Supplementary Materials for

Optoelectronic reservoir computing based on complex-value encoding

Chunxu Ding,[a,†] Rongjun Shao,[a,†] Jingwei Li,[b] Yuan Qu,[a,c] Linxian Liu,[d] Qiaozhi He,[c] Xunbin Wei,[e,*] Jiamiao Yang[a,c,*]

[a]Shanghai Jiao Tong University, School of Electronic Information and Electrical Engineering, No.800, Dongchuan Road, Minhang District, Shanghai, China, 200240
[b]Huawei Technologies Co., Ltd, Bantian Street, Longgang District, Shenzhen, China, 518129
[c]Shanghai Jiao Tong University, Institute of Marine Equipment, No.211, Wenjing Road, Minhang District, Shanghai, China, 200240
[d]Shanxi University, School of Automation and Software Engineering, No.63, South Central East Street, Xiaodian District, Taiyuan, China, 030006
[e]Peking University, Biomedical Engineering Department and International Cancer Institute, No.38, Xueyuan Road, Haidian District, Beijing, China, 100081
[†]These authors contributed equally to this work.

*Corresponding author. Email: xwei@bjmu.edu.cn, jiamiaoyang@sjtu.edu.cn

This file includes:

Supplementary Text
Figures. S1 to S3

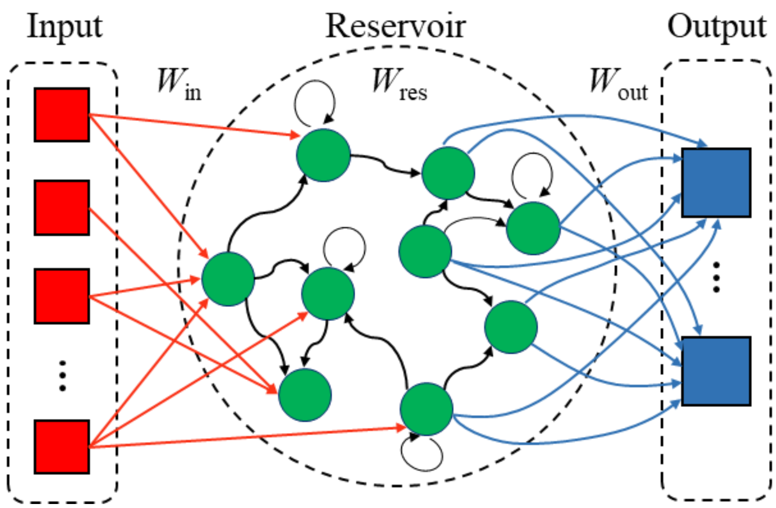

Figure. S1. Typical RC architecture. The basic structure for RC is formed by an input layer, a reservoir, and an output layer. The input information is sent to the reservoir, whose internal connections act like a fixed hidden layer in artificial neural networks. The response of the reservoir is used to produce the desired output after an optimization procedure during the training process (only the output connections are adapted in the training).

