Random Oscillators Network for Time Series Processing

Andrea Ceni* University of Pisa
Andrea Cossu* University of Pisa
Maximilian Stölzle TU Delft
Jingyue Liu TU Delft
Cosimo Della Santina TU Delft
Davide Bacciu University of Pisa
Claudio Gallicchio University of Pisa

Abstract

We introduce the Random Oscillators Network (RON), a physically-inspired recurrent model derived from a network of heterogeneous oscillators. Unlike traditional recurrent neural networks, RON keeps the connections between oscillators untrained by leveraging on smart random initialisations, leading to exceptional computational efficiency. A rigorous theoretical analysis finds the necessary and sufficient conditions for the stability of RON, highlighting the natural tendency of RON to lie at the edge of stability, a regime of configurations offering particularly powerful and expressive models. Through an extensive empirical evaluation on several benchmarks, we show four main advantages of RON. 1) RON shows excellent long-term memory and sequence classification ability, outperforming other randomised approaches. 2) RON outperforms fully-trained recurrent models and state-of-the-art randomised models in chaotic time series forecasting. 3) RON provides expressive internal representations even in a small parametrisation regime making it amenable to be deployed on low-powered devices and at the edge. 4) RON is up to two orders of magnitude faster than fully-trained models.

1 INTRODUCTION

Recurrent Neural Networks (RNNs) represent one of the cornerstones of Machine Learning (ML) since its early beginning (Elman, 1990). The key characteristic of RNNs is to encode temporal information present in the input into the RNN parameters and its hidden state. RNNs can be directly implemented in physical systems as information-processing units (Hauser et al., 2011; Nakajima et al., 2015; Wright et al., 2022; Lee et al., 2022). This type of RNN leverages its intrinsic nonlinear dynamics as a resource to empower analogical systems with flexible computation capabilities.
Following the footprints of these works, we investigate a physically-inspired RNN derived from a continuous-time ODE describing a network of oscillators. We cast this model into the Reservoir Computing (RC) framework (Lukoševičius and Jaeger, 2009) and we dub it Random Oscillators Network (RON). RC is a flexible paradigm of training RNNs which leverages an untrained nonlinear system (the reservoir) mapping input signals into an high dimensional space, and adapting a linear readout layer only, thus resulting in particularly fast training (Nakajima and Fischer, 2021).
Oscillators represent an archetypal dynamical behavior ubiquitous in nature that can be found in chemical reactions such as Belousov–Zhabotinsky, electronic circuits like amplifiers, business cycles, central nervous system diseases like Parkinson's, pendulum clocks, fireflies' light pulses, and others (Pikovsky et al., 2002). This motivates us to inspect to what extent an *untrained* ensemble of input-driven heterogeneous oscillators can be exploited for time series processing. We take inspiration from the coupled oscillatory RNN (coRNN) (Rusch and Mishra, 2023), a recently proposed deep learning model based on oscillators. We extend this model to the case of heterogeneous oscillators coupled only in the position variable, but being uncoupled in the damping terms. Also, our RON follows the principles of RC by leveraging untrained hidden parameters. Our contributions can be summarised as follows:

- We introduce the RON model (Section 2.3), and prove it to be a generalisation of a popular RC model used in many time series processing tasks,

* Equal contribution

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

like classification (Jaeger et al., 2007), and forecasting (Jaeger and Haas, 2004).

- A theoretical analysis of stability (Section 3) reveals that the linearised RON exhibits a tendency towards the identity mapping, a widely recognised important feature in the context of deep learning (He et al., 2016), which in turns equips the RON with an architectural bias towards the edge of stability (Bertschinger and Natschläger, 2004). Precisely, we provide an estimation of the Jacobian's norm of RON (Theorem 3.1), a characterisation of its eigenspectrum (Theorem 3.2), and we derive sufficient and necessary conditions (Proposition 3.3) for the existence and uniqueness of an asymptotically stable solution for RON.
- We test RON on several time series benchmarks (Sections 5-6). In time series classification, RON strongly outperforms randomised models on long-term dependencies tasks and suffers only a slight degradation in performance with respect to fully-trained models, while being two orders of magnitude more efficient. On classification tasks requiring medium to short-term memory with a relatively small number of parameters, RON outperforms all models, thus demonstrating remarkably expressive internal representations. On chaotic systems forecasting, RON outperforms both state-of-the-art randomised and fully-trained recurrent models.

2 FROM HARMONIC OSCILLATOR TO RANDOM OSCILLATORS NETWORKS

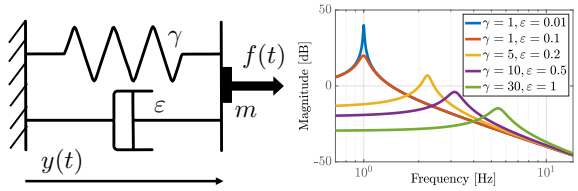
We first provide a gentle introduction to the damped harmonic oscillator, framing it in the context of fading memory systems. Then we introduce a recurrent layer based on heterogeneous oscillators which takes advantage of a fixed nonlinear dynamical system, called *Random Oscillators Network (RON)*, and a fully trained version, called *heterogeneous coupled oscillatory RNN (hcoRNN)*. Finally, we link our RON model to a well-known RC model called Leaky-ESN.

2.1 Damped harmonic oscillator

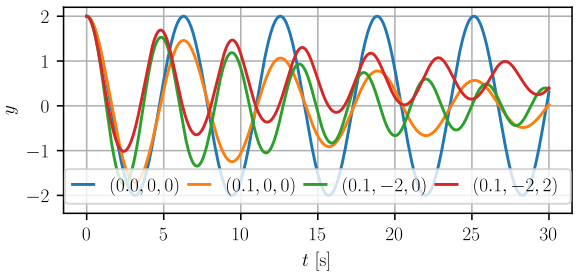
The harmonic oscillator is a mechanical oscillator (Fig. 1(a)) with constant mass and linear damping. The harmonic oscillator is described by the following ODE, where we also include a time-varying forcing $f(t) \in \mathbb{R}$:

$$\ddot{y} = \frac{1}{m}\left(f(t) - \gamma\, y - \varepsilon\, \dot{y}\right). \tag{1}$$

Here, $y \in \mathbb{R}$, $\dot{y}$, and $\ddot{y}$ are defined as the oscillator position, its velocity, and acceleration respectively. Furthermore, m is the point mass, $\gamma \in [0, \infty)$ the constant stiffness, and $\varepsilon \in [0, \infty)$ the damping coefficient. We consider in the following the case with $m = 1$. Such a



(a) Schematic (b) Bode plot



(c) Evolution of a single oscillator a selection of (ε, W, b)

Figure 1: **Panel (a):** Schematic of a driven, damped harmonic oscillator. **Panel (b):** Bode plot for harmonic oscillators with different stiffnesses γ and damping coefficients ε. **Panel (c):** Time evolution of a single harmonic oscillator with $\gamma = 1$ driven by $f(t) = \tanh(\mathrm{W}\, y + \mathrm{b})$ for various values of ε, W, and b.

simple system has already several properties that make it ideal to serve as a fundamental cell of our learning strategy.

First, different choices of γ and ε give rise to quite different transient behaviors, as shown in Fig. 1(c). This reflects in the capability of each oscillator to isolate different portions of the spectrum of the input signal $f(t)$ as can be seen in the Bode plot of the configuration y in Fig. 1(b). Also, note that the velocity $\dot{y}$ can serve as a high pass counterpart to y. Second, these oscillators have *fading memory*. Indeed, the energy of (1) is $E = m\dot{y}^2/2 + \gamma y^2/2$. This is a strictly decreasing function of time for $\varepsilon > 0$, since $\dot{E} = -\varepsilon \dot{y}^2$. In rough terms, the dynamical system forgets its initial condition after a transient.

Still, the linear nature of (1) may be not expressive enough. We thus consider a simple variation on this system obtained by selecting the forcing as the output of a one-dimensional neuron with a hyperbolic tangent as the activation function $f(t) = \tanh(\mathrm{W}\, y + \mathrm{b})$ where $\mathrm{W}, \mathrm{b} \in \mathbb{R}$ (see green and red lines in Fig. 1(c)). This small change substantially enriches the dynamic spectrum of this simple unit, without changing its fundamental character. Considering a bias term $\mathrm{b} \neq 0$ also changes the equilibria of the spring. The equilibria of the oscillator are now

all y verifying $\tanh(\mathrm{W}y + \mathrm{b}) - \gamma\, y = 0$. For $\mathrm{W} < 0$, the oscillator becomes multi-stable with up to three equilibria with one unstable and two stable ones.

2.2 Random Oscillators Network

We combine here several of the fundamental units described above into a network of input-driven damped harmonic oscillators that we will later use to perform computations.

Let us start by introducing a pool of N decoupled mechanical oscillators as in eq. (1):

$$\ddot{\mathbf{y}} = \mathbf{f}(t) - \boldsymbol{\gamma} \odot \mathbf{y} - \boldsymbol{\varepsilon} \odot \dot{\mathbf{y}}, \tag{2}$$

where $\odot$ denotes the point-wise multiplication of vectors, $\boldsymbol{\gamma}, \boldsymbol{\varepsilon} \in \mathbb{R}^N$ are the vectors collecting all γ and ε for each of the oscillators. Similarly, $\mathbf{y}, \dot{\mathbf{y}} \in \mathbb{R}^N$ collect all position and velocity for each of the oscillators. Generalising the nonlinear spring $\tanh(\mathrm{W}\,\bar{y} + \mathrm{b})$ discussed above, we introduce here the following more expressive feedback forcing term

$$\mathbf{f}(t) = \tanh(\mathbf{W}\mathbf{y} + \mathbf{V}\mathbf{u}(t) + \mathbf{b}), \tag{3}$$

where $\mathbf{u}(t)$ is the external input driving the network. The forcing term f can be interpreted as in Fig. 2 as an excitation layer (red box) driving the dynamics of the subsequent layer of heterogeneous oscillators (blue box). Being f a function of all configurations $\mathbf{y}$, this forcing term has the effect of nonlinearly connecting the decoupled oscillators into a network. We notice later in section 4 that this forcing term will generate a closed-loop behavior akin to Rusch and Mishra (2023).

