data transmission rates further. Several examples of optical neuromorphic computing exploiting spectral signal multiplexing have already been explored, including implementations based on optical reservoir computing [19], [29], [30] as well as other optical neuromorphic architectures [20], [31]–[33]. Recent works demonstrated that WDM enhances photonic reservoir computing by enabling parallel processing and increasing computational capacity. A Fabry–Perot semiconductor laser-based RC leveraged multiple wavelength channels to improve signal equalization in optical communications [34]. A microring resonator-based RC exploited WDM for simultaneous multi-task processing, showing its potential for efficient parallel computing [35]. A waveguide-based RC demonstrated improved nonlinear signal equalization across multiple WDM channels [36]. Additionally, WDM has been shown to enhance the memory capacity of RC without requiring external optical feedback by using wavelength-multiplexed delayed inputs in microring-based architectures [37]. In this work, we show that WDM can also be used to enable a faster encoding mask, surpassing the speed limitations of single-channel implementations.

In the temporal domain, up-sampling was used in [38]. However, the advantages of a faster output sampling rate were not leveraged because of the low bandwidth of the readout system. In this work, we demonstrate the advantage of up-sampling in enhancing the performance metrics on different tasks. Also, we would like to point out that unlike the previously studied delay-based RC implementations, [38], [39], we propose approach without additional optical delay, while providing comparable processing capacity. Note that in the majority of the demonstrated optical reservoir computing and ELMs, only one of these degrees of freedom was used.

Photonic ELMs are characterized by their feed-forward architecture, which features untrained internal connections and trained output weights. In the optical domain, the

non-trainable, nonlinear signal transformation characteristic of ELM is performed automatically and cost-effectively through physical signal propagation and registration, leveraging the inherent properties of light and nonlinear optical components and systems. Photonic ELMs have been demonstrated in various setups, including a free-space optical propagation scheme [14] and frequency-multiplexed fiber framework [21]. The implementation using the array of microresonators on an integrated silicon chip [23] achieved notable success in both binary and analog tasks, underscoring the potential of ELMs in photonics for efficient and high-performance machine learning applications. The efficacy of photonic ELMs has been further enhanced by employing feedback alignment for training the input mapping, see, e.g., [40].

Recent applications of optical reservoir computing include: modulation format identification in fiber communications using single dynamical node-based photonic RC [41], machine learning based on RC with time-delayed optoelectronic and photonic systems [42], photonic neuromorphic technologies in channel equalization [43], analog optical computing for artificial intelligence [44] and many others.

In this work, we propose and demonstrate how devices and technologies developed for optical data transmission can be used for computing applications. While individual techniques like high-dimensional mapping, input masking, WDM, and specific nonlinear elements have been explored previously, our contribution focuses on a novel combination of these elements within a flexible framework designed for effective nonlinear mapping. The combined use of signal feature-space expansion via temporal up-sampling and WDM technology for parallel encoding gives a great degree of flexibility in designing the structure of a high-dimensional output. A key element of our approach is the incorporation of a trainable input encoding mask. This trainable mask provides a crucial advantage, allowing for task-specific optimization of the input signal representation before it undergoes nonlinear transformation, thereby enhancing the overall representational capacity and performance of the system. Multiple spectral channels can range from coarse WDM (systems with fewer than eight active wavelengths per fiber) to dense WDM (DWDM). DWDM can offer standard telecom solutions with a number of channels varying from tens to hundreds with typical (but also variable) channel spacing of 50 GHz or 100 GHz within the so-called optical fiber C-band (spectral interval from 1,530 nm to 1,565 nm). Traditional DWDM systems exploit

wavelength-selective switches designed with fixed 50 GHz or 100 GHz filters. Using other fiber spectral bands, DWDM can be extended to thousands of channels. In the temporal domain, data streams with symbol rates as high as 32, 64 Gbaud (and more) in a single fiber can be produced with standard components. Thus, commercially available telecom devices can be utilized to produce a huge dimensional output feature space using only standard conventional technology. Numerous non-linear optical elements, modulators, devices and systems have been developed in the context of optical communications. To illustrate our combined approach featuring this trainable encoding, we consider here a balanced-arm MZI with non-symmetric couplers, which is mathematically equivalent to nonlinear optical loop mirror (NOLM), and semiconductor optical amplifier (SOA) as nonlinear transformers of optical signal. In what follows, we use balanced-arm MZI with non-symmetric couplers and NOLM interchangeably.

2 Methods

2.1 Nonlinearity and effective dimensionality

Many machine learning methods utilize non-linear mapping as part of the data manipulation process, for instance, radial basis functions, most variants of neural networks, kernel-based methods and boosting [45]. These mappings *can be made* in a higher-dimensional space, for instance, by having hidden layers that include more neurons than the input, but these are not carried out in a controlled and structured manner and are not explicitly used as part of the processing method.

At the heart of our method is the nonlinear mapping of input vectors to a high-dimensional space, which facilitates the application of various tasks. While our primary focus is on time series forecasting and prediction, which serves as an example of a regression task, we also demonstrate the versatility of our approach by applying it to a classification problem on a sub-sampled MNIST dataset. In spirit, our methods follow the rationale of SVM [7], where both classification and regression tasks are made possible by mapping them to the high dimensional space where linear separation [46] and approximate regression can be carried out; but, unlike SVM, where the nonlinear transformation is replaced by the corresponding kernel and support vectors should be identified, we rely on the speed of computing

devices and their ability to carry out fast mapping and simple regression.

Clearly, the nature of the nonlinear mapping and its suitability for the data is a crucial but difficult aspect of the method that should be addressed. Relying on available optical telecommunication devices limits the nonlinear mapping we can utilize; nevertheless, one can apply different control parameters that govern the type of mapping achieved, as detailed in Section 2.2. Developing a principled approach for determining the optimal mapping parameters is beyond the scope of the current paper and will be the subject of future research.

For the more general case of noisy data and mapping process, it would be natural to assess the impact of the nonlinear mapping of inputs to the higher-dimensional space using entropic and mutual information measures. However, in this work, we use deterministic mapping and would like to ensure that the mapped data makes full use of the larger space and does not create trivial linear interdependencies. To do that, we will concentrate on the effective dimensionality of the mapped inputs. Transformation of the low-dimensional input signal into high-dimensional feature space will be suitable for computing *only* if the output variables are linearly independent and are not redundant. The output signal should be spanned by nonlinear transformation across a large number of available dimensions to make them linearly separable.

While the mapping employed is complex and includes both nonlinearities and time-dependent components, one can employ linear algebra methodology to determine the effective dimensionality of the feature space. In statistical data analysis terms, the effective dimensionality of the mapped inputs is the number of orthogonal dimensions that would produce the same overall co-variation pattern. This can be easily done using singular value analysis, exploratory factor analysis, principal component analysis and other dimensionality reduction techniques, both linear and non-linear [47]. One may also consider independent component analysis methods to explore the statistical independence property of mapped data. More suitable for measuring the complexity of time series are measures introduced in [48] and [49]. As the starting point for this research, we have adopted the singular value analysis, which is simple and effective. An alternative approach that could be explored is to enforce lower effective dimensionality through regularization during linear regression training of the weight matrix, although this on its own cannot be used to evaluate the feature space dimensionality. In classification problems, dedicated algorithms to maximize the distance of data from the separating hyperplane could be employed [50].

2.2 The architecture of the considered photonic ELM and RC systems

The considered photonic ELM/RC computing systems leverage the idea of using nonlinear transformation of signal into high-dimensional space for computational purposes. There are the following key steps in the computing architecture considered as schematically depicted in Figure 1. The first step is the interface between the real-world signal and an input into the computing system. Second, an input signal is tuned (modulated, coded) using available degrees of freedom of the system - parameters that can be used for tuning. There are two possibilities: In the case of slowly varying parameters, the signal undergoes a masking procedure to maximize the effective dimensionality of the system after the nonlinear transformation. When parameters are modulated fast characteristic of the proposed use of telecom devices, adjusting tuning to the incoming signal creates a possibility of dynamical, controllable masking. Third, the nonlinear element/system maps the input signal onto a high-dimensional feature space. Variables that cannot be linearly separated in a low-dimensional space can be successfully processed using linear algorithms in a high-dimensional space. The final step is signal processing in the high-dimensional space performed by the readout layer. Below, we specify each of these steps considering the implementation of a general computing scheme using telecom-grade optical devices.

2.2.1 Encoding techniques: dynamical masking

To feed information into the proposed optical ELM/RC for computing one needs to encode the input data stream in an electromagnetic field. Consider that the input data has a form of a vector (s_1, s_2, \dots, s_N) . Encoding of the input messages (symbols) onto the optical waveform can be done in different ways exploiting the available degrees of freedom of light. The advantage of the proposed ELM/RC is that we can use easily accessible and well-developed telecom devices. For example, amplitude and/or phase modulation can be employed to encode input symbols in the amplitude/phase of the optical signal in the time domain. Encoding can be done in different frequency channels using established WDM technology. In addition to the symbol encoding, the effective dimensionality of the input can be increased via additional signal-invariant modulation, i.e., masking. Consider a WDM-based encoding scheme with L spectral channels and an optical pulse train having the form:

$$A(t) = \sqrt{P_0} \sum_{l=1}^L \sum_{j=1}^J a_{l,j} g(t - j \cdot T_s) \exp(-i\omega_l t) \quad (1)$$

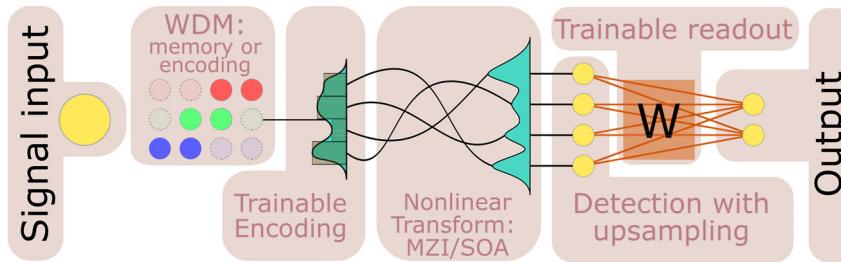


Figure 1: Scheme of the proposed photonic ELMs/RCs.

where P_0 is the power scaling parameter, L is a number of spectral channels (indexed by l) used for coding of the information, T_s is a symbol rate, $a_{l,j}$ for $j = 1, 2, \dots$ is a symbol (in general, a complex number) in the spectral channel l at the temporal position j that is used for encoding the input information in the optical domain. It can be sampled from a discrete set (alphabet) or be continuous.