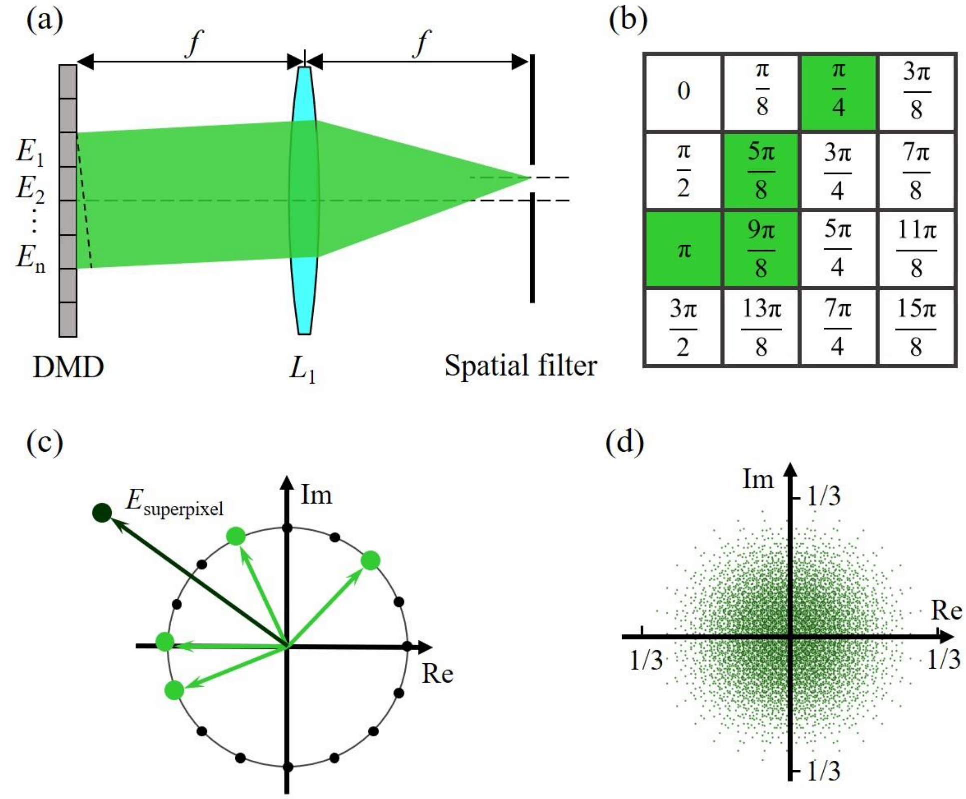

Figure. S2. Schematic of the superpixel modulation. (a) Optical setup of the superpixel modulation. (b) Distribution of phase prefactors in a single superpixel of size 4×4, (c) Sum of the four-pixel responses in (b). (d) Complex-value modes that can be constructed using a single superpixel of size 4×4. 6561 different modes can be constructed.

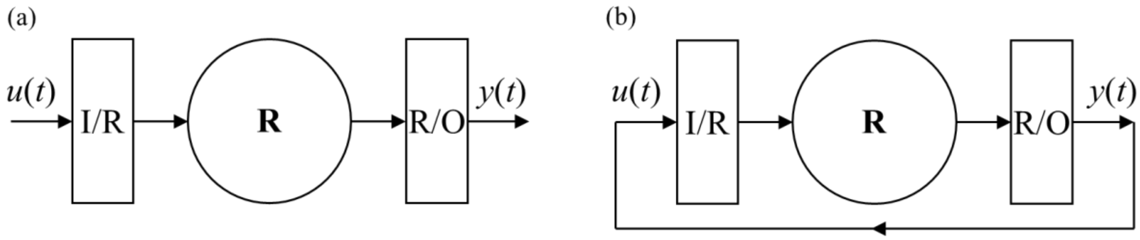

Figure. S3. Schematic illustration of two typical prediction modes. (a) One-step prediction. The Input is an error-free observation. I/R, input-to-reservoir; R, reservoir; R/O, reservoir-to-output. (b) Free-running prediction. The predicted output defines the next input.

Supplementary Note

Supplementary Note 1: Reservoir Computing

Reservoir computing (RC) is originally an RNN-based framework and is therefore suitable for temporal/sequential information processing. Specifically, RC is a unified computational framework, derived from independently proposed RNN models, such as echo state networks (ESNs) and liquid state machines (LSMs). A typical RC structure (Fig. S1) consists of three layers: input, hidden (reservoir), and output layers. The interconnections between neurons within the reservoir and between the three layers are represented by the matrices W_{res}, W_{in}, and W_{out}. Other than the output interconnections W_{out}, the neuron interconnections are all random and fixed. Actually, the connectivity status inside the reservoir needs neither to be deliberately designed nor known. Moreover, the reservoir does not need to change the interconnection weights of the neurons during training. The above features mimic the interconnections and dynamics of biological neurons in human brain.

In RC, input data are transformed into spatiotemporal patterns in a high-dimensional space by an RNN in the reservoir. Then, a pattern analysis from the spatiotemporal patterns is performed in the readout. The time evolution of the neuronal states in the reservoir and the output are respectively described as follows:

$$\mathbf{x}(t) = \alpha f_{NL}\left[W_{in}\mathbf{u}(t) + W_{res}\mathbf{x}(t-1)\right] + (1-\alpha)\mathbf{x}(t-1), \tag{1}$$

and

$$\mathbf{y}(t) = W_{out}\mathbf{x}(t), \tag{2}$$

where $\mathbf{x}$ is the state of neurons inside the reservoir; α is the leak rate that controls the speed of the dynamics of the reservoir without changing its long-term stability; f_{NL} is the nonlinear function; $\mathbf{y}$ is the output. The state evolution of the reservoir can be externally driven, which means that the RC framework is general, allowing other nonlinear dynamical systems to be used as reservoirs.

The distinctive feature of the RC is that only the output weight matrix be trained by linear regression, avoiding the complex back-propagation of traditional RNNs. This simple approach significantly reduces the learning cost of the network. Generally, the output weight matrix can be obtained by the ridge regression algorithm, denoted as

$$W_{out} = \left(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{X}\mathbf{Y}, \tag{3}$$

where $\mathbf{Y} = (\mathbf{y}(t_1), \mathbf{y}(t_2), \mathbf{y}(t_3), ...)$ is the target output matrix; $\mathbf{X} = (\mathbf{x}(t_1), \mathbf{x}(t_2), \mathbf{x}(t_3), ...)$ is the total state matrix formed by collecting the reserve pool states together at each moment of the training phase; $\mathbf{I}$ is the unit array; λ is the regularization factor, which is important to avoid overfitting.