In ML terms, eqs. (2)-(3) describe a recurrent layer with hidden state $\mathbf{y} \in \mathbb{R}^N$, with N being the number of neurons. $\mathbf{W} \in \mathbb{R}^{N \times N}$ are the hidden-to-hidden recurrent connections, $\mathbf{V} \in \mathbb{R}^{N \times I}$ are the input-to-hidden connections, and $\mathbf{b} \in \mathbb{R}^N$ the bias vector. The hyperbolic tangent mediates a nonlinear bounded response in the oscillators.

Introducing the variable $\mathbf{z} = \dot{\mathbf{y}}$, we get the following first-order system of ODEs:

$$\dot{\mathbf{y}} = \mathbf{z}, \tag{4}$$

$$\dot{\mathbf{z}} = \tanh(\mathbf{W}\mathbf{y} + \mathbf{V}\mathbf{u}(t) + \mathbf{b}) - \boldsymbol{\gamma} \odot \mathbf{y} - \boldsymbol{\varepsilon} \odot \mathbf{z}, \tag{5}$$

that we discretise with an implicit (the $\dot{\mathbf{y}}$ equation), and an explicit (the $\dot{\mathbf{z}}$ equation) Euler numerical scheme, via discretisation time step $\tau > 0$, as follows.

Definition 2.1. Random Oscillators Network (RON). The RON is a discrete-time RNN model whose update reads as follows:

$$\begin{aligned} \mathbf{y}_{k+1} &= \mathbf{y}_k + \tau \mathbf{z}_{k+1}, \\ \mathbf{z}_{k+1} &= \mathbf{z}_k + \tau\big(\tanh(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b}) \\ &\qquad - \boldsymbol{\gamma} \odot \mathbf{y}_k - \boldsymbol{\varepsilon} \odot \mathbf{z}_k\big). \end{aligned} \tag{6}$$

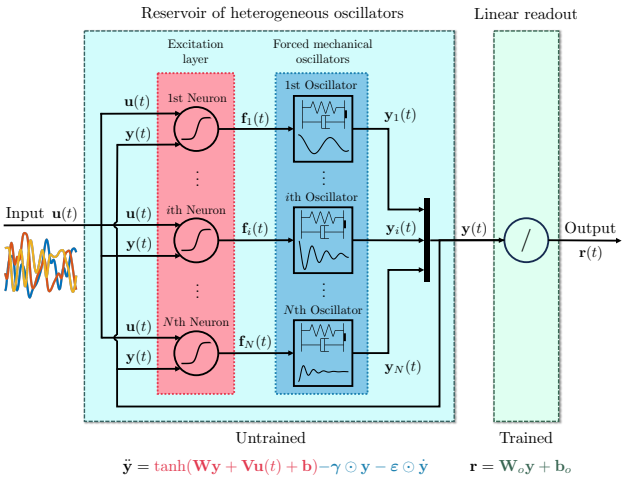


Figure 2: The Random Oscillators Network (RON) consists of N harmonic oscillators forced by coupled neurons with hyperbolic tangent activations. A linear output layer maps the states of the mechanical oscillators in the desired output. This layer is the only one that is adapted during learning.

The hidden states $\mathbf{y}_k$ computed by eqs. (6) are exploited as features encoding crucial temporal information for the processing of the discretised input time series $\mathbf{u}_k$. To solve time series tasks, we stack a linear layer transforming the hidden state $\mathbf{y}$ to an output state $\mathbf{r}$ as follows:

$$\mathbf{r}_{k+1} = \mathbf{W}_o \mathbf{y}_{k+1} + \mathbf{b}_o \tag{7}$$

where $\mathbf{W}_o, \mathbf{b}_o$, are weights and biases of the output layer. Eqs. (6)-(7) describe an RNN model mapping input sequences $\mathbf{u}_k$ into output sequences $\mathbf{r}_k$, see Fig. 2 for a schematic representation of RON model.

In RON, the parameters $\mathbf{W}, \mathbf{V}, \mathbf{b}$, of the feedback forcing term are *randomly generated and kept fixed*. Precisely, they are generated according to a uniform distribution in $(-2, 2)$ for $\mathbf{W}$, $(0, 1)$ for $\mathbf{V}$ and $(-1, 1)$ for $\mathbf{b}$. Then, in line with the RC framework, the matrix $\mathbf{W}$ is scaled with an hyperparameter $\rho > 0$ which tunes its spectral radius, and the matrix $\mathbf{V}$ is scaled with an hyperparameter $\nu > 0$ (see Lukoševičius (2012) for a practical guide to RC techniques). Similarly to ρ, and ν, also the stiffness and damping coefficients of the oscillators, respectively $\boldsymbol{\gamma}$, and $\boldsymbol{\varepsilon}$, are treated as hyperparameters of the RON to be selected via validation techniques. In practice, we create heterogeneous oscillators selecting a midpoint and a radius for each $\boldsymbol{\gamma}$ and $\boldsymbol{\varepsilon}$, thus generating uniformly random values within that interval. An important feature of RON is the heterogeneity of input-driven responses of its reservoir of oscillators. This heterogeneity allows the linear readout to extract more easily the crucial features for solving the task at hand. Remarkably, entries of $\mathbf{W}_o, \mathbf{b}_o$, are the only trainable parameters of an RON, and we use

ridge regression for learning them. The scalar value τ in eqs. (6) is linked to the step size of the numerical integration. Therefore, if one wants to discretise the continuous-time model for the sake of merely reproducing the continuous-time dynamics, then an opportunely small value of the step size is required. However, here we are not interested in reliably simulating trajectories of the continuous-time dynamical system defined by eqs. (4)-(5), but rather to investigate the expressiveness of the physically-inspired discrete-time RNN model of eqs. (6) that we derived. As a consequence, in the remainder of this paper, we will treat $\tau > 0$ as a hyperparameter.

In order to evaluate the impact of randomisation, we consider a version of RON, that we call *heterogeneously coupled oscillatory RNN (hcoRNN)*, where parameters $\mathbf{W}, \mathbf{V}, \mathbf{b}$, of the feedback forcing term are learned via the backpropagation through-time algorithm. Whilst we consider $\tau, \boldsymbol{\varepsilon}, \boldsymbol{\gamma}$, as hyperparameters of the hcoRNN model.

2.3 RON as Leaky-ESN

In the particular case of eqs. (6) with $\boldsymbol{\varepsilon} \equiv \frac{1}{\tau}$, the $\mathbf{z}$-dynamics become completely determined by the $\mathbf{y}$-dynamics. Therefore, the hcoRNN equation becomes

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \tau^2 \tanh(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b}) - \tau^2 \boldsymbol{\gamma} \odot \mathbf{y}_k.$$

Interestingly, setting further $\boldsymbol{\gamma} \equiv 1$, we recover a popular RC model named Leaky-ESN (Jaeger et al., 2007), whose state-update equation reads as follows:

$$\mathbf{y}_{k+1} = \tau^2 \tanh(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b}) + (1 - \tau^2)\mathbf{y}_k. \quad (8)$$

In the context of eq. (8), the hyperparameter τ is interpreted as the squared root of the leak rate of the model.

The Leaky-ESN model has been successfully used in many ML tasks involving time series, like audio processing (Jaeger et al., 2007). Remarkably, the Leaky-ESN with linear output layer as in eq. (7) can accurately learn the climate of chaotic attractors (Lu et al., 2018).

From this perspective, the RON can be interpreted as a generalisation of the Leaky-ESN model, and as such it has the capability to describe both stable complex oscillatory dynamics and chaotic dynamics, provided with an opportune choice of hyperparameters.