Function $g(t)$ describes the shape of a carrier pulse (or an encoding mask) having the temporal scale T_0 , which can be, in general, different from T_s . Thus, in the considered case, we need to encode the input data vector (s_1, s_2, \dots, s_N) into array $a_{l,j}$. Evidently, this can be done in different ways, providing rich opportunities for the manipulation and optimization of the subsequent nonlinear mapping. For example, one can encode multiple consecutive symbols s_i into multiple parallel instances of $a_{l,j}$, effectively parallelizing processing. Or one can introduce a time shift, when $a_{l,j}$ contain shifted copies of s_i in different spectral channels, effectively introducing memory. To illustrate the general concept we consider here only simple intensity modulation ($a_{l,j}$ are real non-negative numbers).

In this work, spectral multiplexing is used differently depending on the task. For time-series forecasting, we introduce memory across channels by encoding time-delayed replicas of a base symbol sequence: specifically, we define $a_{l,j} = a_{1,j-l+1}$ for $l = 1, \dots, L$ and $j \geq l$, such that channel l contains the $(l-1)$ -symbol delayed copy of the original sequence on channel 1. For classification tasks, we encode multiple features or input elements (image pixels) in parallel across different WDM channels at the same temporal index, effectively reducing the sequential processing length. These encoding schemes are shown in Table 1.

One can introduce asymmetry in the optical signal we use the skewed Gaussian pulse of the following form:

$$g(t) = g(t, T_0, \alpha) = \frac{\exp(-\tau^2/2T_0^2)}{1 + \exp(-\alpha\tau/T_0)} \quad (2)$$

where α is a skewness parameter, $\tau = (t - t_1)/t_2$ is the shifted and rescaled time introduced to align the mean and variance of the skewed function with those of a standard

Table 1: WDM-based encoding schemes. (a) Time-series forecasting uses delayed copies across spectral channels to introduce memory. (b) Classification encodes features in parallel across channels to reduce sequential depth.

(a) Time-series forecasting						
WDM channel	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$...
ω_1	s_1	s_2	s_3	s_4	s_5	...
ω_2		s_1	s_2	s_3	s_4	...
ω_3			s_1	s_2	s_3	...
ω_4				s_1	s_2	...
ω_5					s_1	...

(b) Classification			...
WDM channel	$j = 1$	$j = 2$...
ω_1	s_1	s_6	...
ω_2	s_2	s_7	...
ω_3	s_3	s_8	...
ω_4	s_4	s_9	...
ω_5	s_5	s_{10}	...

Gaussian (with $\alpha = 0$, $t_1 = 0$, $t_2 = 1$), scaling parameters $t_{1,2}$ adjust the shift and scale of the distribution, respectively. In this work, we always use t parameters optimized to provide a skewed function that preserves the mean and variance of the original Gaussian distribution, thus allowing for a direct comparison while accounting for asymmetry in the mapping. The use of the asymmetric carrier pulse combined with up-sampling effectively plays a role in the masking procedure that assists the following nonlinear distribution of the same symbol into different parts of an output feature space.

Finally, the encoding mask can also be trained to optimize the performance of the ELM. In this case, we can tweak individual parameters of the encoding mask to maximize the accuracy of the proposed ELM. In this work, we used GWO [51] for global optimization of the encoding mask and Nelder-Mead simplex method for refinement. Figure 2 shows an example of an arbitrary encoding mask limited by the analog bandwidth of the arbitrary waveform generator.

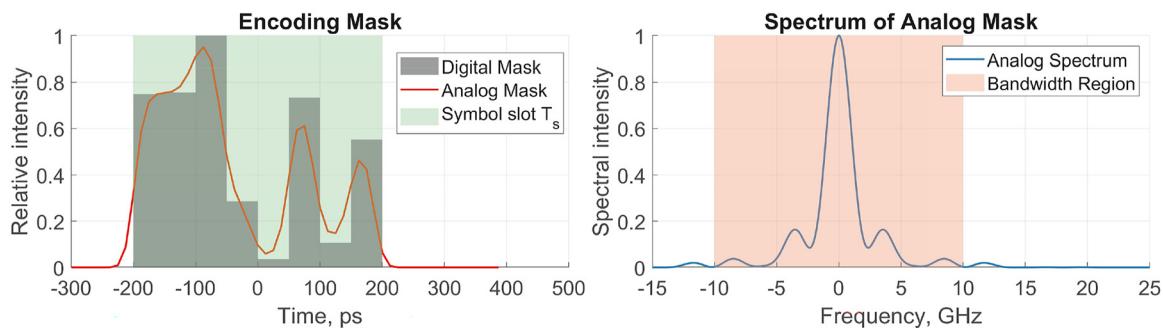


Figure 2: Arbitrary and trainable encoding mask (digital and analog, limited by the bandwidth of arbitrary waveform generator). This example features eight trainable weights.

2.3 Nonlinear transformation in optical domain

NOLM operation can be explained as follows: the input signal power is divided between the two arms of a waveguide loop (for instance, an optical fiber), the signal phase in each arm is changed by the nonlinear propagation, and the resulting signal is formed by coupling the output ports of the arms (see next subsection for details). NOLM can produce overall nonlinear response using unequal coupling ratios, creating asymmetry of the accumulated nonlinear phase shifts, as in the original proposal [52] or by introducing imbalance in nonlinear propagation by using amplifiers (nonlinear amplifying loop mirror – NALM). Evidently, NOLM waveguide device can be realized on different material platforms. The same concept can be implemented in a nonlinear analog of the Mach–Zehnder interferometer, creating an interferometric phase converter to control the sign of the nonlinear phase shift [53].

SOA is a well-developed technology with many attractive characteristics, including compact size, efficient electrical pumping, cost-effectiveness and wideband gain [54]–[56]. However, in high-speed optical communication applications, the nonlinear properties of SOAs, relatively slow gain recovery time, and comparatively high noise figures (compared to other optical amplifiers) pose serious challenges. The carrier dynamics of SOAs have a characteristic scale of several hundred picoseconds. In the context of high-speed optical communications, this produces dependence of an instantaneous SOA gain on the input optical signal power that results in patterning effects – nonlinear distortions with memory. However, these nonlinear and inherent memory features can be attractive for optical computing applications, as demonstrated below. We describe the transfer functions for both NOLM and SOA below. However, for clarity, we primarily present results

based on NOLM-based computing. Examples of optical computing using SOA-based high-dimensional mapping can be found in [57].

The output state is obtained by taking the absolute value squared of the output of the complex transfer functions, i.e. we use intensity-only detection. We assume that the output signal is measured using photodetectors. Additionally, we consider the passive fiber losses are negligible compared to the signal modulation due to nonlinear signal transform.

2.3.1 Nonlinear optical loop mirror

The transfer function of a nonlinear optical loop mirror with a coupler having a split ratio κ is given by the following expression [52]:

$$A_{\text{out}}(t) = \sqrt{\kappa} \cdot \text{NLSE} \left(\sqrt{\kappa} A_{\text{in}}(t), \beta_2, \gamma, L_{\text{NOLM}} \right) + i \sqrt{1 - \kappa} \cdot \text{NLSE} \left(i \sqrt{1 - \kappa} A_{\text{in}}(t), \beta_2, \gamma, L_{\text{NOLM}} \right) \quad (3)$$

Here the input light field $A_{\text{in}}(t)$ is split into two counter-propagating waves in the NOLM with the amplitudes defined by the coupling parameter κ [52]. The function $\text{NLSE}(A, \beta_2, \gamma, L_{\text{NOLM}})$ here represents the solution of the nonlinear Schrödinger equation for the amplitude $A(t, z)$ with a given input $A_{\text{in}}(t)$ after propagating through a fiber of the length L_{NOLM} , with a group velocity dispersion β_2 and nonlinearity parameter γ :

$$\frac{\partial A}{\partial z} = -i \frac{\beta_2}{2} \frac{\partial^2 A}{\partial t^2} + i\gamma |A|^2 A \quad (4)$$

The intensity of the nonlinearly transformed signal can then be measured at the readout stage. When dispersive effects are negligible compared to nonlinear ones (e.g., using

fiber/waveguide near zero-dispersion point or high power signal), then the transfer function is simplified to the compact form:

$$|A_{\text{out}}(t)|^2 = |A_{\text{in}}(t)|^2 \{1 - 2\kappa(1 - \kappa) \times [1 + \cos((1 - 2\kappa)\gamma L_{\text{NOLM}} |A_{\text{in}}(t)|^2)]\}$$

It is seen from this analytical approximation of the NOLM transfer function that by varying parameters κ and L_{NOLM} one can dramatically change the properties of the nonlinear transformation.

In this work, to calculate the input-output signal transformation in nonlinear MZI (NOLM), we numerically simulate signal propagation through fiber by solving Eq. (...) using the fourth-order Runge–Kutta in the interaction picture method [58]. We do it for both arms, i.e. with initial conditions $\sqrt{\kappa}A_{\text{in}}(t)$ and $i\sqrt{1 - \kappa}A_{\text{in}}(t)$. We then combine the solutions at $z = L_{\text{NOLM}}$ as in Eq. (...) and take the absolute value squared to get the signal as registered by a photodetector.