Supplementary Note 2: Superpixel modulation

According to the mechanism of superpixel modulation, the DMD is divided into superpixels: square groups of $n \times n$ neighboring micromirrors. A pinhole filter in the form of a circular aperture is placed slightly off-axis with respect to the lens $L1$ to allow only the first-order diffraction of the DMD to pass through (Fig S2). In this case, each DMD micromirror has a phase prefactor related to its position in the superpixel in the first-order diffraction. When a micromirror is located in the a-th row and b-th column within a superpixel, its phase prefactor is $2\pi(a/n^2 + b/\mathrm{n})$. The pinhole filter blocks the high spatial frequencies, causing the pixels in a superpixel to become blurred and their information to overlap. Therefore, the response of a superpixel is the sum of the individual pixel responses and can be expressed as

$$\tilde{E}_{\mathrm{sp}} = \sum_{a=0}^{n-1}\sum_{b=0}^{n-1} \Psi_{a,b} \exp\left[2\pi i\left(\frac{a}{n^2}+\frac{b}{n}\right)\right], \tag{4}$$

where $\Psi_{a,b}$ represents the state (0 or 1) of the micromirror at the row a and column b. The superpixel technique is a very powerful tool for light modulation. When $n = 4$, the number of modulation modes of one superpixel can be as many as 6561. By turning the micromirrors on or off, the superpixel could modulate the complex-value information into the first-order diffraction of the DMD.

Supplementary Note 3: Mackey-Glass times series

The Mackey-Glass time series is one of the standard models used to test Reservoir Computing algorithms. It is derived from the famous Mackey-Glass equation in mathematical biology. The Mackey-Glass equation has been used to mimic both healthy and pathological behavior in certain biological contexts by controlling the equation's parameters. The equations are defined as:

$$Q'(t) = \frac{\beta Q(t-\tau)}{1+Q(t-\tau)^n} - \gamma Q(t). \tag{5}$$

Here, β, τ, n and γ are the parameters of the equation. The first term corresponds to a delayed response of the system, while the second term can be interpreted as a classical decay with rate γ. This time-delay differential equation, in appearance simple, displays chaotic behavior for certain parameters. In the Mackey-Glass time series prediction, the typical parameters are $\gamma = 0.1$, $\beta = 0.2$, $n = 10$ and $\tau = 17$. The maximal Lyapunov exponent in this case is $\Lambda_{\max} =0.006$.

Supplementary Note 4: Lyaponuv exponent

The chaotic motion is sensitive to initial conditions. Trajectories generated from two infinitesimally close initial values, separates exponentially over time. This phenomenon can be described by the Lyaponuv exponent. For a dynamical system $F(x)$ in a one–dimensional phase space, the derivative $|\mathrm{d}F/\mathrm{d}x|$ determines whether the two points separate or converge after the iteration. If $|\mathrm{d}F/\mathrm{d}x|>1$, the iteration will separate the two points. If $|\mathrm{d}F/\mathrm{d}x|<1$, the iteration will bring the two points closer. However, the motion tendency of the two points continuously changes throughout the iterations as the derivatives change. Therefore, an average over time (or number of iterations) is needed to describe the state of the two adjacent points as a whole. When the exponent of the separation caused by each iteration on average is denoted by Λ, the distance between two points originally separated by ε after n iterations can be expressed as

$$\varepsilon e^{n\Lambda(x_0)} = \left|F^n\left(x_0+\varepsilon\right) - F^n\left(x_0\right)\right|. \tag{6}$$

When $\varepsilon \to 0$ and $n \to \infty$, the above expression can be simplified as

$$\Lambda\left(x_0\right) = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\ln\left|\frac{dF\left(x\right)}{dx}\right|_{x=x_0}. \tag{7}$$

It can be seen that the Λ value is independent of the initial conditions and is known as the Lyapunov exponent of the dynamical system. If $\Lambda < 0$, the neighboring points eventually come together and merge into a single point, corresponding to stable immobile points and periodic motion; if $\Lambda > 0$, the neighboring points eventually separate, corresponding to the instability of the trajectories. Therefore, The Lyapunov index can be used as an important criterion for the existence of chaotic behaviors.

In general, a dynamical system in an n–dimensional phase space has the spectrum of Lyapunov exponents $\{\Lambda_1, \Lambda_2, \ldots, \Lambda_n\}$. The maximum value Λ_{max}, in the spectrum of Lyapunov exponents, named maximum Lyapunov exponent, determines the speed of the divergence of trajectories. The existence of dynamical chaos in a system can be intuitively determined by whether the maximum Lyapunov exponent is greater than zero. A positive Lyapunov exponent means that no matter how close two trajectory lines are initially, the difference between them will evolve at an exponential rate over time to be unpredictable, which is the phenomenon of chaos. Overall, the Lyapunov exponent gives a measure for the total predictability of a system, it characterizes quantitatively the rate of separation of infinitesimally close trajectories in dynamical system.

Supplementary Note 5: Characteristics of two typical prediction modes

The one-step prediction and free-running prediction are two typical prediction modes (Fig. S3). The one-step prediction mode means that only one time-step is predicted, and the input is an error-free observation for each prediction. In this mode, the error of each time-step is independent, and the prediction accuracy does not degrade significantly with the increase of the time-step length. The free-running prediction mode allows for multi-step prediction as the current output is used as input for the next step. However, the prediction accuracy of free-running prediction decreases with increasing time-step length because the current prediction error accumulates into the next step.