3 LINEAR STABILITY ANALYSIS

In this section, we perform a linear stability analysis of the RON model (Definition 2.1). All the proofs can be found in the Supplementary Material. Let us denote $\mathbf{X}_k = \begin{pmatrix} \mathbf{y}_k \\ \mathbf{z}_k \end{pmatrix}$. Then, the RON model can be defined by the input-driven state-update equation $\mathbf{X}_{k+1} = G(\mathbf{X}_k, \mathbf{u}_{k+1})$, where $G : \mathbb{R}^{2N} \times \mathbb{R}^I \to \mathbb{R}^{2N}$ is defined by eqs. (6). The Jacobian of the G map computed on $(\mathbf{X}_k, \mathbf{u}_{k+1})$, denoted with $\mathbf{J}_k$, reads:

$$\mathbf{J}_k = \begin{bmatrix} \frac{\partial \mathbf{y}_{k+1}}{\partial \mathbf{y}_k} & \frac{\partial \mathbf{y}_{k+1}}{\partial \mathbf{z}_k} \\ \frac{\partial \mathbf{z}_{k+1}}{\partial \mathbf{y}_k} & \frac{\partial \mathbf{z}_{k+1}}{\partial \mathbf{z}_k} \end{bmatrix} = \quad (9)$$

$$= \begin{bmatrix} \mathbf{I} + \tau^2 \mathbf{A}_k & \tau\big(\mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon})\big) \\ \tau \mathbf{A}_k & \mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon}) \end{bmatrix}, \quad (10)$$

where

$$\mathbf{A}_k = \mathbf{S}_k \mathbf{W} - \mathrm{diag}(\boldsymbol{\gamma}), \quad (11)$$

$$\mathbf{S}_k = \mathrm{diag}\big(1 - \tanh^2(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b})\big). \quad (12)$$

A widely known stability condition for RC systems (Jaeger, 2001) is given by imposing that the Jacobian is a contraction. This condition of contraction implies the existence and uniqueness of a uniformly asymptotically stable solution for the input-driven system, and it is expressed by imposing the Euclidean norm of the Jacobian to be uniformly less than 1 (Ceni et al., 2020). From eq. (9) we can already see that $\mathbf{J}_k = \mathbf{I} + \tau \begin{bmatrix} \tau \mathbf{A}_k & \mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon}) \\ \mathbf{A}_k & -\mathrm{diag}(\boldsymbol{\varepsilon}) \end{bmatrix}$ has a bias towards the identity mapping for small values of τ. We provide below an upper bound for the Euclidean norm of the Jacobian of eq. (9) based on the following quantities

$$\xi = \max_j |1 - \tau \varepsilon_j|, \quad (13)$$

$$\eta = \max_j |1 - \tau^2 \gamma_j|, \quad (14)$$

$$\sigma = ||\mathbf{W}||, \quad (15)$$

while we denote $\gamma_{\max} = \max_j \gamma_j$, $\varepsilon_{\max} = \max_j \varepsilon_j$, and similarly for the min values we use $\gamma_{\min}, \varepsilon_{\min}$.

Theorem 3.1. *The norm of the Jacobian matrix of the RON model admits the following upper bound*

$$||\mathbf{J}_k|| \leq \max(\eta + \tau^2 \sigma, \xi) + \tau \max(\xi, \gamma_{max} + \sigma). \quad (16)$$

In particular, for $\tau \ll 1$, and assuming $\varepsilon_{min} > 0$, and $\gamma_{max} \geq 1$, the bound reads as follows:

$$||\mathbf{J}_k|| \leq 1 + \tau(\gamma_{max} + \sigma) + O(\tau^2). \quad (17)$$

Theorem 3.1 highlights that although the entire eigenspectrum can be uniformly bounded around a neighbourhood of the identity by means of τ, it is a hard task to find combinations of hyperparameters ensuring that $\sup_k ||\mathbf{J}_k|| < 1$, and so ensuring uniform asymptotic stability for the RON model. One interesting example is given by the particular case of $\boldsymbol{\varepsilon} \equiv \frac{1}{\tau}$, where the

y-dynamics becomes decoupled from the **z**-dynamics, as observed in the previous section; we provide sufficient conditions for contractivity for such a particular case in the Supplementary Material. In the general case, imposing the upper bound of eq. (16) to be less than 1, we obtain *sufficient conditions for a contractive RON*, thus a uniformly asymptotically stable RON in particular. For sake of conciseness, these sufficient conditions for a contractive RON are reported in the Supplementary Material.

These sufficient conditions define a very narrow region of hyperparameters. The difficulty to satisfy these sufficient conditions reflects how disinclined is RON to this strong condition of stability. Imposing such strict conditions of contractivity on the RON model might harm its expressiveness. We might relax the request of contracting at each time step in favour of the less stringent requirement of having all the eigenvalues inside the unit circle. Note however that, for a generic discrete-time linear non-autonomous system, having all eigenvalues inside the unit circle is not sufficient to imply asymptotic stability (Slotine et al., 1991; Kozachkov et al., 2022).

We provide a more precise picture of the eigenvalues distribution of the RON model in the following theorem.

Theorem 3.2. *For all μ eigenvalues of the Jacobian of the RON model there exists a point $\lambda \in \{ 1 - \tau^2 \gamma_j , 1 - \tau \varepsilon_j \}_{j=1}^{N}$ such that*

$$|\mu - \lambda| \leq C, \tag{18}$$

where $C = \tau^2\sigma + \tau \max\big(\xi, \gamma_{max} + \sigma\big)$.

According to Theorem 3.2, the eigenspectrum of the Jacobian of an RON model is contained inside the union of disks of radius C centered on the points $1 - \tau\varepsilon_i, 1 - \tau^2\gamma_i$, see Fig. 3 for a visual representation of this fact, and Fig. 4 for few eigenspectrum configurations of an RON. For an input-free RON, having all eigenvalues inside the unit circle does suffice to have asymptotic stability. We provide a necessary condition based on the strength of the feedback loop, i.e. providing a bound on the σ hyperparameter, for the asymptotic stability of an input-free RON.

Proposition 3.3. ***Necessary conditions.*** *If the input-free RON model is asymptotically stable, then either one of the two cases must hold true*

- *if $\sigma > \xi - \gamma_{max}$, then $\sigma \leq \dfrac{1 - \tau\gamma_{max}}{\tau + \tau^2}$.*
- *if $\sigma \leq \xi - \gamma_{max}$, then $\sigma \leq \dfrac{1 - \tau\xi}{\tau^2}$.*

The idea of the proof relies on imposing that all the disks of Theorem 3.2 lie inside the unit circle. This

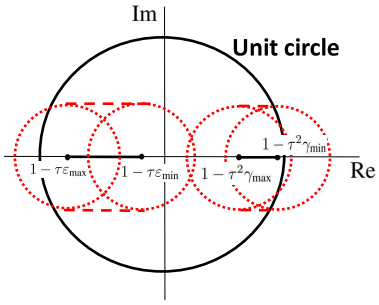


Figure 3: Depiction of the eigenspectrum's bound given by Theorem 3.2 for the Jacobian of an RON model.

translates in the conditions:

$$C \leq \tau\varepsilon_i \leq 2 - C, \tag{19}$$
$$C \leq \tau^2\gamma_i \leq 2 - C, \tag{20}$$

for all $i = 1, \ldots, N$, where C is defined in Theorem 3.2. The necessary conditions of Proposition 3.3 are derived by imposing that $C \leq 1$, which is a necessary condition for inequalities (19)-(20) to hold, otherwise $2 - C < C$.

Remark 3.4 (**Guideline values**). Searching around the edge of stability of RON, we can lose inequalities (19)-(20), pushing C to zero, to derive the following guideline values for the hyperparameters of the oscillators:

$$\varepsilon_{\min} \geq 0, \tag{21}$$
$$\gamma_{\min} \geq 0, \tag{22}$$
$$\varepsilon_{\max} \leq 2/\tau, \tag{23}$$
$$\gamma_{\max} \leq 2/\tau^2. \tag{24}$$

Selecting τ values small enough while satisfying eqs. (21)-(24) will generate typically RON models with an underlying Jacobian just marginally unstable with eigenvalues at most slightly beyond the unitary circle in a neighborhood of the value of 1. Therefore, promoting the computation at the edge of stability, which has been recognised to be useful for time series processing (Legenstein and Maass, 2007), and lately also in typical deep learning applications (Cohen et al., 2021).

4 RELATED WORKS

Reservoir Computing. The key idea behind RC is to treat the internal dynamics of a recurrent neural network as fixed, i.e. untrained (Lukoševičius and Jaeger, 2009). RC leverages smart random initialisations of the recurrent part of the model, called the *reservoir*, and only trains an output layer, usually with linear regression techniques, to decode the internal dynamics

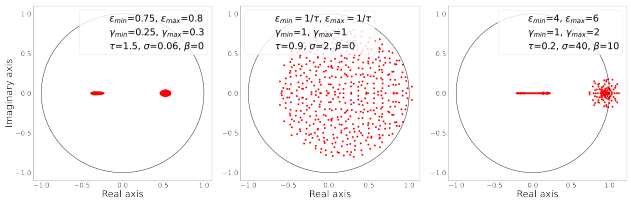


Figure 4: Jacobian's eigenspectrum eq. (9) for few hyperparameter's combinations. Bias vector has β in all its components, input-to-hidden matrix $\mathbf{V}$ is zero matrix. **Left:** a case satisfying the sufficient conditions for a contractive RON. **Centre:** a case of RON coinciding with Leaky ESN. **Right:** a case satisfying the guideline values of Remark 3.4 with strong coupling, i.e. $\sigma \gg 1$.

into an output signal.
The Echo-State Network (ESN) (Jaeger, 2002) is one of the most representative models belonging to the RC paradigm. In this work, we use ESNs to assess the effectiveness of our model against state-of-the-art RC models. The state-update equation of an ESN reads:

$$\mathbf{y}_k = \alpha \tanh(\mathbf{W}\mathbf{y}_{k-1} + \mathbf{V}\mathbf{u}_k + \mathbf{b}) + (1-\alpha)\mathbf{y}_k, \quad (25)$$

where $\mathbf{y}_k \in \mathbb{R}^N, \mathbf{u}_k \in \mathbb{R}^I$ are the internal state and input trajectories at time-step k, respectively and $\mathbf{W} \in \mathbb{R}^{N\times N}, \mathbf{V} \in \mathbb{R}^{N\times I}, \mathbf{b} \in \mathbb{R}^N$ are fixed, randomly-initialised parameters. The hidden matrix $\mathbf{W}$ is randomly initialised and then scaled such that its spectral radius ρ is smaller than 1. Also, the matrix $\mathbf{V}$ is scaled by a scalar value ν. The scalar α is the leak rate, it is entitled to calibrate the time scale of the ESN's dynamics according to that of the input signal. All the three ρ, ν, and α, are task-specific and are usually determined by a model selection phase (e.g., with random/grid search). We adopt this approach for all ESNs used in our experiments.

coRNN. We compare the performance of RON against the coupled oscillatory RNN (coRNN) from Rusch and Mishra (2023). coRNN also builds on a network of oscillators. A coRNN has unique scalar values $\boldsymbol{\gamma} \equiv \gamma, \boldsymbol{\varepsilon} \equiv \varepsilon$, and it is defined by the following equation:

$$\begin{aligned}\mathbf{y}_{k+1} &= \mathbf{y}_k + \tau \mathbf{z}_{k+1},\\ \mathbf{z}_{k+1} &= \mathbf{z}_k + \tau\big(\tanh(\mathbf{W}\mathbf{y}_k + \mathcal{W}\mathbf{z}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b}) \\ &\qquad - \gamma\mathbf{y}_k - \varepsilon\mathbf{z}_k\big).\end{aligned} \quad (26)$$

Differently from RON, the coRNN model is fully trained, does not use heterogeneous oscillators and requires an additional hidden-to-hidden adaptive matrix $\mathcal{W}$. In particular, for the same number of units, the coRNN model has more trainable parameters than RON (and hcoRNN), thus larger computational time for training and inference.