2.3.2 Semiconductor optical amplifier

Nonlinear transformation of the input optical signal $A_{\text{in}}(t) = \sqrt{P_{\text{in}}(t)} \exp(i\phi_{\text{in}}(t))$ by SOA to the output field $A_{\text{out}}(t) = \sqrt{P_{\text{out}}(t)} \exp(i\phi_{\text{out}}(t))$ is governed by the well-established model [54], [55]:

$$\begin{aligned} P_{\text{out}}(t) &= P_{\text{in}}(t) \exp[h(t)] \\ \phi_{\text{out}}(t) &= \phi_{\text{in}}(t) - \frac{\beta}{2}h(t) \\ \frac{dh}{dt} &= -\frac{h - h_0}{\tau_c} - \frac{P_{\text{in}}(t)}{E_{\text{sat}}} [\exp(h) - 1] \end{aligned} \quad (5)$$

Here in/out index denotes the input/output signal, β is the linewidth enhancement (Henry) factor, h_0 parameter is related to the small signal gain $G_0 = \exp(h_0)$, τ_c is the gain recovery time, E_{sat} is a characteristic saturation energy. It is seen that the nonlinear transformation of the input temporal signal stream by SOA creates an effective device memory defined by the delayed gain recovery: gain at a certain point in time $h(t)$ depends on the signal in the previous moments. In our previous work [57] we demonstrate that memory capacity associated with this effect to be ≥ 3.5 . We demonstrate that this property can be exploited to create a high-dimensional feature output. Eqs. (5) are directly integrated using Runge–Kutta method.

By varying SOA current (linked to h_0 parameter), we can dynamically control the nonlinear transformation.

2.4 Readout approaches to implement high-dimensional feature space

One of the key advantages of implementing nonlinear mapping to high-dimensional signals is the possibility to use simple processing, well-established telecom technologies and components at the readout layer. Though signal polarization and spatial modes can also be controlled and manipulated with telecom-grade devices, here we focus on the frequency and temporal domains at the output.

In the temporal domain, applying the up-sampling technique, data encoded in the symbol time interval (at the baud rate) and spread by the nonlinear transformation can be measured at the sampling rate of the receiver. When data encoding is implemented on both the amplitude and the optical phase of the carrier pulses, the standard telecom coherent receiver can be used to recover both amplitude and phase information at the sampling rate. Up-sampling here means the temporal sampling of the nonlinearly transformed signal (intensity in this considered illustration) at a higher rate than the encoding symbol (baud) rate. The up-sampling coefficient M describes how many points we get for a single input pulse. Thus, the output of the considered ELM/RC is a nonlinearly transformed high-dimensional representation of the input signal, which is a key component of the computing. This process for NOLM and $M = 16$ is illustrated in Figure 3.

Applying a modulator at the sampling rate, we can adjust the readout weight in the optical domain or, using high-speed optoelectronics available in telecom, change weights in the electrical domain. Here, we simulate readout by multiplying the row-vector of intensities at the output of the ELM/RC with the regression column-vector, determined during the training procedure (see Section 2.5).

In this work, we utilize WDM solely for encoding the input signal across multiple spectral channels. After the nonlinear transformation, these channels are not separated at the output stage. Instead, we assume a detection scheme where the combined optical output, containing all WDM channels, impinges on a single photodetector. The temporal up-sampling readout approach is then applied to this aggregated signal.

While this joint detection approach is employed here, it is worth noting that an alternative readout strategy could involve separating the different spectral channels using well-developed WDM technology. In such a scenario, the output optical signal would pass through a WDM de-multiplexer, and the temporal up-sampling

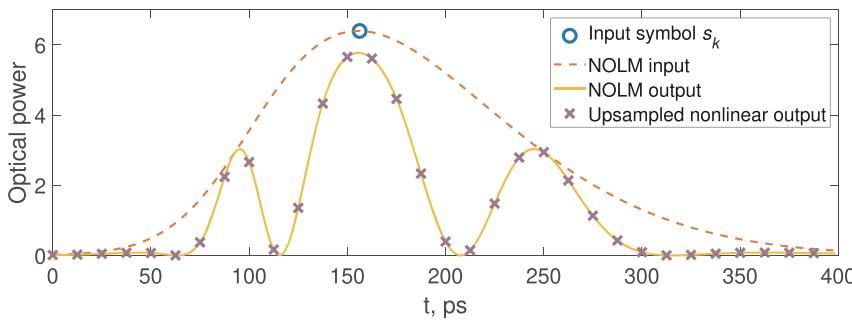


Figure 3: Input symbols, their optical encoding, NOLM output, and the samples captured by a photodetector. For readability, 16 samples per symbol are shown instead of the 32 used in experiments.

readout approach could be applied independently to each spectral channel in parallel. However, exploring such parallel processing via output WDM separation is beyond the scope of the current study, which focuses on the impact of WDM encoding combined with joint detection.

2.5 Training and validation

In this work, we show how the proposed approach can be utilized for the classical machine learning task of time series forecasting. The general scheme of the proposed ELM/RC includes: (i) encoding of the input signal/vector onto the optical field, (ii) a nonlinear element that transforms the input signal (in this work we consider two examples: the nonlinear loop mirror [52] and semiconductor optical amplifier), (iii) trainable readout W that includes detection with up-sampling. The trainable output layer W is straightforward and easily implementable linear regression, enabling the device to be applied in time series forecasting. The scheme of the proposed device is shown in Figure 1.

We would like to stress that the proposed general concept is not limited to the particular choice of the element that implements nonlinear signal transformation. It can be implemented with a variety of nonlinear sub-systems.

To train the system, we must provide examples of correct answers (targets) to given feature vectors. We pass multiple sections of the sequence to forecast through the system, collect the feature (row) vectors and assemble a so-called feature matrix \mathbf{X} . Then, we construct a (column) vector \mathbf{Y} of correct answers or targets by putting the correct next symbol in front of the corresponding row of features.

For example, if one wants to train the system to predict using N symbols, one can use M symbol sequences to construct the following feature matrix:

$$\begin{aligned}
 s_1, s_2, \dots, s_N &\rightarrow \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1K} \end{bmatrix} \\
 s_2, s_3, \dots, s_{N+1} &\rightarrow \begin{bmatrix} x_{21} & x_{22} & \dots & x_{2K} \end{bmatrix} \\
 s_3, s_2, \dots, s_{N+2} &\rightarrow \begin{bmatrix} x_{31} & x_{32} & \dots & x_{3K} \end{bmatrix} \\
 &\vdots \\
 s_M, s_{M+1}, \dots, s_{M+N} &\rightarrow \begin{bmatrix} x_{M1} & x_{M2} & \dots & x_{MK} \end{bmatrix}
 \end{aligned} \tag{6}$$

Here, K denotes the dimensionality of the output vector, N denotes number of symbols fed into RC/ELM to generate features for next symbol prediction, M denotes number of training samples and the symbol \rightarrow denotes the procedure of encoding the symbols in a waveform, propagating this waveform through the physical system and detecting it. The row vectors on the right-hand side of the arrows are the feature vectors that correspond to the input symbol sequences.

When training to predict the next symbol, the corresponding target vector is:

$$\mathbf{Y} \equiv \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} s_{N+1} \\ s_{N+2} \\ \vdots \\ s_{M+N+1} \end{bmatrix} \tag{7}$$

In front of each (row) vector of features, we place the next symbol according to the sequence of symbols used to generate these features. This is what we refer to as the single-step prediction task. We also evaluated our approach for autoregressive multi-step prediction, where the single-step prediction is applied iteratively, updating the time frame to incorporate newly predicted symbols.

The process of training a model involves identifying the regression vector θ that minimizes the mean squared

error (MSE) between the predicted values, represented as $\mathbf{X}\theta$, and the actual target values, \mathbf{Y} . The dimensionality of θ corresponds to the number of output features, K . In the following, we also use the normalized mean squared error (NMSE), defined as the MSE divided by the variance of \mathbf{Y} . NMSE provides interpretability, as $1 - \text{NMSE} = R^2$, which represents the proportion of variance explained by our model. The goal is to find θ such that the difference between these predicted and actual values is as small as possible, which can be expressed as:

$$\min_{\theta} \|\mathbf{Y} - \mathbf{X}\theta\|_2^2 \quad (8)$$

To achieve this, an effective approach is to use the Moore–Penrose pseudoinverse, denoted by \mathbf{X}^\dagger . This pseudoinverse offers a least-squares solution to the equation $\mathbf{Y} = \mathbf{X}\theta$. When the matrix \mathbf{X} is decomposed using singular value decomposition (SVD) as $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^*$, the pseudoinverse can be computed as:

$$\mathbf{X}^\dagger = \mathbf{V}\Sigma^{-1}\mathbf{U}^* \quad (9)$$

In this context, Σ^{-1} is formed by taking the reciprocal of each nonzero singular value in Σ and transposing the resulting matrix. Using this pseudo-inverse, the optimal θ is given by:

$$\theta = \mathbf{X}^\dagger \mathbf{Y} \quad (10)$$

However, calculating the pseudo-inverse involves inverting the singular values, which can amplify even minor variations in the feature matrix, leading to significant fluctuations in the regression vector. This issue is particularly problematic when noise is present in the data, as it can result in unstable and unreliable predictions. Regularization techniques are employed to mitigate this problem.

We use a widely used regularization technique, L_2 regularization, which penalizes large coefficients by adding a term proportional to the square of the coefficient magnitudes. This can be implemented by truncating the singular values in the SVD of \mathbf{X} , which reduces the influence of smaller singular values associated with less important features.

