Non-Dissipative Random Oscillators Networks through Negative Feedback Coupling

Gioele Zerini
Computer Science Department
University of Pisa
Pisa, Italy
0009-0008-8464-5135

Andrea Ceni
Computer Science Department
University of Pisa
Pisa, Italy
andrea.ceni@unipi.it

Andrea Cossu
Computer Science Department
University of Pisa
Pisa, Italy
andrea.cossu@unipi.it

Claudio Gallicchio
Computer Science Department
University of Pisa
Pisa, Italy
claudio.gallicchio@unipi.it

Davide Bacciu
Computer Science Department
University of Pisa
Pisa, Italy
davide.bacciu@unipi.it

***Abstract*—The careful design of neural architectures is a prominent research area in machine learning. Recurrent neural networks are notoriously tricky to train, making their architectural design often a determining factor for performance. We adopt the reservoir computing approach, where the recurrent component of the network is left untrained, but its architectural design guarantees effective sequential data processing abilities. Inspired by the Random Oscillators Network (RON), we propose the Non-Dissipative RON (ND-RON) and benchmark it against popular time series datasets, showcasing its effectiveness against RON and other popular reservoir computing approaches. We provide mathematical proof of the increased ability of ND-RON to propagate information without dissipation over long time spans, surpassing other reservoir computing approaches and remaining competitive with fully trainable recurrent models while requiring one order of magnitude fewer parameters.**

***Index Terms*—reservoir computing, recurrent neural networks, dynamical systems**

I. Introduction

Sequential data processing requires the integration of information at multiple time scales. Recurrent neural networks are naturally designed to iteratively update existing representations based on new temporally-correlated information [7]. Recently, the study of recurrent neural networks has seen a resurgence, due to the discovery of efficient recurrent architectures (e.g., state-space models) [8] and due to the advantage offered by recurrent networks to linearly scale with the length of the input sequence [10].

One of the most difficult challenges in sequential data processing is the accurate propagation of information over time. This requires the recurrent model to maintain a certain degree of *stability*, thus preventing its latent representation from either exploding or vanishing. However, excessive stability prevents the system from adapting to downstream tasks. The trade-off between stability and non-dissipative propagation of information remains an open challenge.

This work has been supported by the EU-EIC project EMERGE (Grant number 101070918.)

Reservoir computing [11] offers a well-established framework for constructing RNNs where the recurrent component remains fixed and untrained. The resulting dynamical system is not adaptive, and the ability of the network to solve a given task depends only on two factors: i) the *trained* readout (usually a linear layer) taking as input the latent representation of the recurrent component and ii) the architectural bias induced by the design of the fixed recurrent module. As the training process of a linear readout is fairly standard (e.g., Ridge regression, gradient descent), the main knob to tweak is the architectural design of the recurrent network.

This is the central point of our work. Many studies in reservoir computing proved that there is a vast richness of architectural properties that can be enforced without the need to train the recurrent connections [6] (we briefly present some of the main approaches in Section II). We build on this foundation by investigating the Random Oscillators Network (RON) [4], a reservoir computing model that leverages damped oscillators working at heterogeneous frequencies as computational units. We introduce an antisymmetric coupling between the RON units, giving rise to the *Non-Dissipative RON* (ND-RON) (Figure 1). This antisymmetric coupling enhances the RON model's ability to stably propagate information over time. This is a crucial requirement, especially for time series classification tasks where the class prediction is performed at the end of the input sequence based also on information seen at the beginning of the time series. Therefore, the model needs to be able to preserve information in its hidden state, possibly for a very long time.

As we demonstrate in Section III, the antisymmetric coupling allows the information contained in the hidden state of the recurrent model to be preserved during subsequent updates. In addition to the mathematical proof of the stability of the ND-RON model, we empirically validate its performance on several time series tasks. ND-RON surpasses the performance of other reservoir computing paradigms. Importantly, adding antisymmetric coupling to the reservoir computing baselines

does not grant the same performance improvement exhibited by ND-RON.

Our model also shows competitive performance against fully-trained recurrent models while leveraging only a fraction of the trainable parameters and reducing the computational time of the training phase by orders of magnitude.

The rest of the paper is organised as follows.

Section II reviews related work on reservoir computing highlighting different architectural approaches to efficient sequential data processing.

Section III introduces the Non-Dissipative Random Oscillators Network (ND-RON), detailing its design, mathematical foundations, and the advantages of its antisymmetric coupling in preserving information over time.

Section IV presents experimental results, benchmarking ND-RON against both reservoir computing models and fully trainable recurrent networks on several time series classification tasks. Section V concludes the paper by summarizing the contributions of ND-RON, discussing its potential real-world applications, and outlining directions for future research.