5 EXPERIMENTS

Our empirical evaluation focuses on two key RON properties, discussed in the theoretical analysis of Section 2[1]:

1. We study the impact of weight *randomisation* by comparing the performance of an RON against fully-trained hcoRNN, coRNN, and LSTM (Hochreiter and Schmidhuber, 1997), on both sequence classification and time series forecasting benchmarks. We also highlight the advantages of RON in terms of computational efficiency.

2. We study the role played by the dynamical system *stability* in an RON. To this end, we show the stability properties of the best models, according to our findings of Section 3.

To guarantee a fair comparison, we adopt the experimental setup of Rusch and Mishra (2023), where the coRNN model was first introduced, and we extend it with additional benchmarks. For each benchmark and model, we performed grid search on a separate validation set to obtain the best models which were then evaluated on the held-out test set. We report all the details related to model selection and best configurations in the Supplementary Material.

Sequence classification benchmarks. We use 6 classification benchmarks, sMNIST, psMNIST, npCIFAR-10, FordA, Adiac and uWaveGesture[2]. The MNIST and CIFAR-based benchmarks are used to test the long-term memory capabilities of the model, while FordA and Adiac for medium and short-term memory, respectively.
In sMNIST, the model observes one pixel at a time and it is required to predict the digit class after having processed all the 784 pixels. The psMNIST benchmark is the same as sMNIST, except that the pixels in an image are shuffled according to a fixed, random permutation.
The npCIFAR-10 benchmark presents each RGB image of CIFAR-10 in a row-wise fashion (flattening the RGB channels into a single vector), leading to sequences of 32 elements. A randomly generated suffix is added to each sequence, reaching a sequence length of 1,000 time steps. Since the information required to classify the image is contained at the very beginning of the sequence, the model needs to extend its memory over hundreds of steps. FordA is a binary classification task

[1]The code to reproduce the experiments is available at `https://github.com/AndreaCossu/RandomizedCoupledOscillators/`

[2]FordA, Adiac and uWaveGesture from `timeseriesclassification.com`

Table 1: Test Accuracy (for sMNIST, psMNIST, npCIFAR-10, FordA, Adiac), and test NRMSE (for Lorenz96, Mackey-Glass). Mean values computed over 5 trials, empirical standard deviation in subscript. Bold for best accuracy for fully-trained (red) and randomised models (green), underline for overall best accuracy.

Model	Fully-trained			Randomised	
	LSTM	coRNN	hcoRNN (our)	Leaky ESN	RON (our)
sMNIST ↑	$0.9860_{\ 0.0017}$	$\underline{\mathbf{0.9921}_{\ \mathbf{0.0002}}}$	$0.9871_{\ 0.0011}$	$0.9211_{\ 0.0020}$	$\mathbf{0.9780}_{\ \mathbf{0.0006}}$
psMNIST ↑	$0.8761_{\ 0.0390}$	$0.9435_{\ 0.0224}$	$\underline{\mathbf{0.9635}_{\ \mathbf{0.0048}}}$	$0.8503_{\ 0.0150}$	$\mathbf{0.9301}_{\ \mathbf{0.0054}}$
npCIFAR-10 ↑	$0.1000_{\ 0.0000}$	$\underline{\mathbf{0.5841}_{\ \mathbf{0.0033}}}$	$0.5548_{\ 0.0031}$	$0.2060_{\ 0.0016}$	$\mathbf{0.4158}_{\ \mathbf{0.0101}}$
FordA ↑	$0.5803_{\ 0.0432}$	$0.7003_{\ 0.1535}$	$\underline{\mathbf{0.7944}_{\ \mathbf{0.0859}}}$	$0.5461_{\ 0.0320}$	$\mathbf{0.6885}_{\ \mathbf{0.0385}}$
Adiac ↑	$0.4793_{\ 0.0187}$	$0.4517_{\ 0.0252}$	$\underline{\mathbf{0.5586}_{\ \mathbf{0.0706}}}$	$0.6928_{\ 0.0116}$	$\underline{\mathbf{0.7313}_{\ \mathbf{0.0050}}}$
uWaveGesture ↑	0.59 ± 0.05	0.85 ± 0.01	$\mathbf{0.87}_{\ \mathbf{0.018}}$	0.86 ± 0.01	$\mathbf{0.89}_{\ \mathbf{0.01}}$
Lorenz96 ↓	$2.4 \times 10^{-1}_{3.6\times10^{-2}}$	$\mathbf{2.1 \times 10^{-1}_{5.2\times10^{-2}}}$	$2.6 \times 10^{-1}_{2.5\times10^{-2}}$	$2.0 \times 10^{-3}_{2.0\times10^{-4}}$	$\underline{\mathbf{1.6 \times 10^{-3}_{1.7\times10^{-4}}}}$
Mackey-Glass ↓	$\mathbf{3.4 \times 10^{-2}_{3.2\times10^{-3}}}$	$6.2 \times 10^{-2}_{1.5\times10^{-2}}$	$5.4 \times 10^{-2}_{4.9\times10^{-3}}$	$3.0 \times 10^{-2}_{1.4\times10^{-3}}$	$\underline{\mathbf{1.8 \times 10^{-2}_{6.5\times10^{-3}}}}$

from real-world data (of time series of length 500) to diagnose whether a certain symptom exists or does not exist in an automotive subsystem.
Adiac is used to measure the ability of the model to classify among a large number of classes (37) on relatively short sequences (176 time steps).
uWaveGestures is a benchmark that represents different gestures measured from accelerometers.

Chaotic systems forecasting. In Rusch and Mishra (2023), the authors discussed how coRNN models are not tailored to time series forecasting for chaotic systems, due to their inability (by design) to generate chaotic dynamics. We show that, instead, our RON model is very effective in predicting chaotic systems. Following Rusch and Mishra (2023), we ran time-series forecasting experiments on the Lorenz96 chaotic system. The Lorenz96 system is defined by the following differential equation $\dot{x_i} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + F$, with $i = 1, \dots, 5$ and F an external driving force. The task consists in predicting the next 25-th state of the system in the chaotic regime with $F = 8$. The training, validation, and test sets are composed of 128 trajectories of length 2000. Each trajectory is independently generated by solving the Lorenz96 equation with a random initial condition sampled uniformly from $[F - 0.5, F + 0.5]$ and a discretisation time-step of 0.01. In addition, we also study the popular Mackey-Glass chaotic system (Mackey and Glass, 1997). Similar to Jaeger and Haas (2004), the task is to predict the 84-th next state of the system. As commonly done for time series forecasting, we used an initial washout of length 200 (the first 200 steps are used to warm up the model, but are not used when evaluating its performance). The performance of the models is measured by the Normalised Root Mean Squared Error (NRMSE), where normalisation is performed by diving the RMSE by the root mean square of the target trajectory, as in Rusch and Mishra (2023).

6 RESULTS

Table 1 reports the results on all benchmarks for LSTM, Leaky ESN, coRNN, hcoRNN and RON. Results are divided in two categories: fully-trained models, and randomised ones. We highlight in bold the best accuracy within each category, in red for fully-trained and green for randomised models, underlining the overall better. Within each benchmark, we use the same number of trainable parameters for each model. For CIFAR, Lorenz96, and MNIST-based benchmarks we keep a large number of trainable parameters similarly to Rusch and Mishra (2023), while for FordA, Adiac, uWaveGesture and Mackey-Glass we explore a smaller parametrisation regime (Table 2).

On the long-term memory benchmarks, sMNIST, psMNIST and npCIFAR-10, our RON model outperforms the Leaky ESN by a large margin. It is important to stress that, usually, fully-trained models achieve higher performance than randomised RC models in long-term memory benchmarks. Instead, except for the very challenging npCIFAR-10, RON not only outperforms ESN models but also significantly narrows the gap with respect to fully-trained models.

FordA appears a challenging task to solve for LSTM and Leaky ESN models, arguably due to the low parametrisation regime. Here oscillators-based models exhibit a clear advantage, in particular our hcoRNN achieves the best overall accuracy. On the Adiac and uWaveGesture benchmarks, randomised RC models show an advantage over fully-trained models, and in particular RON achieves the best overall accuracy.