In truncated SVD, a threshold is set, and singular values below this threshold are zeroed out, effectively lowering the rank of Σ . This approach can be represented as solving the following regularized problem:

$$\min_{\theta} \|\mathbf{Y} - \mathbf{X}\theta\|_2^2 + \lambda_{\text{reg}} \|\theta\|_2^2 \quad (11)$$

Here, λ_{reg} is a regularization parameter that balances the trade-off between fitting the data well and keeping θ

small. The adjusted singular values, denoted by Σ_r , are defined as:

$$\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r, 0, \dots, 0) \quad (12)$$

where r is the number of singular values retained. This leads to a regularized estimate of θ :

$$\theta_r = \mathbf{V}_r \Sigma_r^{-1} \mathbf{U}_r^* \mathbf{Y} \quad (13)$$

In this expression, \mathbf{V}_r and \mathbf{U}_r correspond to the matrices associated with the first r singular values.

We carefully adjusted λ_{reg} to minimize MSE on the validation subset.

In what follows, we train a simple linear regression model on this high-dimensional output vector space. To achieve good performance of the proposed computing system, the high-dimensional output vector space must be non-degenerate, providing distinct and linearly independent vectors for diverse inputs. The feature matrix in the regression consists of output vectors, where the number of features, or components of these vectors, determines the dimensionality. However, this perceived dimensionality can be misleading due to the potential degeneracy among the columns, reducing the effective dimensionality of the output. When some columns are linear combinations of others, they do not contribute new information, are useless for regression, and lead to a low-rank feature matrix. This shows the importance of characterizing the true dimensionality of the space represented by the matrix, which can significantly differ from just the number of columns. We use singular value decomposition to analyze the effective dimensionality as it provides insight into the matrix structure by factorizing it into two unitary matrices and a diagonal one containing singular values. These singular values, forming the singular value spectrum, indicate the significance of each dimension in capturing the matrix's variability.

To evaluate and demonstrate the depth of the feature space formed by the optical encoding and nonlinear transformation, we introduce 1000 input symbols x_k from 0 to 1 through the PELM and construct a feature matrix using output row vectors.

We illustrate the dependence of the singular value spectrum of the feature matrix on $\gamma P_0 L_{\text{NOLM}}$ and α in Figure 4, and also present a feature matrix derived from up-sampled signals that were not processed through the NOLM. One can see that the feature space for simply encoded and up-sampled input symbols is highly degenerate, as all feature variability can be explained with a single singular value: the second singular value is more than 10 orders of magnitude lower than the first (see no NOLM line in Figure 4). This

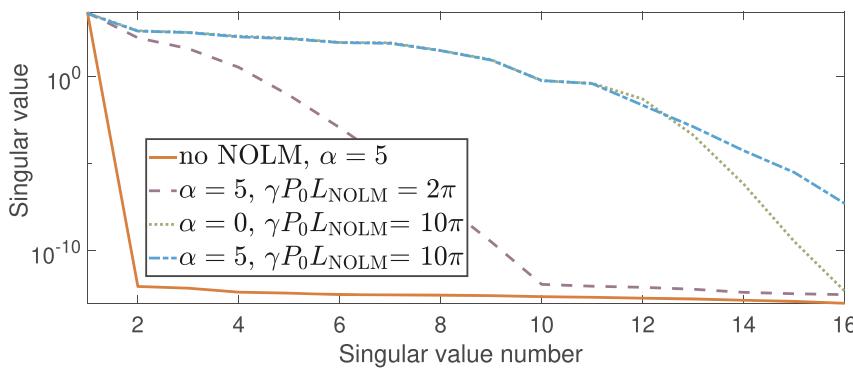


Figure 4: Improvement in the spectrum of singular values of the feature matrix after applying NOLM. The graph shows a richer singular value spectrum at NOLM output, indicating higher effective dimensionality.

corresponds to the effective rank 1 feature space. When NOLM is used for nonlinearly dispersing the signal, the (non)-degeneracy of the feature space depends on the nonlinear phase shift and symmetry of the encoding function g . This comparison reveals how the NOLM significantly influences the inherent dimensionality of the output state. When applying the WDM technique for the readout procedure, one can retrieve a full 2D map of features in time-frequency axes.

2.6 Simulation hyperparameters

To ensure reproducibility and clarity, we summarize below all hyperparameters used across our numerical experiments in Table 2. These include system-specific parameters for the encoding and nonlinear transformation stages, as well as settings related to the learning process and readout. The table covers both nonlinear MZI/NOLM- and SOA-based implementations, along with details of the receiver and training configuration.

3 Benchmarks

For evaluating the performance, we used three different time series with different levels of complexity and the MNIST classification benchmark. When simulating the series, we employ typical parameters commonly used for the evaluation of machine learning models, specifically using the Mackey–Glass series in the form:

$$\frac{dx}{dt} = \frac{0.2x(t-17)}{1+x(t-17)^10} - 0.1x(t) \quad (14)$$

And another time series is given by the solution of the Rossler attractor:

$$\begin{aligned} \frac{dx}{dt} &= -y - z \\ \frac{dy}{dt} &= x + 0.2y \\ \frac{dz}{dt} &= 0.2 + z(x - 5.7) \end{aligned} \quad (15)$$

These dynamical systems are known for being chaotic and are often used for estimating the performance of forecasting algorithms [59]. For 3D attractor, we used only the x component for training and testing the prediction accuracy. The inherent unpredictability of chaotic systems, characterized by their sensitivity to initial conditions, makes them

Table 2: All hyperparameters used in the simulation.

Category	Parameter	Value
Encoding	Symbol interval T_s	400 ps
	Samples per Symbol	32
	Peak power P_0	50 W
	Num WDM channels	5
	WDM frequency step $\Delta\nu$	50 GHz
	Encoding mask size	8
	Analog bandwidth	20 GHz
MZI/NOLM	Nonlinearity γ	$0.8 \text{ W}^{-1}\text{km}^{-1}$
	GVD β_2	$26 \text{ ps}^2\text{km}^{-1}$
	Length L	100 m
	Coupling ratio κ	0.3
SOA	Recovery time τ	200 ps
	Henry factor β	5
	Log gain h_0	6.91
	Saturation energy E_{sat}	8 pJ
Receiver	Sampling rate	80 GSa/s
	Samples per symbol	32
Learning	Training symbols	5,000
	Testing symbols	100–5,000
	Symbols N	15
	Regularization λ_{reg} (optimized per task and device SOA/NOLM)	10^{-8} to 10^{-2}

ideal benchmarks for testing the limits of predictive models. In such environments, even the slightest variation in initial conditions can lead to vastly different outcomes, challenging the algorithms to capture the complex dynamics at play.

To solve Eq. (14), we used the dde23 MATLAB solver with an adaptive step size to ensure target relative and absolute tolerances of $\text{RelTol} = 1e - 6$ and $\text{AbsTol} = 1e - 8$. The solution was then interpolated onto a uniform grid with a step size of 1, following the approach in Jaeger and Haas [60]. For the Rössler attractor, we used the ode45 MATLAB solver with adaptive step to ensure the same relative and absolute tolerances. We then interpolated the solution onto a uniform grid with a step size of 0.25. For the Rössler attractor, the time series was rescaled to lie within the range $[0.1, 1.1]$ to avoid zero or negative values, which are incompatible with optical power levels in our simulation. In contrast, the Mackey–Glass time series was used without normalization, as its amplitude remained within a suitable range for our modeling.

To characterize the randomness of these systems, we estimate the Lyapunov time, which measures the rate at which nearby trajectories in the system's phase space diverge. Specifically, the Lyapunov time is inversely related to the Lyapunov exponent of the system, indicating how quickly initial uncertainties or errors grow over time. To estimate the Lyapunov time for each dynamical system, we numerically integrated them from slightly different initial conditions and calculated the L_2 norm of the difference of these trajectories over time. These trajectories for the considered systems are shown in Figure 5. The Lyapunov time was estimated as the inverse slope of the difference in the logarithmic scale. In what follows, we show how the increased chaoticity of and reduction in Lyapunov time leads to reduced prediction ability of the proposed PELM. Chaoticity increases from the Mackey–Glass series to the

Rössler attractor, with a corresponding decrease in Lyapunov time, indicating faster unpredictability in systems like Rössler due to exponential growth of initial differences, as shown in Figure 5.

To validate our Lyapunov time estimation based on trajectory divergence, we also computed the Lyapunov exponent for the Rössler attractor using the standard variational method based on linearization of the system equations. Specifically, we integrated both the original system and its linearized form (variational equations) and periodically orthonormalized the perturbation vectors using QR decomposition to estimate the largest Lyapunov exponent. This yielded a value for inverse Lyapunov exponent of approximately 13.4, which is consistent with the rate inferred from the divergence of nearby trajectories shown in Figure 5. For the Mackey–Glass system, a similar analysis could not be performed due to its infinite-dimensional nature as a delay differential equation, which complicates the formulation and integration of variational equations. Therefore, for Mackey–Glass, we relied solely on the trajectory-based method to estimate the Lyapunov time.