II. Related Works

Reservoir computing is an efficient paradigm to design recurrent neural networks whose recurrent component is kept fixed after initialization. The *Echo State Network* (ESN) is one of the most representative approaches in reservoir computing. The state-update equation of an ESN reads:

$$\mathbf{h}_{t+1} = \sigma(\mathbf{W}\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}), \tag{1}$$

where σ is a nonlinear activation function (e.g., hyperbolic tangent) and the hidden state $\mathbf{h}_t \in \mathbb{R}^H$ at time step t encodes the history of the input sequence $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_t, \dots, \mathbf{u}_T)$, $\mathbf{u}_i \in \mathbb{R}^I$ up to time t. The matrix $\mathbf{W}$, $\mathbf{V}$ and the bias vector b are left untrained after initialization. The reservoir computing principles provide a solid mathematical foundation to choose a smart initialization for $\mathbf{W}$ and $\mathbf{V}$. The most important property is the Echo-State Property (ESP) which guarantees *fading memory*: the hidden state of the ESN depends only on the recent past. Although a sufficient condition for the ESP exists, in practice a rule of thumb based on the spectral radius of $\mathbf{W}$ has proven to be effective. After random uniform initialization, the matrix $\mathbf{W}$ is rescaled via $\mathbf{W} \leftarrow \rho \frac{\mathbf{W}}{\max|\text{eig}(\mathbf{W})|}$, where $\max|\text{eig}(\mathbf{W})|$ is the largest absolute value among the eigenvalues of $\mathbf{W}$ and $\rho < 1$ is a scalar hyperparameter that controls the degree of fading memory of the system.

The input-to-hidden matrix $\mathbf{V}$ and the bias $\mathbf{b}$ are randomly and uniformly initialized in a fixed interval $[-a, a]$. The matrix $\mathbf{V}$ is then scaled by a hyper-parameter ν called *input scaling*: $\mathbf{V} \leftarrow \nu\mathbf{V}$.

The Leaky ESN introduces a leakage term in the state-update equation controlled by a scalar hyper-parameter $\alpha > 0$:

$$\mathbf{h}_{t+1} = (1-\alpha)\mathbf{h}_t + \alpha\sigma(\mathbf{W}\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}). \tag{2}$$

Leaky ESN is an effective model for time series prediction tasks and, as any reservoir computing model, a very efficient one as its recurrent component does not undergo any training phase.

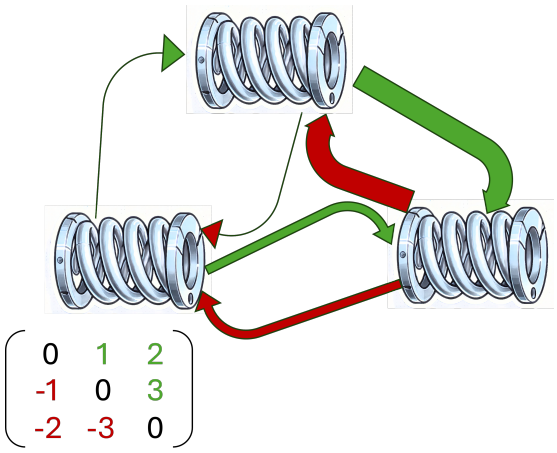

Fig. 1. Physical systems in negative feedback coupling. The matrix regulating the connections between the mechanical springs is antisymmetric ($\mathbf{A} = -\mathbf{A}^T$). The width of the arrows represents the connection strength, the positive feedback is green, and the negative feedback is red. Each spring receives a signal from the other springs of the same magnitude as the signal it sends but of inverse sign.

Similar to our work, the *Euler State Network* (EuSN) introduces a variant of the ESN by leveraging antisymmetric matrices. The state update of the EuSN follows the equation:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \tau\sigma(\mathbf{A}\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}), \tag{3}$$

where τ is the step size of integration of the underlying continuous dynamical system and it is treated as a scalar hyper-parameter. The EuSN parametrizes $\mathbf{A}$ as $(\mathbf{W} - \mathbf{W}^T) - \delta\mathbf{I}$, where δ is a diffusion coefficient to stabilize the numerical integration and $\mathbf{W} - \mathbf{W}^T$ is antisymmetric by construction. *Antisymmetric Recurrent Neural Networks* (A-RNNs) [5] are the fully-trainable version of the EuSN, where the parameters $\mathbf{W}, \mathbf{V}, \mathbf{b}$ of the network are trained via backpropagation through time.