As shown by the Lorenz96 and Mackey-Glass results, RON obtains the best overall performance when modelling chaotic dynamical systems. RON sharply outper-

Table 2: Number of trainable parameters and total training time (in minutes) for each benchmark and model. The number of parameters includes linear classifier/predictor. hcoRNN is comparable in both size and training time to a coRNN from Rusch and Mishra (2023).

Model Parameters	sMNIST $\approx 134k$	psMNIST $\approx 134k$	npCIFAR-10 $\approx 52k$	FordA $\approx 0.1k$	Adiac $\approx 3.7k$	Lorenz96 $\approx 34k$	Mackey-Glass $\approx 1k$
hcoRNN (our)	230m	230m	360m	16m	2m	8m	1m
Leaky ESN	11m	12m	5m	0.05m	0.03m	1m	0.5m
RON (our)	11m	12m	5m	0.05m	0.03m	1m	0.5m

Table 3: Validity of guideline values of Remark 3.4 and necessary conditions of Prop. 3.3 for best RON configurations from Table 1 for all tasks. The value of the best hyper-parameters can be found in the Supplementary Material.

$\varepsilon_{\text{min}} \geq 0$	$\gamma_{\text{min}} \geq 0$	$\tau\varepsilon_{\text{max}} \leq 2$	$\tau^2\gamma_{\text{max}} \leq 2$	Prop. 3.3
✓	✓	✓	✓	✓

forms fully-trained models by two orders of magnitudes in the Lorenz96 task, while surpassing state-of-the-art RC models. Aligned with what was hypothesised by Rusch and Mishra (2023), fully-trained models like coRNN struggle in predicting chaotic behaviours. However, the same family of models shows excellent performance in the same task when equipped with randomisation properties.
Generally in all the considered tasks, RON always outperforms Leaky ESN, demonstrating that a pool of randomly connected heterogeneous oscillators can provide a set of internal representations for time series processing purposes richer than typical ESN models. Overall, either hcoRNN or RON are the best-performing models in 6 out of 7 benchmarks, highlighting the effectiveness of their architectural bias.

RON stability. We verified whether or not the best RON configurations (provided in the Supplementary Material) satisfy the *guideline values* of Remark 3.4 and the necessary conditions of Prop. 3.3. We can observe from Table 3 that the best RON in our experiments always satisfy both the guideline and the necessary conditions. We also compared the performance of our best RON against an RON that satisfies the *sufficient* conditions ensuring a contracting map. The results of this comparison, which can be found in the Supplementary Material, show that the *sufficient* conditions are overly restrictive and do not allow to learn properly any of the time series tasks. Therefore, RON requires to go beyond contractivity and towards edge of stability configurations, where it performed best according to all our experiments.

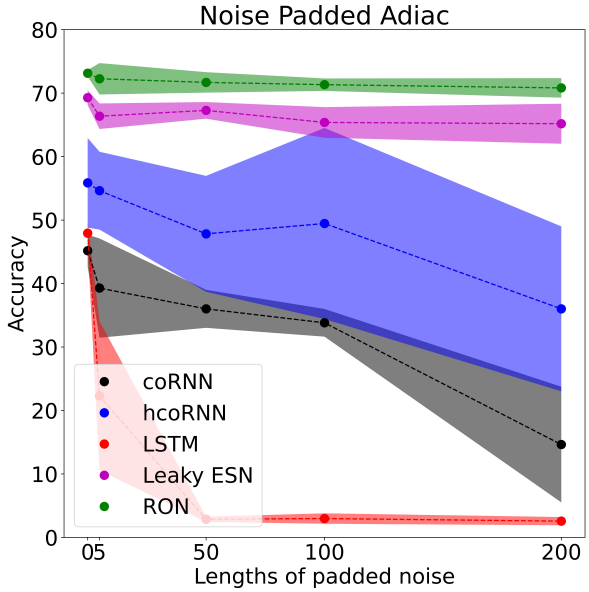


Figure 5: Accuracy on the npAdiac dataset. The x-axis shows the padded sequence length.

Study on varying sequence length. We considered the noise-padded Adiac task (npAdiac) to test the performance of all models to varying time-series lengths. The npAdiac dataset is created by appending after each Adiac time series (composed by 176 time steps) a number of 5, 50, 100, and 200 time steps of Gaussian noise (mean 0 and std 1). We report the accuracy results in Figure 5. The dots represent the average accuracy over 5 trials, the coloured shades cover one standard deviation. We used the same hyperparameters used for the Adiac task and reported in Table 5.
LSTM appears extremely sensitive to the padding of noise. In general, randomised models appear more resilient than fully-trained ones. In particular, hcoRNN keeps a better performance than coRNN and LSTM, while RON is always the overall best performing.

Comparison with expRNN. We compare RON against the expRNN model (Lezcano-Casado and Martínez-Rubio, 2019), a fully-trainable RNN with

Table 4: Accuracy/loss for the expRNN model and RON on Adiac, FordA and Mackey-Glass datasets.

	Adiac ↑	FordA ↑	MG (10^{-2}) ↓
expRNN	$0.62_{0.05}$	$0.59_{0.06}$	$6.46_{1.71}$
RON	$\mathbf{0.73_{0.01}}$	$\mathbf{0.69_{0.04}}$	$\mathbf{1.8_{0.65}}$

orthogonal recurrent weights. Results are shown in Table 4. RON always outperforms expRNN. On Adiac, the expRNN achieves the best performance among fully-trained models. On FordA, the expRNN surpasses the LSTM, but it underperforms with respect to oscillatory-based models. ExpRNN does not seem suitable for chaotic time series forecasting, achieving the worst performance among the considered models.

Computational efficiency. Due to the untrained recurrent layer, the training time can be up to two orders of magnitudes smaller in RON than in fully-trained models (Table 2). In fact, RON does not need to be trained with back-propagation through time, greatly improving its computational efficiency both in terms of time and energy consumption, compared to (h)coRNN and LSTM.
The oscillatory-based recurrent models require more hyperparameters tuning than other recurrent models like LSTM. This often makes it challenging to train models like hcoRNN and coRNN. However, the lower training time for RON allows us to explore a wider range of configurations than in fully-trained models. This appears to be crucial, since the (h)coRNN is quite sensitive to the choice of its hyper-parameters, like τ. Training times for coRNN and LSTM are comparable to those reported for hcoRNN in Table 2.

7 CONCLUSION AND FUTURE WORKS

We developed a theoretical and empirical analysis of recurrent dynamical systems based on randomly coupled oscillators. We introduced the RON model, which builds on a dynamical recurrent layer made up of untrained heterogeneous oscillators with randomisation properties. Our theoretical analysis shows how RON can be effectively driven towards edge of stability configurations. We empirically evaluated RON on a set of sequence classification and time series forecasting benchmarks. Our results show that RON exhibits an effective long-term memory while greatly improving the performance in chaotic systems prediction. The computational efficiency of RON in terms of training time and the number of parameters is orders of magnitude better than fully-trainable recurrent networks, including those based on coupled oscillatory systems.

Looking ahead at future works, we plan to investigate deep architectural organisations of the randomised oscillators in RON. Following the principles of Deep RC (Gallicchio et al., 2017), Deep RON would be able to learn richer latent representations at multiple temporal scales. Due to its physically-inspired design, RON is amenable to implementation in neuromorphic hardware (like mechanical or optical systems) (Rajendran et al., 2019) that can mimic the behaviour of coupled oscillators. Due to its efficient design and restricted training protocol, RON can be deployed in resource-constrained environments, like at the edge, on devices with limited computational capacity. This can lead towards a more sustainable and energy-efficient way of leveraging deep learning models for time series processing.

Acknowledgements

This work has been supported by EU-EIC EMERGE (Grant No. 101070918), EU NextGenerationEU programme under the funding schemes PNRR-PE-AI (PE00000013) FAIR - Future Artificial Intelligence Research, and NEURONE, a project funded by the Italian Ministry of University and Research (PRIN 20229JRTZA).

References

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

Bertschinger, N. and Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural computation*, 16(7):1413–1436.

Ceni, A., Ashwin, P., Livi, L., and Postlethwaite, C. (2020). The echo index and multistability in input-driven recurrent neural networks. *Physica D: Nonlinear Phenomena*, 412:132609.

Cohen, J. M., Kaur, S., Li, Y., Kolter, J. Z., and Talwalkar, A. (2021). Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211.

Gallicchio, C., Micheli, A., and Pedrelli, L. (2017). Deep reservoir computing: A critical experimental analysis. *Neurocomputing*, 268:87–99.

Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., and Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biological cybernetics*, 105:355–370.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13.

Jaeger, H. (2002). Adaptive nonlinear system identification with echo state networks. *Advances in neural information processing systems*, 15.

Jaeger, H. and Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80.

Jaeger, H., Lukoševičius, M., Popovici, D., and Siewert, U. (2007). Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3):335–352.

Kozachkov, L., Ennis, M., and Slotine, J.-J. (2022). Rnns of rnns: Recursive construction of stable assemblies of recurrent neural networks. *Advances in Neural Information Processing Systems*, 35:30512–30527.

Lee, R. H., Mulder, E. A., and Hopkins, J. B. (2022). Mechanical neural networks: Architected materials that learn behaviors. *Science Robotics*, 7(71):eabq7278.

Legenstein, R. and Maass, W. (2007). Edge of chaos and prediction of computational performance for neural circuit models. *Neural networks*, 20(3):323–334.

Lezcano-Casado, M. and Martínez-Rubio, D. (2019). Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3794–3803. PMLR.

Lu, Z., Hunt, B. R., and Ott, E. (2018). Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104.

Lukoševičius, M. (2012). A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 659–686. Springer.