For the classification task, we took 10,000 samples from the MNIST dataset and split them 50/50 for training/testing. Since linear classifiers already achieve around 90 % accuracy on the original 784-pixel images, we increased the task difficulty by downsampling the images to just 15 pixels. We accomplished this using column-pivoting QR decomposition to identify and retain the most informative pixels. To do this, we reshaped the MNIST dataset into a matrix in the form 10,000 by 784 and applied column-pivoting QR decomposition, following the data-driven QR sensing paradigm, which is basically a compressed sensing approach performed on a tailored basis [61]. The pivoting QR factorization seeks to reorder the columns such that the most “informative” columns are moved to the forefront. The

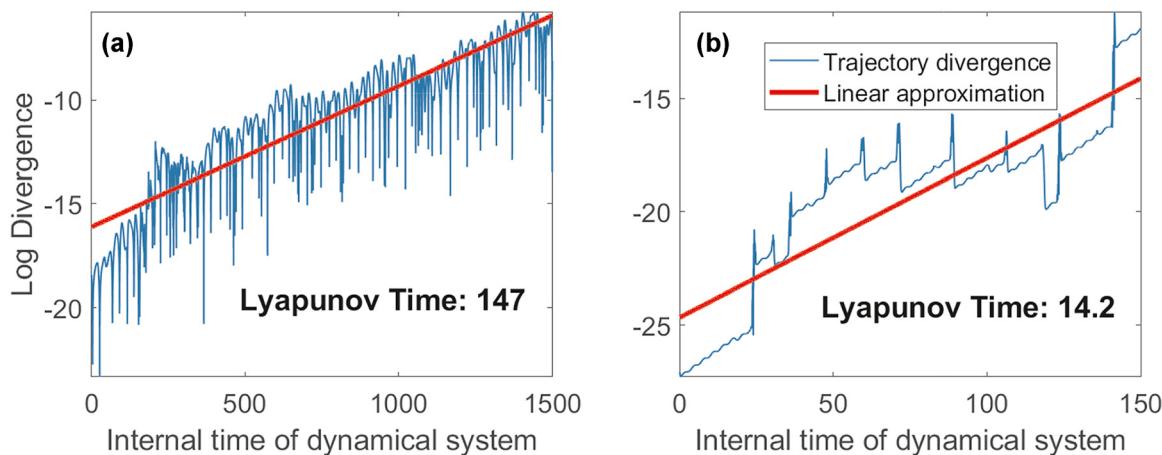


Figure 5: Comparison of Lyapunov times for different dynamical systems, illustrating how increased chaoticity, from the (a) Mackey–Glass series to (b) Rossler attractor, results in reduced Lyapunov time and increased unpredictability.

measure of “informativeness” here is rooted in the magnitudes of the diagonal entries of the upper triangular matrix \mathbf{R} resulting from the QR factorization. Columns, or, in our case, pixels, corresponding to larger diagonal entries in \mathbf{R} are deemed more significant. Thus, we select the most informative columns. To ensure a proper evaluation, this decomposition was performed exclusively on the training subset to prevent any information from the testing subset from influencing the pre-processing stage. The identified pixel positions were subsequently used to downsample images in the testing subset. We used linear discriminant analysis (LDA) for multi-class classification using either original selected pixels (linear classification) or processed by the photonic ELM. Applying this approach, we observed an accuracy of 42 % on the testing subset using linear classification. The 15 selected pixels were then input into the ELM to achieve a high-dimensional nonlinear mapping. A linear classifier was trained on the mapped samples from the training subset within this higher-dimensional feature space. The performance of the classifier was then evaluated using the mapped samples from the testing subset. When using WDM for encoding, we simultaneously transmit 5 different pixels across 5 separate spectral channels, effectively sending the input in batches of 5 pixels at a time, with 3 such batches per image.

4 Results and discussion

To test the ability of the approach to capture the dynamics of the considered systems, we used regularized linear regression over the feature space of the output of the PELM. Regularization parameter was optimized for long-term autoregressive prediction by testing the long-term prediction accuracy of the model for regularization parameters from 10^{-10} to 10^{-2} , and choosing the value that provides the best accuracy on the autoregressive prediction. So the value of the regularization parameter was selected to avoid overfitting the local prediction while making the model capable of predicting the long-scale dynamics of the three test dynamical systems. Regularized pseudo-inversion of the feature matrix was used to learn the regression vector from the training data: regularization is performed by replacing all singular values that are below the current regularization parameter with zeros. Singular values and pseudoinversion are calculated by using singular value decomposition of the feature matrix. Then, we segment the validation data into sequences that the model will use to make predictions. Each sequence includes a set of input symbols followed by the symbols to be predicted. The number of sequences is determined based on the size of the validation data and the length of the sequences to be predicted. Typically, we

used 10–15 validation sequences. For each input sequence, we predict the sequence of symbols that follow by taking an input sequence and, in a loop, making a prediction for the next symbol in the sequence. After each prediction, the input sequence is updated by removing its first symbol and appending the predicted one. This updated sequence is then used for the next iteration. This process is repeated for the length of the pre-defined autoregressive prediction length. We used different autoregressive prediction lengths for different dynamical systems: from 1,000 for the Mackey–Glass system to 250 for the Rössler attractor.

The simulation parameters that we used for training and testing the proposed system, are given below. We set the receiver sampling rate of 80 GSa/s, with 32 samples recorded per symbol to provide the up-sampling of the encoded signal.

We used symbol-to-pulse train encoding with Gaussian, skewed Gaussian, or a trainable mask. The input pulse had a peak power of 50 W. Once more, we would like to stress that this power corresponds to NOLM based on SMF and can be substantially scaled down by using highly nonlinear fiber, or integrated MZI/NOLM with other material platform. When employing the trainable mask, it was first set digitally with ideal (infinitely steep) transitions, then simulated for encoding by an electro-optical modulator with a 20 GHz analog bandwidth. For WDM multi-channel spectral encoding, we used a frequency step of $\Delta\nu = 50$ GHz between channels, with 5 channels corresponding to frequency shifts of 0, $-\Delta\nu$, $+\Delta\nu$, $-2\Delta\nu$, and $+2\Delta\nu$. For time series forecasting, we encoded time-shifted copies of the same symbol sequences in other spectral channels so that channel 2 contains a 1-symbol delayed copy of the channel 1 sequence, channel 3 contains a 2-symbol delayed copy of the channel 1 sequence, etc. A similar approach was used for memory enhancement in RC in microresonators [37]. However, unlike the approach in [37], our scheme additionally employs trainable encoding. Theoretical studies [62] have also explored memory enhancement via input-stage modifications, particularly in SOAs and microring resonators. Our work extends these insights by demonstrating an implementation in a WDM-based framework, offering flexible spectral encoding, trainable masking for dynamic input modulation, and a scalable approach using commercially available telecom components. We did not demultiplex the WDM channels at the output and instead simulated the readout as being measured directly from the nonlinear transform’s output using a single photodiode. Using WDM for the classification task, we simply encoded different pixels into different WDM channels in parallel, thus reducing the number of sequential symbols to be fed into the ELM.

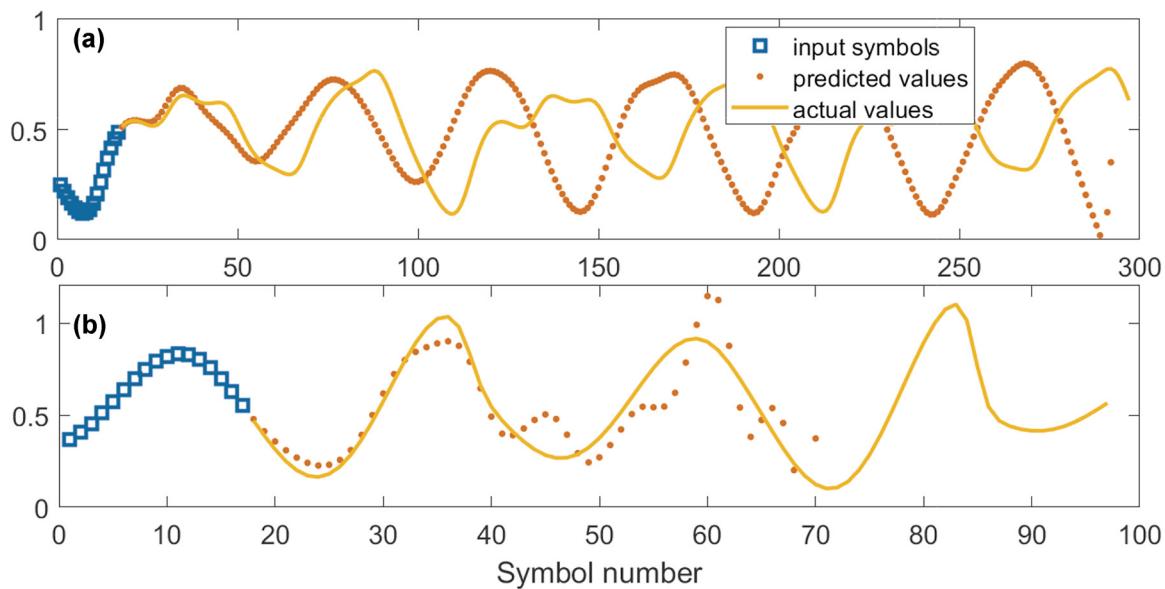


Figure 6: Performance, for comparison, of the linear model on time-series prediction tasks. Poor prediction accuracy is achieved for all the Mackey–Glass (a) and Rössler (b) time series, which indicates the inability of the linear model to capture the complex dynamics of these systems.

The NOLM was characterized using parameters of a single mode fiber: a nonlinear coefficient $\gamma = 0.8 \times 10^{-3} \text{ W}^{-1} \text{ m}^{-1}$, group velocity dispersion coefficient $\beta_2 = 26 \times 10^{-3} \text{ ps}^2 \text{ m}^{-1}$, fiber length $L_{\text{NOLM}} = 100 \text{ m}$, and the coupling ratio $\kappa = 0.3$. Evidently, employing highly nonlinear fibers or other material platforms, all resulting optical pulse parameters can be easily scaled.

For the time series forecasting task, the model was trained using 4000 symbols and tested on 100–1,000 symbols depending on the chaoticity of the time series. We used batches of $N = 15$ symbols to predict the next one. To make a multi-step prediction, we utilized the autoregressive approach, when the newly predicted symbols are used to predict the next ones. A regularization amplitude was optimized for performance and to prevent overfitting. For the classification task, the model was trained using 5000 images from the training set, which were sub-sampled using column-pivoting QR decomposition. The pixel locations chosen for the training subset were then applied to the testing subset during the testing phase, which also included 5000 sample images.