We build our work based on a reservoir computing model called the *Random Oscillators Network*, which is a network where each unit behaves as a damped harmonic oscillator. The discrete-time state-update equation of RON reads:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \tau\mathbf{z}_{t+1}, \tag{4}$$

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \tau(\sigma(\mathbf{W}\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}) - \boldsymbol{\gamma} \odot \mathbf{h}_t - \boldsymbol{\varepsilon} \odot \mathbf{z}_t), \tag{5}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^H$ and $\boldsymbol{\varepsilon} \in \mathbb{R}^H$ control the stiffness and the damping of each oscillator unit, respectively. The variable $\mathbf{z}$ is the oscillator's velocity, while the variable $\mathbf{h}$ represents the oscillator's position.

All reservoir computing models we consider in this work employ a linear predictor (the readout) trained in closed form

with Ridge regression. The linear predictor takes as input the last hidden state $\mathbf{h}_T$.

III. Non-Dissipative Random Oscillators Network

Inspired by theoretical insights from the literature [9], ensuring that the use of antisymmetric weight constraints promotes minimal dissipation of the information flow, we introduce the discrete-time state-update equation of ND-RON, as follows:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + \tau \mathbf{z}_{t+1}, \tag{6}$$

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \tau(\sigma(\mathbf{A}\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b}) - \boldsymbol{\gamma} \odot \mathbf{h}_t - \boldsymbol{\varepsilon} \odot \mathbf{z}_t), \tag{7}$$

where $\mathbf{A} = (\mathbf{W} - \mathbf{W}^T) - \delta\mathbf{I}$, is an antisymmetric matrix with a diffusion term $-\delta\mathbf{I}$. We use $\sigma = \tanh$ as a nonlinear activation function.

When introducing a new RNN model, it is essential to ensure that its dynamics remain well-behaved over time, avoiding the risk of exploding states that can arise from instability. One of the most effective ways to assess this is through linear stability analysis, a standard method for understanding the behavior of a dynamical system in the vicinity of its fixed points. For discrete-time RNNs, this involves studying how small perturbations evolve from one time step to the next, which is described by the eigenvalues of the system's Jacobian matrix, i.e. the eigenvalues of the matrix of all the partial derivatives. A key quantity in this context is the spectral radius, the largest absolute value of the matrix eigenvalues. Stability requires that this spectral radius remains less than or equal to one; when it exceeds one, perturbations grow uncontrollably, leading to unstable behavior. This approach provides a practical framework for identifying necessary conditions for stability and analyzing how different network parameters shape the dynamics of the system.

A. Necessary condition for stability of ND-RON

Let us denote $\mathbf{H}_t = \begin{pmatrix} \mathbf{h}_t \\ \mathbf{z}_t \end{pmatrix}$. Then, the ND-RON model can be defined by the input-driven state-update equation $\mathbf{H}_{t+1} = G(\mathbf{H}_t, \mathbf{u}_{t+1})$, where $G : \mathbb{R}^{2H} \times \mathbb{R}^I \to \mathbb{R}^{2H}$ is defined by eqs. (6) - (7). In the remainder of this section, we assume that $\mathrm{diag}(\boldsymbol{\varepsilon}) = \varepsilon\mathbf{I}$ and $\mathrm{diag}(\boldsymbol{\gamma}) = \gamma\mathbf{I}$ are scalar matrices. The Jacobian of the G map computed on $(\mathbf{H}_t, \mathbf{u}_{t+1})$, denoted with $\mathbf{J}_k$, reads:

$$\mathbf{J}_t = \begin{bmatrix} \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{h}_t} & \frac{\partial \mathbf{h}_{t+1}}{\partial \mathbf{z}_t} \\ \frac{\partial \mathbf{z}_{t+1}}{\partial \mathbf{h}_t} & \frac{\partial \mathbf{z}_{t+1}}{\partial \mathbf{z}_t} \end{bmatrix} = \tag{8}$$

$$= \begin{bmatrix} \mathbf{I} + \tau^2\mathbf{M}_t & \tau\big(1-\tau\varepsilon\big)\mathbf{I}\big) \\ \tau\mathbf{M}_t & \big(1-\tau\varepsilon\big)\mathbf{I} \end{bmatrix}, \tag{9}$$

where

$$\mathbf{M}_t = \mathbf{D}_t(\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I}) - \gamma\mathbf{I}, \tag{10}$$

$$\mathbf{D}_t = \mathrm{diag}\big(\sigma'((\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I})\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b})\big). \tag{11}$$

Notice that, when using $\sigma = \tanh$ as nonlinearity (evaluating at the origin without input and without bias) then the diagonal matrix of (11) is the identity matrix. Therefore, the Jacobian reads:

$$\mathbf{J}_0 = \begin{bmatrix} \mathbf{I} + \tau^2(\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I} - \gamma\mathbf{I}) & \tau\big(1-\tau\varepsilon\big)\mathbf{I} \\ \tau(\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I} - \gamma\mathbf{I}) & \big(1-\tau\varepsilon\big)\mathbf{I} \end{bmatrix}. \tag{12}$$

In the proposition below we estimate the location of the eigenvalues of $\mathbf{J}_0$.