Lukoševičius, M. and Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer science review*, 3(3):127–149.

Mackey, M. C. and Glass, L. (1997). Oscillation and Chaos in Physiological Control Systems. *Science*, 197(4300):287–289.

Nakajima, K. and Fischer, I. (2021). *Reservoir Computing*. Springer.

Nakajima, K., Hauser, H., Li, T., and Pfeifer, R. (2015). Information processing via physical soft body. *Scientific reports*, 5(1):10487.

Pikovsky, A., Rosenblum, M., and Kurths, J. (2002). Synchronization: a universal concept in nonlinear science.

Rajendran, B., Sebastian, A., Schmuker, M., Srinivasa, N., and Eleftheriou, E. (2019). Low-Power Neuromorphic Hardware for Signal Processing Applications: A Review of Architectural and System-Level Design Approaches. *IEEE Signal Processing Magazine*, 36(6):97–110.

Rusch, T. K. and Mishra, S. (2023). Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*.

Slotine, J.-J. E., Li, W., et al. (1991). *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ.

Wright, L. G., Onodera, T., Stein, M. M., Wang, T., Schachter, D. T., Hu, Z., and McMahon, P. L. (2022). Deep physical neural networks trained with backpropagation. *Nature*, 601(7894):549–555.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes
   (b) Complete proofs of all theoretical results. Yes
   (c) Clear explanations of any assumptions. Yes

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. Yes

(b) The license information of the assets, if applicable. Yes

(c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable

(d) Information about consent from data providers/curators. Not Applicable

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. Not Applicable

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

A Model selection

We provide the complete experimental setup used for model selection. Table 5 report the grid search performed during model selection on all benchmarks, with the best value in bold.
We used the same number of adaptive parameters for each model. This means that randomised models use more hidden units than the fully-trained ones, since they only have trainable hidden-to-output parameters. To get the total number of adaptive parameters for RON and ESNs, one simply needs to multiply the number of hidden units and the output size (number of readout units).
RON and ESNs use 13,000 units for sMNIST and psMNIST, 5,200 units for npCIFAR-10, 6,800 units for Lorenz96, 1,000 units for Mackey-Glass, and 100 units for Adiac and FordA. Fully-trained models use 256 units for sMNIST and psMNIST, 128 units for npCIFAR-10, 130 units for Lorenz96, and 22 units for Mackey-Glass. hcoRNN, coRNN, LSTM, use respectively 8,6,4, units for FordA, and 44,34,25, units for Adiac, this to keep the same number of trainable parameters. Fully-trained models have been trained for the same number of epochs (120) as the original coRNN model from Rusch and Mishra (2023). Due to the high computational cost of both coRNN and hcoRNN, the grid search for hcoRNN was performed around the best hyper-parameters found by the coRNN original paper (Rusch and Mishra, 2023).
For Leaky ESN and RON, the recurrent weight matrix was uniformly initialised in $[-2, 2]$, before scaling the spectral radius. The input-to-state matrix was uniformly initialised in $[0, 1]$.

For the Stable-RON model, we used the following configuration: $\tau = 1.1, \gamma = 0.58 \pm 0.03, \epsilon = 0.77 \pm 0.10, \rho = 0.01$. These hyper-parameters satisfy the sufficient conditions of Proposition 3.2, but are overly restrictive for the model.

B Proof of Theorem 3.1

We will make use of the following lemma.

Lemma B.1. *Let be given two square matrices* $\mathbf{M}, \mathbf{N}$ *of the same dimension. Then it holds that*

$$\textit{(i)} \quad \left|\left|\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix}\right|\right| \leq \max(||\mathbf{M}||, ||\mathbf{N}||),$$

$$\textit{(ii)} \quad \left|\left|\begin{bmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{N} & \mathbf{0} \end{bmatrix}\right|\right| \leq \max(||\mathbf{M}||, ||\mathbf{N}||).$$

Proof. We notice that for any unitary vector $\mathbf{X} = \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}$ it holds

$$\left|\left|\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}\right|\right|^2 = \|\mathbf{My}\|^2 + \|\mathbf{Nz}\|^2 \leq$$

$$\leq \max(\|\mathbf{My}\|, \|\mathbf{Nz}\|)^2,$$

from which it follows that $\left|\left|\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix}\right|\right| \leq \max(||\mathbf{M}||, ||\mathbf{N}||)$. For the antidiagonal case, note that $\begin{bmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{N} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}$, hence we have for any unitary vector $\mathbf{X} = \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}$ that

$$\left|\left|\begin{bmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{N} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}\right|\right|^2 = \left|\left|\begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix}\right|\right|^2 \leq$$

Table 5: Model selection configurations. α is the ESN leaky rate, ν is the input scaling, ρ is the spectral radius. For Adiac and FordA, the coRNN model has been validated on the same grid of hcoRNN. The selected configuration for coRNN on FordA was $\tau = 0.25, \epsilon = 3, \gamma = 1$, and on Adiac was $\tau = 0.1, \epsilon = 1, \gamma = 5$.

sMNIST	Configuration
hcoRNN	$\epsilon = 4.7 \pm \{1, \mathbf{0.5}\}, \gamma = 2.7 \pm \{1, \mathbf{0.5}\}, \tau = \{0.42, \mathbf{0.042}\}$
Leaky ESN	$\alpha = \{1, 0.5, 0.1, 0.01, \mathbf{0.001}\}, \rho = \{900, 90, 9, \mathbf{0.999}, 0.99, 0.9\}, \nu = \{10, \mathbf{1}, 0.1\}$
RON	$\tau = \{0.42, \mathbf{0.042}\}, \rho = \{900, 90, \mathbf{9}, 0.9\}, \nu = \{10, \mathbf{1}, 0.1\}, \epsilon = \{5.1, \mathbf{0.51}\} \pm \{1, \mathbf{0.5}\}, \gamma = \{\mathbf{2.7}, 0.27\} \pm \{\mathbf{1}, 0.5\}$
psMNIST	
hcoRNN	$\epsilon = 8.0 \pm \{1, \mathbf{0.5}\}, \gamma = 0.4 \pm \{\mathbf{1}, 0.5\}, \tau = \{0.76, \mathbf{0.076}\}$
Leaky ESN	$\alpha = \{1, 0.5, 0.1, \mathbf{0.01}, 0.001\}, \rho = \{900, 90, 9, \mathbf{0.999}, 0.99, 0.9\}, \nu = \{10, 1, \mathbf{0.1}\}$
RON	$\tau = \{0.76, \mathbf{0.076}\}, \rho = \{900, 90, 9, \mathbf{0.9}\}, \nu = \{10, \mathbf{1}, 0.1\}, \epsilon = \{8, \mathbf{0.8}\} \pm \{\mathbf{1}, 0.5\}, \gamma = \{\mathbf{4}, 0.4\} \pm \{\mathbf{1}, 0.5\}$
npCIFAR-10	
hcoRNN	$\epsilon = 12.7 \pm \{1, \mathbf{0.5}\}, \gamma = 1.3 \pm \{1, \mathbf{0.5}\}, \tau = \{0.76, \mathbf{0.076}\}$
Leaky ESN	$\alpha = \{1, 0.5, 0.1, 0.01, \mathbf{0.001}\}, \rho = \{\mathbf{900}, 90, 9, 0.9\}, \nu = \{10, \mathbf{1}, 0.1\}$
RON	$\tau = \{0.34, \mathbf{0.034}\}, \rho = \{900, 90, \mathbf{9}, 0.9\}, \nu = \{10, 1, \mathbf{0.1}\}, \epsilon = \{\mathbf{12.7}, 1.27\} \pm \{1, \mathbf{0.5}\}, \gamma = \{13, \mathbf{1.3}\} \pm \{1, \mathbf{0.5}\}$
Lorenz96	
hcoRNN	$\epsilon = \{15, 10, \mathbf{2}, 1\} \pm \{\mathbf{0.5}, 1\}, \gamma = \{15, 10, \mathbf{2}, 1\} \pm \{\mathbf{0.5}, 1\}, \tau = \{1.5, 0.8, \mathbf{0.5}, 0.1, 0.01\}$
Leaky ESN	$\alpha = \{1, \mathbf{0.5}, 0.1\}, \rho = \{900, 90, 9, \mathbf{0.9}\}, \nu = \{10, 1, \mathbf{0.1}\}$
RON	$\tau = \{1, 0.7, 0.5, \mathbf{0.17}, 0.1, 0.05, 0.01, 0.001\}, \rho = \{90, 9, 0.999, \mathbf{0.99}, 0.9\}, \nu = \{10, 1, \mathbf{0.1}\}, \epsilon = \{\mathbf{10}, 5, 2, 1\} \pm \{1, \mathbf{0.5}\}, \gamma = \{\mathbf{10}, 5, 2, 1\} \pm \{\mathbf{1}, 0.5\}$
Mackey-Glass	
hcoRNN	$\epsilon = \{\mathbf{10}, 2, 1\} \pm \{\mathbf{0.5}, 1\}, \gamma = \{10, \mathbf{2}, 1\} \pm \{\mathbf{0.5}, 1\}, \tau = \{0.8, \mathbf{0.1}\}$
Leaky ESN	$\alpha = \{\mathbf{1}, 0.5, 0.1\}, \rho = \{900, 90, 9, \mathbf{0.9}\}, \nu = \{10, \mathbf{1}, 0.1\}$
RON	$\tau = \{5, 2, 1, 0.5, 0.3, 0.2, \mathbf{0.17}, 0.05\}, \rho = \{0.999, 0.99, \mathbf{0.9}\}, \nu = \{\mathbf{10}, 1, 0.1\}, \epsilon = \{5, \mathbf{2}, 1\} \pm \{1, \mathbf{0.5}\}, \gamma = \{15, 10, 5, \mathbf{2}, 1\} \pm \{\mathbf{1}, 0.5\}$
FordA	
hcoRNN	$\epsilon = \{5, 3, \mathbf{0.5}\} \pm \{5, 3, \mathbf{0.5}\}, \gamma = \{5, 3, \mathbf{1}\} \pm \{5, \mathbf{1}, 0.5\}, \tau = \{\mathbf{0.25}, 0.1, 0.05, 0.01\}$
Leaky ESN	$\alpha = \{1, 0.9, 0.7, \mathbf{0.5}, 0.1, 0.01, 0.001, 0.0001\}, \rho = \{7, 5, 2, 0.999, \mathbf{0.99}, 0.9, 0.7, 0.5\}, \nu = \{50, 10, 5, 1, \mathbf{0.5}, 0.1.0.01, 0.001\}$
RON	$\tau = \{\mathbf{0.2}, 0.05, 0.01\}, \rho = \{9, \mathbf{0.9}\}, \nu = \{10, 1, \mathbf{0.1}\}, \epsilon = \{\mathbf{5}, 0.5\} \pm \{\mathbf{5}, 2.5, 0.5\}, \gamma = \{3, \mathbf{1}\} \pm \{1, \mathbf{0.5}, 0.25\}$
Adiac	
hcoRNN	$\epsilon = \{5, 1, \mathbf{0.5}\} \pm \{5, 1, \mathbf{0.5}\}, \gamma = \{5, 3, \mathbf{1}\} \pm \{5, \mathbf{1}, 0.5\}, \tau = \{\mathbf{0.25}, 0.1, 0.05, 0.01\}$
Leaky ESN	$\alpha = \{1, 0.9, 0.7, 0.5, 0.1, 0.01, 0.001, \mathbf{0.0001}\}, \rho = \{7, 5, 2, 0.999, \mathbf{0.99}, 0.9, 0.7, 0.5\}, \nu = \{\mathbf{50}, 10, 5, 1, 0.5, 0.1.0.01, 0.001\}$
RON	$\tau = \{0.05, \mathbf{0.01}\}, \rho = \{\mathbf{9}, 0.9\}, \nu = \{\mathbf{10}, 1, 0.1\}, \epsilon = \{\mathbf{5}, 0.5\} \pm \{5, 2.5, \mathbf{0.5}\}, \gamma = \{\mathbf{3}, 1\} \pm \{\mathbf{1}, 0.5, 0.25\}$