We began by applying simple linear regression and linear classification to the upsampled raw data to establish a reference performance baseline. This initial approach served as a benchmark for evaluating the improvements offered by the nonlinear systems used. As expected, the linear regression model exhibited poor performance across both chaotic time-series prediction tasks and the sub-sampled MNIST classification task. Specifically, the tasks

included autoregressive predictions on the Mackey–Glass (MG) and Rössler (R) time series, with results illustrated in Figure 6.

Although the single-step prediction NMSE appeared relatively low, at $\text{NMSE}_{\text{MG}} \approx 5 \times 10^{-6}$, $\text{NMSE}_{\text{R}} \approx 2 \times 10^{-4}$, the model failed at long-term prediction. This failure was due to the linear system's inability to capture and replicate the underlying dynamics of these complex systems.

For the classification task on the sub-sampled MNIST dataset, the linear model's performance was similarly limited. The model struggled to separate the digit classes, which

	0	350	30	24		31	18	7	4	1	29
1	37	274	35	4	26	37	43	41	6	11	
2	88	47	211	5	25	14	20	69	11	18	
3	17	24	46	98	68	54	21	74	75	40	
4	17	7	36	10	265	30	35	22	30	41	
5	9	30	46	17	36	220	41	14	28	53	
6	30	62	33	27	48	76	138	17	41	29	
7	31	49	89	5	39	23	13	231	2	4	
8	52	20	22	44	38	26	71	20	154	52	
9	18	43	12	15	49	37	43	44	15	218	
	0	1	2	3	4	5	6	7	8	9	
True Class											Predicted Class

Figure 7: Confusion matrix for the linear model on the sub-sampled MNIST classification task. The high number of off-diagonal entries indicates frequent misclassification.

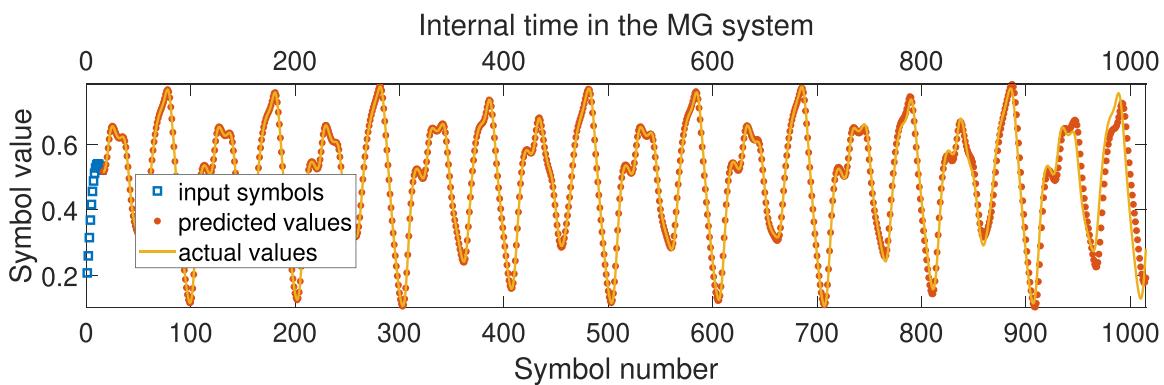


Figure 8: Example of autoregressive prediction of Mackey–Glass series for 1,000 symbols using PELM.

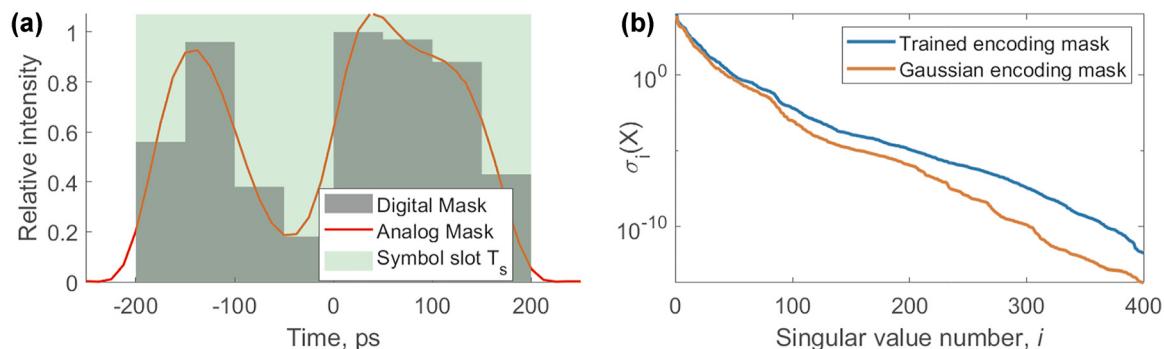


Figure 9: Trained encoding mask (a) and corresponding singular spectrum change (b) in the feature matrix of the PELM model. The model performance was enhanced by using an 8-slot trainable encoding mask, allowing each symbol to be encoded with 8 independently optimized amplitudes.

led to high misclassification rates. This shortfall is evident when examining the confusion matrix in Figure 7, where off-diagonal elements are prominent, showing a high misclassification rate.

The overall accuracy on the test subset for the subsampled MNIST is only 42 %.

Now we determined the baseline accuracy of the linear models on the chosen tasks, and we can compare the accuracy achieved using PELM with the parameters described above. Figure 8 shows an example of the increased predictive ability of the model when high-dimensional nonlinear mapping is enabled via passing the encoded signal through the PELM.

NMSE averaged over multiple predicted sequences, similar to shown in Figure 8 but taken from different parts of the Mackey–Glass sequence, is equal to $\text{NMSE} = 0.097$ for the autoregressive prediction depth of 1,000 symbols, which is a significant improvement compared to the purely linear model, shown in Figure 6a.

The performance of the PELM can be improved further by employing a trainable encoding mask instead of using a simple Gaussian shape for the optical encoding of the

input symbols. We employed an 8-slot trainable encoding mask, such that each symbol is encoded with 8 independent amplitudes. The waveform of each corresponding pulse is then described as a convolution of this arbitrary-shaped vector of length 8 with a finite-bandwidth response function of the modulator. We used the raised cosine function as the response function, and the analog bandwidth of the

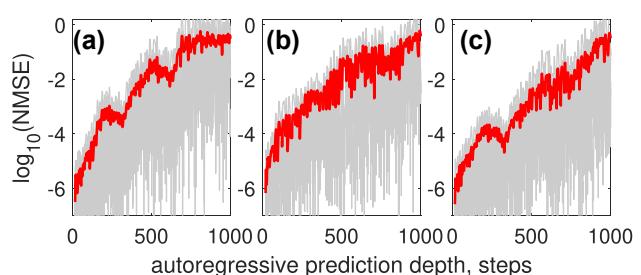


Figure 10: NMSE as a function of autoregressive prediction depth for different encoding techniques: (a) single-channel Gaussian pulse encoding, (b) 5-channel WDM with Gaussian pulse encoding, and (c) single-channel trained encoding mask. Errors for individual predicted sequences are shown in gray, and the average error across all predicted sequences is highlighted in red.

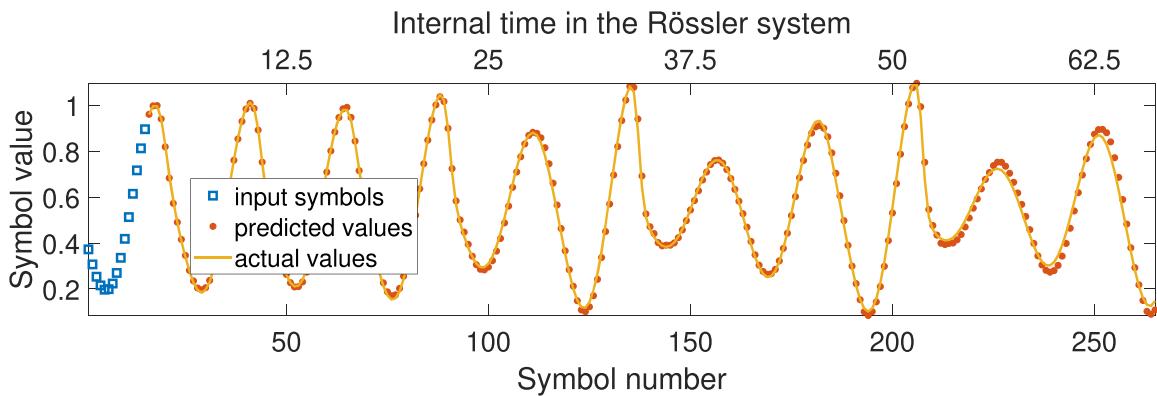


Figure 11: Example of autoregressive forecasting of the Rössler time series using a single-channel Gaussian-shaped encoding.

modulator was set to 20 GHz. The encoding mask was trained by a combination of global and local optimization techniques. We used GWO [51] for global optimization and Nelder–Mead simplex method for local optimization. The trained mask and the corresponding change in the singular spectrum of the feature matrix are shown in Figure 9. The corresponding improvement in performance for autoregressive prediction depth of 1,000 symbols is reduced to $\text{NMSE} = 0.036$, a 2.7 times improvement compared to simple Gaussian encoding mask.

When using another telecom technology, WDM, for encoding (i.e., combining multiple signals within a single optical channel before passing them through the PELM), performance can also be improved compared to the simpler single-channel encoding approach. When using 5 channels with 50 GHz separation and employing single-symbol shift, as described above, we managed to achieve $\text{NMSE} = 0.054$ for simple Gaussian encoding and $\text{NMSE} = 0.024$ for trained encoding mask. The encoding mask used was the same for all spectral channels. The performance can be further optimized by using individual trainable encoding masks for each WDM channel, but this requires further research and training via higher-order parameter optimization, which is beyond the scope of this work.