Proposition 1. *If μ is an eigenvalue of $\mathbf{J}_0$, then μ is at distance at most $\tau \max\{|1-\tau\varepsilon|, \sqrt{\lambda_{\max}^2 + (\delta+\gamma)^2}\}$ from the set of values $\{1-\tau\varepsilon\} \cup \{1 - \tau^2(\delta+\gamma \pm i\lambda) : i\lambda$ is an eigenvalue of $\mathbf{W} - \mathbf{W}^\top\}$, where λ_{max} is the spectral radius of the skew-symmetric matrix $\mathbf{W} - \mathbf{W}^\top$.*

A necessary condition for ND-RON stability is that the set of these values is completely contained inside the unit circle. In the following theorem, we formally write a few inequalities that ND-RON's hyperparameters must satisfy for stability.

Theorem 1. *Assume an ND-RON model with $\delta, \tau, \varepsilon, \gamma > 0$. If the ND-RON of eqs* (6)-(7) *is stable, then*

$$\varepsilon \leq \frac{2}{\tau}, \tag{13}$$

$$\delta + \gamma \leq \frac{2}{\tau^2}. \tag{14}$$

Moreover, each eigenvalue $i\lambda$ of the skew-symmetric matrix $\mathbf{W} - \mathbf{W}^\top$ must have modulus upper-bounded as follows:

$$|\lambda| \leq \tau\sqrt{(\delta+\gamma)\big[2 - \tau^2(\delta+\gamma)\big]}. \tag{15}$$

Proof. A necessary condition for ND-RON stability is that the set of values $\{1-\tau\varepsilon\} \cup \{1 - \tau^2(\delta+\gamma \pm i\lambda) : i\lambda$ is an eigenvalue of $\mathbf{W} - \mathbf{W}^\top\}$, derived in Proposition 1, is entirely contained inside the unit circle. The farthest of these points from the origin is either $1-\tau\varepsilon$ or $1-\tau^2(\delta+\gamma+i\lambda_{\max})$, where $\lambda_{\max}$ is the spectral radius of the skew-symmetric matrix $\mathbf{W} - \mathbf{W}^\top$. It follows that a necessary condition for ND-RON stability is:

$$\max\{|1-\tau\varepsilon|, \sqrt{\big(1-\tau^2(\delta+\gamma)\big)^2 + \lambda_{\max}^2}\} \leq 1. \tag{16}$$

In particular, it must hold that $|1-\tau\varepsilon| \leq 1$, and $\sqrt{\big(1-\tau^2(\delta+\gamma)\big)^2 + \lambda_{\max}^2} \leq 1$. The first implies that $\varepsilon \leq \frac{2}{\tau}$, while the second implies that $\big(1-\tau^2(\delta+\gamma)\big)^2 + \lambda_{\max}^2 \leq 1$, i.e. that $\lambda_{\max}^2 \leq \tau^2(\delta+\gamma)\big[2 - \tau^2(\delta+\gamma)\big]$, from which it follows that $2 - \tau^2(\delta+\gamma) \geq 0$. Moreover, each eigenvalue $i\lambda$ of $\mathbf{W} - \mathbf{W}^\top$ satisfies $|\lambda| \leq |\lambda_{\max}| \leq \tau\sqrt{(\delta+\gamma)\big[2 - \tau^2(\delta+\gamma)\big]}$. □ □

We can see from eq. (15) that, for a given setting of τ, γ, positive values of the diffusion term δ can facilitate the necessary condition of stability to hold. In this sense, larger values of δ can help the stability of the ND-RON, but at the cost of a higher dissipation rate of the information flow. On the other hand, values too large of δ, e.g. $\delta \geq \frac{2}{\tau^2} - \gamma$, can cause instabilities. Therefore, δ must be carefully tuned for the specific task at hand.

We use the insights from Theorem 1 to guide the model selection of ND-RON in the experimental section.