$$\leq \max(\|\mathbf{M}\|, \|\mathbf{N}\|)^2 \left(\left\| \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \right\|^2 \right) =$$

$$= \max(\|\mathbf{M}\|, \|\mathbf{N}\|)^2.$$

□

Lemma B.1 allows us to prove Theorem 3.1 whose statement we report here below.

Theorem. *The norm of the Jacobian matrix of the hcoRNN and RON models admit the following upper bound*

$$||\mathbf{J}_k|| \leq \max(\eta + \tau^2\sigma, \xi) + \tau \max(\xi, \gamma_{max} + \sigma). \quad (27)$$

In particular, for $\tau \ll 1$, and assuming $\varepsilon_{min} > 0$, and $\gamma_{max} \geq 1$, the bound reads

$$1 + \tau(\gamma_{max} + \sigma) + O(\tau^2). \quad (28)$$

Proof. We decompose the Jacobian of a RON in the sum of two matrices, one diagonal one anti-diagonal, as follows

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{I} - \tau^2 \mathrm{diag}(\boldsymbol{\gamma}) + \tau^2 \mathbf{S}_k \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon}) \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \tau(\mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon})) \\ \tau \mathbf{A}_k & \mathbf{0} \end{bmatrix}. \quad (29)$$

The upper left block of the diagonal matrix in eq. (29) admits the following bound.

$$||\mathbf{I} - \tau^2 \mathrm{diag}(\boldsymbol{\gamma}) + \tau^2 \mathbf{S}_k \mathbf{W}|| \leq \quad (30)$$

$$\leq ||\mathbf{I} - \tau^2 \mathrm{diag}(\boldsymbol{\gamma})|| + \tau^2 ||\mathbf{S}_k \mathbf{W}|| \leq \quad (31)$$

$$\leq \max_j |1 - \tau^2 \gamma_j| + \tau^2 ||\mathbf{W}|| = \eta + \tau^2 \sigma, \quad (32)$$

where we used the triangle inequality, and the fact that $||\mathbf{S}_k \mathbf{W}|| \leq ||\mathbf{W}||$, due to the definition of $\mathbf{S}_k$.

The bottom right block of the diagonal matrix in eq. (29), is itself diagonal, and admits the following exact estimation.

$$||\mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon})|| = \max_j |1 - \tau \varepsilon_j| = \xi. \quad (33)$$

The upper right block of the diagonal matrix in eq. (29), is itself diagonal, and admits the following exact estimation.

$$||\tau(\mathbf{I} - \tau \mathrm{diag}(\boldsymbol{\varepsilon}))|| = \tau \max_j |1 - \tau \varepsilon_j| = \tau \xi. \quad (34)$$

The bottom left block of the diagonal matrix in eq. (29)

admits the following bound.

$$||\tau(\mathbf{S}_k\mathbf{W} - \text{diag}(\boldsymbol{\gamma}))|| \leq \tau(||\mathbf{S}_k\mathbf{W}|| + ||\text{diag}(\boldsymbol{\gamma})||) \leq \tag{35}$$

$$\leq \tau(||\mathbf{W}|| + \gamma_{\max}) = \tau(\sigma + \gamma_{\max}). \tag{36}$$

Putting together eqs. (30)-(36), and Lemma B.1, we obtain

$$||\mathbf{J}_k|| \leq \max(\eta + \tau^2\sigma, \xi) + \tau \max(\xi, \sigma + \gamma_{\max}),$$

which is the thesis.

In particular, for small enough values of τ we have that $\tau\varepsilon_{\max} \leq 1$, and $\tau^2\gamma_{\max} \leq 1$, which in turns imply that $\xi = 1 - \tau\varepsilon_{\min}$, and that $\eta = 1 - \tau^2\gamma_{\min}$, respectively. Furthermore, assuming that $\varepsilon_{\min} > 0$, and $\gamma_{\max} \geq 1$, we have that $\xi < 1 \leq \sigma + \gamma_{\max}$. Therefore, the bound reads

$$\max(1 - \tau^2\gamma_{\min} + \tau^2\sigma, 1 - \tau\varepsilon_{\min}) + \tau(\sigma + \gamma_{\max}).$$

Finally note that for $\varepsilon_{\min} > 0$, and small enough $\tau \ll 1$, we have that $\tau^2(\gamma_{\min} - \sigma) < \tau\varepsilon_{\min}$, and so that $1 - \tau\varepsilon_{\min} \leq 1 - \tau^2\gamma_{\min} + \tau^2\sigma$. Hence, the bound has the following expansion for small values of τ

$$1 + \tau(\gamma_{\max} + \sigma) + O(\tau^2). \tag{37}$$

□

C Contractivity for the particular case of $\boldsymbol{\varepsilon} \equiv \frac{1}{\tau}$

We already noticed that for the particular case of $\boldsymbol{\varepsilon} \equiv \frac{1}{\tau}$ the **z**-dynamics of the hcoRNN and RON equations become unidirectionally driven by the **y**-dynamics. In such a case, we can focus only on the **y**-dynamics which reads

$$\mathbf{y}_{k+1} = (\mathbf{I} - \tau^2\text{diag}(\boldsymbol{\gamma}))\mathbf{y}_k + \tau^2 \tanh(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b}). \tag{38}$$

We provide the following sufficient conditions for contraction in the particular case of (38).

Proposition C.1. *In the particular case of $\boldsymbol{\varepsilon} \equiv \frac{1}{\tau}$, the hcoRNN and RON models are contractive whenever*

(i) $\sigma < \gamma_{min}$, if $\tau^2(\gamma_{min} + \gamma_{max}) \leq 2$;

(ii) $\sigma < \frac{2 - \tau^2\gamma_{max}}{\tau^2}$, if $\tau^2(\gamma_{min} + \gamma_{max}) > 2$.

Proof. The Jacobian of eq. (38) reads $\mathbf{J}_k = \mathbf{I} + \tau^2\mathbf{A}_k = (\mathbf{I} - \tau^2\text{diag}(\boldsymbol{\gamma})) + \tau^2\mathbf{S}_k\mathbf{W}$. Therefore, it holds $||\mathbf{J}_k|| \leq ||\mathbf{I} - \tau^2\text{diag}(\boldsymbol{\gamma})|| + ||\tau^2\mathbf{S}_k\mathbf{W}|| \leq \eta + \tau^2\sigma$. Thus, $||\mathbf{J}_k|| < 1$ holds whenever $\sigma < \frac{1 - \eta}{\tau^2}$. Finally note that, due to the definition of η, there are two possibilities, either $\eta = 1 - \tau^2\gamma_{\min}$, if $\tau^2(\gamma_{\min} + \gamma_{\max}) \leq 2$, or $\eta = \tau^2\gamma_{\max} - 1$, if $\tau^2(\gamma_{\min} + \gamma_{\max}) > 2$. The first case implies the thesis of (i), while the second case implies the thesis of (ii). □

Note that, in order for (i) and (ii) to hold in Proposition C.1, two necessary conditions must hold, namely $\gamma_{\min} \geq 0$, and $\tau^2\gamma_{\max} \leq 2$.