Figure 10 shows how NMSE depends on autoregressive prediction depths for different encoding techniques: single channel with Gaussian pulse encoding (a), 5-channel WDM with Gaussian pulse encoding (b), and single channel with trained encoding mask (c). The errors for individual predicted sequences are depicted in gray, while the average error across all predicted sequences is highlighted in red.

Compared to other photonic RC approaches for Mackey–Glass prediction, our method outperforms [63], achieving an NMSE of 0.01 for predicting 300 symbols (six quasi-periods), whereas [63] reports an NMSE of 0.1. In

comparison to the delay-based RC in [26], our approach achieves similar accuracy for single-step prediction; however, a direct comparison of multi-step prediction performance is not possible due to the lack of reported error metrics in [26].

When testing on the Rössler attractor data, we similarly can get boost in prediction accuracy when utilizing multiple WDM channels and trainable encoding mask. Figure 11 shows an example of autoregressive forecasting of the Rossler time series using single channel Gaussian-shape encoding.

Using 3 WDM channels for encoding or a trainable encoding mask reduces NMSE across autoregressive prediction depths, as shown in Figure 12.

The overall accuracy for Rössler time series prediction using NOLM again can be improved from $\text{NMSE}_R = 1.16 \times 10^{-1}$ to $\text{NMSE}_R = 5.7 \times 10^{-2}$ when using WMD encoding and to $\text{NMSE}_R = 9.79 \times 10^{-3}$ when using trained encoding mask even with a single encoding channel.

We also tested the proposed approach on SOA as a non-linear mapping device. All system parameters are kept the same except the peak power $P_0 = 0.7$ W. SOA parameters

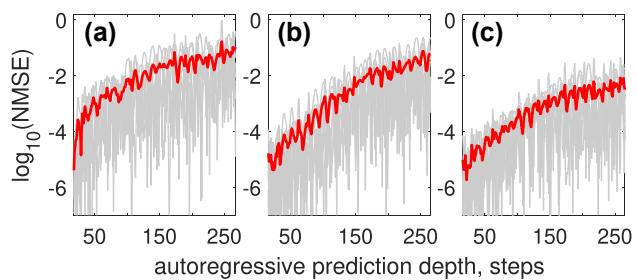


Figure 12: NMSE versus autoregressive prediction depth for the Rössler time series (a); the results demonstrate reduced prediction error when using 3 WDM channels (b) or a trainable encoding mask (c), compared to single-channel encoding.

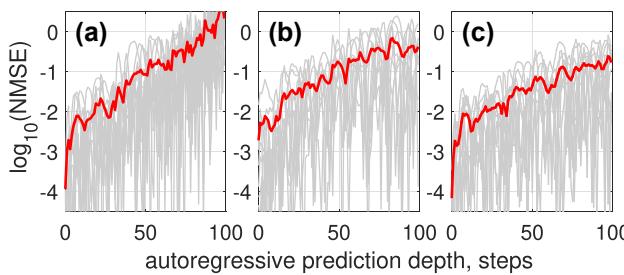


Figure 13: NMSE versus autoregressive prediction depth when using SOA as a nonlinear mapping device, for the Rössler time series (a); the results demonstrate reduced prediction error when using 3 WDM channels (b) or a trainable encoding mask (c), compared to single-channel encoding.

used for simulation (see eq. (5)) are the following: $\beta = 5$, small signal gain 30 dB, $\tau_c = 200$ ps, $E_{sat} = 8$ pJ. All time series and computing parameters are identical to those in Figure 12, except that the nonlinear transform is now performed using an SOA. Figure 13 shows how applying WDM encoding (b) and trainable encoding mask (c) outperform the standard Gaussian encoding mask (a) for Rössler attractor prediction task.

Finally, In the classification task, the proposed approach (see Figure 14 for the improved confusion matrix) demonstrates a substantial performance improvement over the baseline linear model on the sub-sampled MNIST classification task. With WDM encoding, we achieve comparable performance, with the added advantage of encoding multiple symbols simultaneously across different spectral channels.

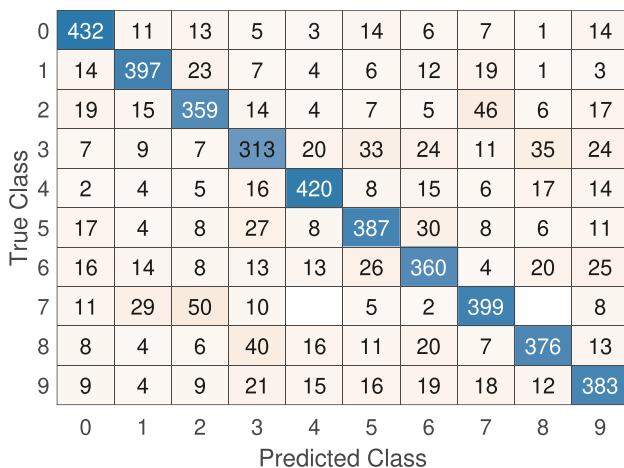


Figure 14: Confusion matrix for the NOLM-based Extreme Learning Machine (ELM) model on the sub-sampled MNIST classification task. The model shows significantly improved performance compared to the linear model, with a reduced number of off-diagonal misclassifications.

Testing on the sub-sampled MNIST classification task shows an overall test accuracy of 77 % for the NOLM-based ELM model, a significant increase from the linear model's 42 % accuracy.

5 Conclusions

We proposed and demonstrated through numerical modeling photonic ELM and RC designs based on the well-developed telecommunication technology and component platform. The important advantage of this approach is the combination of low-cost, high-speed characteristics of linear and nonlinear elements with the frequency parallelism techniques well-established in wave-division-multiplexing optical communication systems. High-capacity and high-speed optical communication are utilized to create a large number of degrees of freedom in the time-frequency domain for controllable encoding and relatively low-power parallel nonlinear mapping of the input signal into high-dimensionality output for linear processing. The use of multiple frequency channels for encoding input information is important for the efficiency of nonlinear mapping. Indeed, in linear time-invariant optical systems, the frequency components in the output and input signals are the same. The nonlinear optical transformation leads to frequency component mixing and generation of new harmonics providing conditions for efficient mapping to higher-dimensional space.

It is important to point out that while we focused here on the forecasting of time series and a classical classification task, a similar projection to the high-dimensional case is also highly relevant and can be employed for various other tasks following the methodology of SVM [7]. The difference, however, is that the proposed approach does not require the use of kernel-based methods. Key to improving the success of the non-linear projection used is adjusting the mapping parameters in a way that will provide an easier separation of features. This is an aspect of the research that is still ongoing.

Research funding: Horizon Europe ITN Postdigital Plus (No. 101169118), Engineering and Physical Sciences Research Council (project EP/W002868/1) and UK Multidisciplinary Centre for Neuromorphic Computing (UKRI982).

Author contributions: SKT conceived the research and proposed using WDM for encoding and up-sampling for high-dimensional mapping. EM proposed using NOLM for nonlinear mapping. EM developed numerical models. EM performed simulations. EM processed the results. All authors have accepted responsibility for the entire content of this manuscript and consented to its submission to the journal,

reviewed all the results and approved the final version of the manuscript.

Conflict of interest: Authors state no conflicts of interest.

Research ethics: The conducted research is not related to either human or animal use.

Data availability: The datasets generated and/or analysed during the current study are available from the corresponding author upon reasonable request.