B. Non-dissipativity of ND-RON

In the previous section, we focused on the stability of ND-RON via the estimation of the spectral radius in the worst-case scenario. However, it is useful to estimate how the volume of information evolves in the hidden space while processing a time series. In this regard, we make use of the (pseudo) Local Lyapunov Exponents (LLEs) [2, 1]. As for the spectrum of eigenvalues, there is a spectrum of LLEs, one for each dimension of a dynamical system. Each LLE quantifies the rate at which nearby trajectories either converge (when negative) or diverge (when positive), for each dimension of the hidden state space. In particular, the i-th LLE, denoted LLE_i, is defined as the logarithmic average of the absolute value of the eigenvalues of the Jacobian evaluated at all times $t = 1, \dots, T$, where T is the length of the time series, i.e. $\text{LLE}_i = \frac{1}{T}\sum_{t=1}^{T} \ln|\lambda_i(\mathbf{J}_t)|$, where $|\lambda_i(\mathbf{J}_t)|$ denotes the i-th absolute eigenvalue of $\mathbf{J}_t$, assuming to have all the eigenvalues ordered according to their absolute values. Together, the LLEs give us a first-order approximation of how a volume of information in the hidden state evolves under the driving of the input time series. Non-dissipation is achieved for values of LLEs that stay around the value of zero: a balance between contraction and expansion. This means that the volume of information is more or less preserved on average during the T time steps of input processing. In the following, we analytically derive bounds on the LLEs of an input-driven ND-RON, proving that the ND-RON is non-dissipative by architectural design.

Theorem 2. *Assume an ND-RON model with $\delta, \tau, \varepsilon, \gamma > 0$. The LLEs of an ND-RON are bounded as follows:*

$$\ln(1-\tau\xi) \leq \textit{LLE}_i \leq \ln(1+\tau\xi), \tag{17}$$

where $\xi = \max\{\varepsilon, \tau\eta\} + \max\{|1-\tau\varepsilon|, \eta\}$, and $\eta = \rho(\mathbf{W} - \mathbf{W}^\top) + \delta + \gamma$. In particular, for the case of small τ and assuming $1 < \eta < \frac{\varepsilon}{\tau}$, we obtain that all the LLEs of ND-RON are approximately contained in the following interval centered at zero that involves all the characteristic hyper-parameters of ND-RON:

$$\textit{LLE}_i \in \Big[-\tau\big(\varepsilon + \rho(\mathbf{W} - \mathbf{W}^\top) + \delta + \gamma\big)\ , \tag{18}$$

$$\tau\big(\varepsilon + \rho(\mathbf{W} - \mathbf{W}^\top) + \delta + \gamma\big)\Big]. \tag{19}$$

Proof. We start writing the input-driven Jacobian of (9) as $\mathbf{J}_t = \mathbf{I} + \tau\mathbf{E}_t$ where

$$\mathbf{E}_t = \begin{bmatrix} \tau\mathbf{M}_t & \big(1-\tau\varepsilon\big)\mathbf{I} \\ \mathbf{M}_t & -\varepsilon\mathbf{I} \end{bmatrix}. \tag{20}$$

Bauer-Fike theorem [3] implies that, if $\lambda_i(\mathbf{J}_t)$ is an eigenvalue of $\mathbf{J}_t$, then

$$|\lambda_i(\mathbf{J}_t) - 1| \leq \tau \left|\left| \begin{bmatrix} \tau\mathbf{M}_t & \big(1-\tau\varepsilon\big)\mathbf{I} \\ \mathbf{M}_t & -\varepsilon\mathbf{I} \end{bmatrix} \right|\right| \tag{21}$$

$$\leq \tau\Big(\max\{\tau||\mathbf{M}_t||, \varepsilon\} + \max\{||\mathbf{M}_t||, |1-\tau\varepsilon|\}\Big). \tag{22}$$

Now, recalling that

$$\begin{aligned} \mathbf{M}_t =& \mathbf{D}_t(\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I}) - \gamma\mathbf{I}, \\ \mathbf{D}_t =& \text{diag}\big(\sigma'((\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I})\mathbf{h}_t + \mathbf{V}\mathbf{u}_{t+1} + \mathbf{b})\big), \end{aligned}$$

we can upper-bound $||\mathbf{M}_t|| = ||\mathbf{D}_t(\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I}) - \gamma\mathbf{I}|| \leq ||\mathbf{D}_t(\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I})|| + \gamma \leq ||\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I}|| + \gamma$, where the last inequality follows from the fact that $||\mathbf{D}_t|| \leq 1$ due to the choice of $\sigma = \tanh$ as nonlinearity. Similarly, we can upper-bound $||\mathbf{W} - \mathbf{W}^\top - \delta\mathbf{I}|| \leq ||\mathbf{W} - \mathbf{W}^\top|| + \delta = \rho(\mathbf{W} - \mathbf{W}^\top) + \delta$, where the last equality follows from the fact that $\mathbf{W} - \mathbf{W}^\top$ is skew-symmetric. Therefore, putting all together, and denoting $\eta = \rho(\mathbf{W} - \mathbf{W}^\top) + \delta + \gamma$, we have the following bound:

$$|\lambda_i(\mathbf{J}_t) - 1| \leq \tau \underbrace{\big(\max\{\tau\eta, \varepsilon\} + \max\{\eta, |1-\tau\varepsilon|\}\big)}_{=\xi}. \tag{23}$$

This implies that $1 - \tau_\xi \leq |\lambda_i(\mathbf{J}_t)| \leq 1 + \tau\xi$, for all eigenvalues of $\mathbf{J}_t$ and for each t. Therefore, applying the logarithm we get $\ln(1-\tau_\xi) \leq \ln|\lambda_i(\mathbf{J}_t)| \leq \ln(1+\tau\xi)$, for all eigenvalues of $\mathbf{J}_t$ and for each t. Recalling the definition of LLEs, i.e. $\text{LLE}_i = \frac{1}{T}\sum_{t=1}^{T} \ln|\lambda_i(\mathbf{J}_t)|$, we can conclude that

$$\ln(1-\tau\xi) \leq \frac{1}{T}\sum_{t=1}^{T} \ln|\lambda_i(\mathbf{J}_t)| \leq \ln(1+\tau\xi). \tag{24}$$

Moreover, for the case of small τ, and assuming $1 < \eta < \frac{\varepsilon}{\tau}$, we get that $\xi = \varepsilon + \rho(\mathbf{W} - \mathbf{W}^\top) + \delta + \gamma$. Therefore, we obtain the approximations $\ln(1 \pm \tau\xi) \approx \pm\tau\xi = \pm\tau\big(\varepsilon + \rho(\mathbf{W} - \mathbf{W}^\top) + \delta + \gamma\big)$, for both the lower and the upper bound. □ □

Theorem 2 gives us a recipe for the setting of the hyper-parameters of ND-RON. Suppose to process a time series of length T. An initial condition infinitesimally close to the origin, after T time steps, will be at most at distance $e^{T\max_i\{\text{LLE}_i\}}$ from the origin. In particular, we can afford to set a relatively large spectral radius for $\mathbf{W} - \mathbf{W}^\top$ and values of $\gamma, \varepsilon, \delta$, as long as τ is small enough to counterbalance the term $T\tau(\varepsilon + \rho(\mathbf{W} - \mathbf{W}^\top) + \gamma + \delta)$. As an example, for a time series of length $T = 300$, setting $\tau = 10^{-2}$, and $\gamma, \varepsilon, \delta$, $\rho(\mathbf{W} - \mathbf{W}^\top)$ around 1, we get that after T time steps the last hidden state will have in the worst-case a norm of approximately $e^{12} \approx 10^5$, which is still reasonable to perform classification at the last hidden state. We design the model selection procedure to avoid hidden-state norms that become too large or too small.

IV. Experiments

We tested the performance of ND-RON against several reservoir computing approaches like the Leaky ESN, the

TABLE I

AVERAGE ACCURACY AND STANDARD DEVIATION OVER 5 RUNS OBTAINED FROM EACH RESERVOIR COMPUTING MODEL AND TRAINABLE RECURRENT MODEL ON THE CONSIDERED BENCHMARKS. BOLD DENOTES THE BEST PERFORMANCE AMONG ND-RON AND THE OTHER RESERVOIR COMPUTING MODELS. UNDERLINE DENOTES THE BEST PERFORMANCE AMONG ND-RON AND THE FULLY TRAINABLE MODELS.

	Leaky-ESN	**aESN**	**RON**	**EuSN**	**ND-RON** (ours)	**GRU** (trainable)	**A-RNN** (trainable)
Adiac	$0.6928_{0.0116}$	$0.6793_{0.0099}$	$0.7090_{0.0125}$	$0.4221_{0.0154}$	$\mathbf{0.7315_{0.0043}}$	$0.6322_{0.0235}$	$\underline{0.7749_{0.0119}}$
sMNIST	$0.8798_{0.0022}$	$0.8803_{0.0040}$	$0.9568_{0.0020}$	$0.6321_{0.0401}$	$\mathbf{0.9627_{0.0006}}$	$\underline{0.9686_{0.0165}}$	$0.9598_{0.0060}$
psMNIST	$0.8284_{0.0272}$	$0.9067_{0.0054}$	$0.8760_{0.0050}$	$0.4380_{0.0244}$	$\mathbf{0.9106_{0.0065}}$	$0.8790_{0.0085}$	$\underline{0.9118_{0.0039}}$
Trace	$0.9640_{0.0206}$	$0.9740_{0.0080}$	$0.9920_{0.0075}$	$0.8200_{0.0228}$	$\mathbf{0.9940_{0.0049}}$	$\underline{0.9979_{0.0040}}$	$0.7960_{0.0206}$
Mallat	$0.9011_{0.0097}$	$0.9179_{0.0162}$	$0.9142_{0.0274}$	$0.3364_{0.0306}$	$\underline{\mathbf{0.9262_{0.0179}}}$	$0.2142_{0.0021}$	$0.6803_{0.0521}$
Libras	$0.7911_{0.0083}$	$0.7489_{0.0147}$	$0.7900_{0.0151}$	$0.7722_{0.0136}$	$\mathbf{0.7956_{0.0166}}$	$\underline{0.8344_{0.0221}}$	$0.8044_{0.0266}$