References

- [1] R. W. Keyes, “Fundamental limits in digital information processing,” *Proc. IEEE*, vol. 69, no. 2, pp. 267–278, 1981.
- [2] H. Jaeger, “Towards a generalized theory comprising digital, neuromorphic and unconventional computing,” *Neuromorph. Comput. Eng.*, vol. 1, no. 1, p. 012002, 2021.
- [3] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for modern deep learning research,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 9, pp. 13693–13696, 2020.
- [4] H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks—with an erratum note,” *Bonn, Germany: German Natl. Res. Cent. Infor. Technol. GMD Tech. Rep.*, vol. 148, no. 34, p. 13, 2001.
- [5] W. Maass, T. Natschläger, 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.
- [6] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.
- [7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, 1 ed. Cambridge, Cambridge University Press, 2000.
- [8] T. W. Hughes, I. A. D. Williamson, M. Minkov, and S. Fan, “Wave physics as an analog recurrent neural network,” *Sci. Adv.*, vol. 5, no. 12, p. eaay6946, 2019.
- [9] H. Jaeger, B. Noheda, and W. G. van der Wiel, “Toward a formal theory for computing machines made out of whatever physics offers,” *Nat. Commun.*, vol. 14, no. 1, p. 4911, 2023.
- [10] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” *Phys. Rev. Lett.*, vol. 120, no. 2, p. 024102, 2018.
- [11] B. Penkovsky, X. Porte, M. Jacquiot, L. Larger, and D. Brunner, “Coupled nonlinear delay systems as deep convolutional neural networks,” *Phys. Rev. Lett.*, vol. 123, no. 5, p. 054101, 2019.
- [12] N. Stroev and N. G. Berloff, “Analog photonics computing for information processing, inference, and optimization,” *Adv. Quantum Technol.*, vol. 6, no. 9, p. 2300055, 2023.
- [13] L. G. Wright, et al., “Deep physical neural networks trained with backpropagation,” *Nature*, vol. 601, no. 9, pp. 549–555, 2022.
- [14] D. Pierangeli, G. Marcucci, and C. Conti, “Photonic extreme learning machine by free-space optical propagation,” *Photon. Res.*, vol. 9, no. 8, pp. 1446–1454, 2021.
- [15] G. Marcucci, D. Pierangeli, and C. Conti, “Theory of neuromorphic computing by waves: Machine learning by rogue waves, dispersive shocks, and solitons,” *Phys. Rev. Lett.*, vol. 125, no. 9, p. 093901, 2020.
- [16] U. Teğin, M. Yıldırım, İ. Oğuz, C. Moser, and D. Psaltis, “Scalable optical learning operator,” *Nat. Comput. Sci.*, vol. 1, no. 8, pp. 542–549, 2021.
- [17] M. Sorokina, S. Sergeyev, and S. Turitsyn, “Fiber echo state network analogue for high-bandwidth dual-quadrature signal processing,” *Opt. Express*, vol. 27, pp. 2387–2395, 2019.
- [18] I. Oğuz, et al., “Programming nonlinear propagation for efficient optical learning machines,” *Adv. Photon.*, vol. 6, no. 1, p. 016002, 2024.
- [19] M. Sorokina, “Multi-channel optical neuromorphic processor for frequency-multiplexed signals,” *J. Phy.: Photon.*, vol. 3, p. 014002, 2020.
- [20] B. Wang, T. F. de Lima, B. J. Shastri, P. R. Prucnal, and C. Huang, “Multi-wavelength photonic neuromorphic computing for intra and inter-channel distortion compensations in wdm optical communication systems,” *IEEE J. Sel. Top. Quantum Electron.*, vol. 29, no. 2, pp. 1–12, 2023.
- [21] A. Lupo, L. Butschek, and S. Massar, “Photonic extreme learning machine based on frequency multiplexing,” *Opt. Express*, vol. 29, no. 18, pp. 28257–28276, 2021.
- [22] A. Röhm, L. Jaurigue, and K. Lüdge, “Reservoir computing using laser networks,” *IEEE J. Sel. Top. Quantum Electron.*, vol. 26, no. 1, pp. 1–8, 2020.
- [23] S. Biasi, R. Franchi, L. Cerini, and L. Pavesi, “An array of microresonators as a photonic extreme learning machine,” *APL Photon.*, vol. 8, no. 9, p. 096105, 2023.
- [24] K. Vandoorne, et al., “Experimental demonstration of reservoir computing on a silicon photonics chip,” *Nat. Commun.*, vol. 5, no. 1, p. 3541, 2014.
- [25] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman, “Parallel reservoir computing using optical amplifiers,” *IEEE Trans. Neural Network.*, vol. 22, no. 9, pp. 1469–1481, 2011.
- [26] S. Ortín, et al., “A unified framework for reservoir computing and extreme learning machines based on a single time-delayed neuron,” *Sci. Rep.*, vol. 5, no. 1, p. 14945, 2015.
- [27] 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.
- [28] C. Mesaritakis, A. Bogris, A. Kapsalis, and D. Syridis, “High-speed all-optical pattern recognition of dispersive Fourier images through a photonic reservoir computing subsystem,” *Opt. Lett.*, vol. 40, no. 14, pp. 3416–3419, 2015.
- [29] L. Butschek, A. Akroud, E. Dimitriadou, A. Lupo, M. Haelterman, and S. Massar, “Photonic reservoir computer based on frequency multiplexing,” *Opt. Lett.*, vol. 47, no. 4, pp. 782–785, 2022.
- [30] A. Lupo, E. Picco, M. Zajnulina, and S. Massar, “Deep photonic reservoir computer based on frequency multiplexing with fully analog connection between layers,” *Optica*, vol. 10, no. 11, pp. 1478–1485, 2023.
- [31] X. Xu, et al., “11 tops photonic convolutional accelerator for optical neural networks,” *Nature*, vol. 589, no. 7840, pp. 44–51, 2021.
- [32] X. Xu, et al., “Neuromorphic computing based on wavelength-division multiplexing,” *IEEE J. Sel. Top. Quantum Electron.*, vol. 29, no. 2, pp. 1–12, 2023.

[33] B. Dong, *et al.*, “Higher-dimensional processing using a photonic tensor core with continuous-time data,” *Nat. Photon.*, vol. 17, no. 12, pp. 1080–1088, 2023.

[34] R.-Q. Li, Y.-W. Shen, B.-D. Lin, J. Yu, X. He, and C. Wang, “Scalable wavelength-multiplexing photonic reservoir computing,” *APL Machine Learning*, vol. 1, no. 3, 2023, <https://doi.org/10.1063/5.0158939>.

[35] B. J. Giron Castro, C. Peucheret, D. Zibar, and F. Da Ros, “Multi-task photonic reservoir computing: wavelength division multiplexing for parallel computing with a silicon microring resonator,” *Adv. Opt. Technol.*, vol. 13, p. 1471239, 2024.

[36] E. Gooskens, F. Laporte, C. Ma, S. Sackesyn, J. Dambre, and P. Bienstman, “Wavelength dimension in waveguide-based photonic reservoir computing,” *Opt. Express*, vol. 30, no. 9, pp. 15634–15647, 2022.

[37] B. J. G. Castro, C. Peucheret, and F. Da Ros, “Wavelength-multiplexed delayed inputs for memory enhancement of microring-based reservoir computing,” in *CLEO: Science and Innovations*, Washington, D.C., Optica Publishing Group, 2024, p. JW2A-28.

[38] K. Takano, *et al.*, “Compact reservoir computing with a photonic integrated circuit,” *Opt. Express*, vol. 26, no. 22, pp. 29424–29439, 2018.

[39] M. Nakajima, K. Tanaka, and T. Hashimoto, “Scalable reservoir computing on coherent linear photonic processor,” *Commun. Phys.*, vol. 4, no. 1, p. 20, 2021.

[40] V. Kilic and M. A. Foster, “Training photonic extreme learning machines using feedback alignment,” in *2021 Conference on Lasers and Electro-Optics (CLEO)*, Washington, D.C., IEEE, 2021, pp. 1–2.

[41] Q. Cai, *et al.*, “Modulation format identification in fiber communications using single dynamical node-based photonic reservoir computing,” *Photon. Res.*, vol. 9, no. 1, pp. B1–B8, 2021.

[42] Y. K. Chembo, “Machine learning based on reservoir computing with time-delayed optoelectronic and photonic systems,” *Chaos: An Interdiscipl. J. Nonlinear Sci.*, vol. 30, no. 1, 2020, <https://doi.org/10.1063/1.5120788>.

[43] A. Argyris, “Photonic neuromorphic technologies in optical communications,” *Nanophotonics*, vol. 11, no. 5, pp. 897–916, 2022.

[44] J. Wu, *et al.*, “Analog optical computing for artificial intelligence,” *Engineering*, vol. 10, pp. 133–145, 2022.

[45] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Berlin, Heidelberg, Springer-Verlag, 2006.

[46] T. M. Cover, “Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition,” *IEEE Trans. Electron. Comput.*, vols. EC-14, no. 3, pp. 326–334, 1965.

[47] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge, Cambridge University Press, 2012.

[48] H. Y. D. Sigaki, M. Perc, and H. V. Ribeiro, “History of art paintings through the lens of entropy and complexity,” *Proc. Natl. Acad. Sci.*, vol. 115, no. 37, pp. E8585–E8594, 2018.

[49] C. Bandt and B. Pompe, “Permutation entropy: A natural complexity measure for time series,” *Phys. Rev. Lett.*, vol. 88, no. 17, p. 174102, 2002.

[50] J. K. Anlauf and M. Biehl, “The Adatron: An adaptive perceptron algorithm,” *Europhys. Lett.*, vol. 10, p. 687, 1989.

[51] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Adv. Eng. Software*, vol. 69, pp. 46–61, 2014.

[52] N. Doran and D. Wood, “Nonlinear-optical loop mirror,” *Opt. Lett.*, vol. 13, no. 1, pp. 56–58, 1988.

[53] I. R. Gabitov and P. M. Lushnikov, “Nonlinearity management in a dispersion-managed system,” *Opt. Lett.*, vol. 27, pp. 113–115, 2002.

[54] A. Sobhanian, *et al.*, “Semiconductor optical amplifiers: recent advances and applications,” *Adv. Opt. Photon.*, vol. 14, pp. 571–651, 2022.

[55] G. Agrawal and N. Olsson, “Self-phase modulation and spectral broadening of optical pulses in semiconductor laser amplifiers,” *IEEE J. Quantum Electron.*, vol. 25, no. 11, pp. 2297–2306, 1989.

[56] A. Mecozzi and J. Mørk, “Saturation induced by picosecond pulses in semiconductor optical amplifiers,” *J. Opt. Soc. Am. B*, vol. 14, no. 4, pp. 761–770, 1997.

[57] E. Manuylovich, A. E. Bednyakova, D. A. Ivoilov, I. S. Terekhov, and S. K. Turitsyn, “Soa-based reservoir computing using upsampling,” *Opt. Lett.*, vol. 49, no. 20, pp. 5827–5830, 2024.

[58] J. Hult, “A fourth-order runge–kutta in the interaction picture method for simulating supercontinuum generation in optical fibers,” *J. Lightwave Technol.*, vol. 25, no. 12, pp. 3770–3775, 2007.

[59] T. Chomiak and B. Hu, “Time-series forecasting through recurrent topology,” *Commun. Eng.*, vol. 3, no. 1, p. 9, 2024.

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

[61] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, “Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns,” *IEEE Control Syst. Mag.*, vol. 38, no. 3, pp. 63–86, 2018.

[62] T. Hülser, F. Köster, L. Jaurigue, and K. Lüdge, “Role of delay-times in delay-based photonic reservoir computing,” *Opt. Mater. Express*, vol. 12, no. 3, pp. 1214–1231, 2022.

[63] J. Dong, M. Rafayelyan, F. Krzakala, and S. Gigan, “Optical reservoir computing using multiple light scattering for chaotic systems prediction,” *IEEE J. Sel. Top. Quantum Electron.*, vol. 26, no. 1, pp. 1–12, 2019.