antisymmetric ESN and the EuSN. As an ablation experiment, we also compared with the RON model without the antisymmetric coupling, to showcase the benefit of the non-dissipative propagation of information. Finally, we provide results also for fully-trainable recurrent models, where the recurrent weight matrices are learned via backpropagation through time. In particular, we used the Gated Recurrent Unit (GRU) network and the A-RNN network. The latter also leverages antisymmetric coupling between the units.

We used popular time series benchmarks for sequence classification: Adiac, Trace, Mallat, and Libras, publicly available from timeseriesclassification.com. *Adiac* is a dataset containing sinusoidal-like patterns extracted from images with the objective of identifying 37 different diatoms classes. Input sequences are composed of 176 time steps and the training and test set contain 390 and 391 sequences, respectively. *Trace* is a dataset to recognize 4 different classes of instrument failures in a nuclear power plant. The dataset contains 100 sequences for both training and test, all of length 275. *Mallat* is a signal processing dataset composed of 8 different classes of input signals of length 1024, with 55 examples reserved for training and 2345 for test. This dataset is especially challenging due to the very low number of examples. We expect fully-trainable models to struggle to learn meaningful patterns as they will likely tend to overfit. We will show the benefit of the reservoir computing approach in this context. *Libras* is a dataset that requires to classify 15 different hand movements from the Brazilian sign language. The dataset contains 180 training sequences and 180 test sequences, all of length 45.

We also leveraged the popular sequential MNIST (sMNIST) and permuted sequential MNIST (psMNIST) datasets as popular benchmarks to test the effectiveness of the models for the long-range propagation of information. They both take each MNIST image one pixel at a time, leading to sequences of 784 time steps. psMNIST randomly permutes all pixels according to a fixed random permutation before processing the images. The objective is to classify each image into one of the possible ten digits. Numerical explosion or vanishing of the hidden state of the recurrent models would prevent solving both benchmarks as they require propagating information far into the future.

A. Experimental setup

We split all datasets into training, validation and test sets, leveraging the validation set to tune the hyperparameters before conducting the model assessment on the test set. Across all datasets, the reservoir computing models evaluated in this work (Leaky ESN, aESN, RON, EuSN and ND-RON) were configured to use the same number of hidden units, ensuring a fair comparison of their structural characteristics. Specifically, the hidden layer sizes were set to: 50 units for Trace, 100 units for ADIAC and Mallat, 150 units for LIBRAS and 1000 units for sMNIST and psMNIST.

For the fully-trainable models we used the following number of hidden units:

- GRU: 28 units for ADIAC, 14 units for Mallat, 23 units for LIBRAS, 6 units for Trace and 55 units for sMNIST and psMNIST;
- A-RNN: 44 units for ADIAC, 24 units for Mallat, 39 units for LIBRAS, 11 units for Trace and 94 units for sMNIST and psMNIST.

Upon paper acceptance, we will publicly release the code to reproduce all experiments as well as the full set of hyperparameters used during the model selection phase and the best hyperparameters found for the model assessment phase. We will also release the proofs of all theorems which, due to space limit, we were unable to publish in this version.

B. Results

Table I reports the average accuracy of ND-RON against all reservoir computing models. Table I provides the average accuracy of ND-RON against fully-trainable models. To run a fair assessment, we used a comparable number of trainable parameters for all models: Adiac uses 3.7k parameters, sMNIST and psMNIST 10k, Trace 200, Mallat 800 and Libras 2.25k.

ND-RON shows an excellent performance across all benchmarks, surpassing all reservoir computing models, sometimes even by a large margin. Of particular interest is the impressive performance gain of ND-RON with respect to the EuSN: around 30 points on Adiac and sMNIST, around 50 points on psMNIST and around 60 points on Mallat. The EuSN is the closest model to ND-RON in terms of architectural bias, as it also employs antisymmetric coupling as well as an Euler discretization method. However, the oscillatory units in ND-RON

are much more effective than the EuSN units, highlighting the importance of heterogeneous oscillatory dynamics in the latent space. This is also confirmed by the results of RON, where the antisymmetric coupling is missing. RON retains the second-best performance, although ND-RON clearly outperforms it. The non-dissipative propagation of information through the antisymmetric coupling boosts RON's performance. The Leaky ESN and its antisymmetric version remain top-performing reservoir computing methods, achieving good performance on almost all benchmarks. As they do not require numerical integration in their forward pass, they still represent a valid trade-off when one needs a simple but effective computational unit.

With respect to the fully trainable models (Table I), ND-RON remains competitive in terms of average accuracy, outperforming GRU and A-RNN on Mallat. In fact, due to the very low number of training examples in Mallat, fully trainable models are not able to learn meaningful patterns. On the contrary, on sMNIST and psMNIST fully-trainable models reach high values of accuracy. This is aligned with known results on the importance of training for long-term correlations [12].

Even though untrained in the recurrent dynamics, ND-RON achieves a very close performance also in these datasets, surpassing fully-trainable models like the A-RNN in sMNIST and the GRU on psMNIST. Crucially, ND-RON achieves this excellent performance **while saving orders of magnitude of training time**. In fact, ND-RON only trains a linear readout on top of the latent states, while GRU and A-RNN require backpropagation through time. This results in only *a few seconds of training on sMNIST for reservoir computing models like ND-RON* (around 66 seconds) against *around 2 hours* (depending on the number of epochs) of the fully trainable models.

Finally, we verify that all the configurations found by the model selection satisfy the necessary conditions for the ND-RON stability found in Section III-A. Namely, on all datasets $\varepsilon \leq \frac{2}{\tau}$ and $\delta + \gamma \leq \frac{2}{\tau^2}$.

V. Conclusion and Future Work

We introduced ND-RON, a recurrent neural network based on the reservoir computing paradigm which allows efficient and non-dissipative propagation of information over time. ND-RON relies on a fixed recurrent component, initialized following the reservoir computing guidelines. In addition, it employs an antisymmetric coupling across the hidden units that strikes a good balance between stability and expressive power. We provided theoretical guarantees on the non-dissipation of information in ND-RON. Further, we empirically validated its effectiveness on several time series benchmarks, showing how ND-RON surpasses other popular reservoir computing approaches while remaining competitive (or sometimes even better) than fully-trainable recurrent models.

The antisymmetric coupling present in ND-RON mirrors popular coupling configurations present in the physical world. Moreover, while backpropagation through time and fully trainable models cannot be implemented in physical systems, ND-RON is compatible with neuromorphic or alternative hardware substrates. Deploying ND-RON in the real world is would provide an energy-efficient and sustainable solution for time series processing at the edge and under tight latency constraints.

References

[1] Henry DI Abarbanel, Reggie Brown, and Matthew B Kennel. Local lyapunov exponents computed from observed data. *Journal of Nonlinear Science*, 2, 1992.

[2] B Bailey and Douglas W Nychka. Local lyapunov exponents: predictability depends on where you are. *Nonlinear Dynamics and Economics*, 1997.

[3] F. L. Bauer and C. T. Fike. Norms and exclusion theorems. *Numerische Mathematik*, 2(1):137–141, 1960.

[4] Andrea Ceni, Andrea Cossu, Maximilian W. Stölzle, Jingyue Liu, Cosimo Della Santina, Davide Bacciu, and Claudio Gallicchio. Random Oscillators Network for Time Series Processing. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238, pages 4807–4815. PMLR, April 2024.

[5] Bo Chang, Minmin Chen, Eldad Haber, and Ed H. Chi. AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks. In *International Conference on Learning Representations*, 2018.

[6] Andrea Cossu, Andrea Ceni, Davide Bacciu, and Claudio Gallicchio. Sparse Reservoir Topologies for Physical Implementations of Random Oscillators Networks. In *Proceedings of the 4th International Conference on AI-ML Systems*, AIMLSystems '24, pages 1–9, New York, NY, USA, 2025. Association for Computing Machinery.

[7] Jeffrey L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990.

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *CoRR*, abs/2312.00752, 2023.

[9] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse problems*, 34(1):014004, 2017.

[10] Weizhe Hua, Zihang Dai, Hanxiao Liu, and Quoc Le. Transformer quality in linear time. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9099–9117. PMLR, 17–23 Jul 2022.

[11] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[12] Antonio Orvieto, Samuel L. Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting Recurrent Neural Networks for Long Sequences. In *Proceedings of the 40th International Conference on Machine Learning*, pages 26670–26698. PMLR, July 2023.