D Sufficient conditions for a contractive RON

Proposition D.1. ***Sufficient conditions.*** *If $\frac{\xi - \eta}{\tau^2} \leq \xi - \gamma_{max}$ then the hcoRNN and RON models are asymptotically uniformly stable whenever one of the following three conditions holds:*

- $\sigma \leq \frac{\xi - \eta}{\tau^2}$, *and* $\xi < \frac{1}{1 + \tau}$,
- $\frac{\xi - \eta}{\tau^2} < \sigma \leq \xi - \gamma_{max}$, *and* $\sigma < \frac{1 - \tau\xi - \eta}{\tau^2}$,
- $\sigma \geq \xi - \gamma_{max}$, *and* $\sigma < \frac{1 - \eta - \tau\gamma_{max}}{\tau(1 + \tau)}$.

If $\xi - \gamma_{max} < \frac{\xi - \eta}{\tau^2}$ then the RON model is stable whenever one of the following three conditions hold:

- $\sigma \leq \xi - \gamma_{max}$, *and* $\xi < \frac{1}{1 + \tau}$,
- $\xi - \gamma_{max} < \sigma \leq \frac{\xi - \eta}{\tau^2}$, *and* $\sigma < \frac{1 - \xi}{\tau} - \gamma_{max}$,
- $\sigma \geq \frac{\xi - \eta}{\tau^2}$, *and* $\sigma < \frac{1 - \eta - \tau\gamma_{max}}{\tau(1 + \tau)}$.

Proof. Recall the upper bound of the Jacobian found in Theorem 3.1, that we denote for the purpose of the proof as

$$c = \max(\eta + \tau^2\sigma, \xi) + \tau \max(\xi, \sigma + \gamma_{\max}). \tag{39}$$

The proof consists in studying the inequality $c \leq 1$. The proof is divided in 4 cases.

CASE 1.
Assume that $\eta + \tau^2\sigma \leq \xi$ and $\gamma_{\max} + \sigma \leq \xi$. These

assumptions hold if and only if $\sigma \leq \min(\frac{\xi-\eta}{\tau^2}, \xi - \gamma_{\max})$. If such assumptions are true, then the constant (39) reads $c = \xi + \tau\xi$. Therefore, by Theorem 3.1, the Jacobian has norm less than 1 whenever $\xi < \frac{1}{1+\tau}$.

CASE 2.
Assume that $\eta + \tau^2\sigma \geq \xi$ and $\gamma_{\max} + \sigma \leq \xi$. These assumptions hold if and only if $\frac{\xi-\eta}{\tau^2} \leq \sigma \leq \xi - \gamma_{\max}$. If such assumptions are true, then the constant (39) reads $c = \eta + \tau^2\sigma + \tau\xi$. Therefore, by Theorem 3.1, the Jacobian has norm less than 1 whenever $\sigma < \frac{1-\tau\xi-\eta}{\tau^2}$.

CASE 3.
Assume that $\eta + \tau^2\sigma \leq \xi$ and $\gamma_{\max} + \sigma \geq \xi$. These assumptions hold if and only if $\xi - \gamma_{\max} \leq \sigma \leq \frac{\xi-\eta}{\tau^2}$. If such assumptions are true, then the constant (39) reads $c = \xi + \tau(\sigma + \gamma_{\max})$. Therefore, by Theorem 3.1, the Jacobian has norm less than 1 whenever $\sigma < \frac{1-\xi}{\tau} - \gamma_{\max}$.

CASE 4.
Assume that $\eta + \tau^2\sigma \geq \xi$ and $\gamma_{\max} + \sigma \geq \xi$. These assumptions hold if and only if $\sigma \geq \max(\xi - \gamma_{\max}, \frac{\xi-\eta}{\tau^2})$. If such assumptions are true, then the constant (39) reads $c = \eta + \tau^2\sigma + \tau(\sigma + \gamma_{\max})$. Therefore, by Theorem 3.1, the Jacobian has norm less than 1 whenever $\sigma < \frac{1-\xi-\tau\gamma_{\max}}{\tau(1+\tau)}$.

The statement of the Proposition D.1 organises results depending on whether $\frac{\xi-\eta}{\tau^2} \leq \xi - \gamma_{\max}$, or vice versa. □

Table 6: Test Accuracy for sMNIST, psMNIST and npCIFAR-10 and test NRMSE for Lorenz96. Contractive-RON refers to a RON satisfying the *sufficient* conditions for contractivity.

	Contractive-RON	RON
sMNIST ↑	0.15	0.95
psMNIST ↑	0.22	0.90
npCIFAR-10 ↑	0.10	0.33
Lorenz96 ↓	3.9×10^{-1}	4.9×10^{-4}

E Proof of Theorem 3.3

The proof is a straightforward application of the Bauer-Fike's theorem (Bauer and Fike, 1960) that we report here for ease of comprehension.

Theorem E.1 (Bauer-Fike)**.** *Let* $\mathbf{D}$ *be a diagonalisable matrix, and let* $\mathbf{H}$ *be the eigenvector matrix such that* $\mathbf{D} = \mathbf{H}\mathbf{\Lambda}\mathbf{H}^{-1}$ *where* $\mathbf{\Lambda}$ *is the diagonal matrix of the eigenvalues of* $\mathbf{D}$*. Let* $\mathbf{E}$ *be an arbitrary matrix of the same dimension of* $\mathbf{D}$*. Then, for all* μ *eigenvalues of* $\mathbf{D} + \mathbf{E}$*, there exists an eigenvalue* λ *of* $\mathbf{D}$ *such that*

$$|\mu - \lambda| \leq \|\mathbf{H}\|\|\mathbf{H}^{-1}\|\|\mathbf{E}\|. \tag{40}$$

Let us denote

$$\mathbf{E}_k = \begin{bmatrix} \tau^2\mathbf{S}_k\mathbf{W} & \tau\big(\mathbf{I} - \tau\mathrm{diag}(\boldsymbol{\varepsilon})\big) \\ \tau\mathbf{A}_k & \mathbf{0} \end{bmatrix}. \tag{41}$$

The norm of the matrix $\mathbf{E}_k$ can be bounded as stated in the following lemma.

Lemma E.2. *The matrix* $\mathbf{E}_k$ *admits the following upper bound*

$$||\mathbf{E}_k|| \leq C, \tag{42}$$

where C *is defined as follows*

$$C = \tau^2\sigma + \tau\max\big(\xi, \gamma_{max} + \sigma\big). \tag{43}$$

Proof. We decompose the matrix $\mathbf{E}_k$ in its diagonal and antidiagonal parts, and apply Lemma B.1 obtaining the thesis. □

Then, Theorem E.1 in combination with Lemma E.2 give us all the ingredients to prove Theorem 3.3, whose statement we report here below.

Theorem. *For all* μ *eigenvalues of the Jacobian of the hcoRNN and RON models there exists a point* $\lambda \in \{\ 1 - \tau^2\ \gamma_j\ ,\ 1 - \tau\ \varepsilon_j\ \}_{j=1}^N$ *such that*

$$|\mu - \lambda| \leq C, \tag{44}$$

where $C = \tau^2\sigma + \tau\max\big(\xi, \gamma_{max} + \sigma\big)$.

Proof. We decompose the Jacobian in the sum of two matrices as follows

$$\mathbf{J}_k = \begin{bmatrix} \mathbf{I} - \tau^2\mathrm{diag}(\boldsymbol{\gamma}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \tau\mathrm{diag}(\boldsymbol{\varepsilon}) \end{bmatrix} + \begin{bmatrix} \tau^2\mathbf{S}_k\mathbf{W} & \tau\big(\mathbf{I} - \tau\mathrm{diag}(\boldsymbol{\varepsilon})\big) \\ \tau\mathbf{A}_k & \mathbf{0} \end{bmatrix}, \tag{45}$$

and apply the Bauer-Fike's Theorem E.1, choosing $\mathbf{D} = \begin{bmatrix} \mathbf{I} - \tau^2\mathrm{diag}(\boldsymbol{\gamma}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \tau\mathrm{diag}(\boldsymbol{\varepsilon}) \end{bmatrix}$, and $\mathbf{E} = \mathbf{E}_k$ as defined in eq. (41). Noticing that $\mathbf{D}$ is already diagonalised, i.e. $\mathbf{D} = \mathbf{\Lambda}$, thus the eigenvector matrix $\mathbf{H}$ is the identity matrix, and the eigenspectrum of $\mathbf{D}$ is the set of all the points $\{1 - \tau^2\gamma_j, 1 - \tau\varepsilon_j\}_{j=1}^N$. The norm of the matrix $\mathbf{E}_k$ is bounded with C as stated in Lemma E.2. □

F Proof of Proposition 3.4

Lemma F.1. *If the input-free hcoRNN and RON models are asymptotically stable, then $C \leq 1$, where C is defined in eq.* (43)*.*

Proof. For an input-free hcoRNN and RON it is sufficient to have a single eigenvalue outside the unit circle to lose asymptotic stability. By logical contraposition it follows that having all eigenvalues inside the unit circle is a necessary condition for asymptotic stability. We make use of eq. (18) to impose all eigenvalues inside the unit circle. In particular, the inequalities to satisfy can be expressed in terms of $\gamma_{\min}, \gamma_{\max}, \varepsilon_{\min}, \varepsilon_{\max}$, and are the following

$$1 - \tau^2\gamma_{\min} + C \leq 1, \tag{46}$$
$$1 - \tau^2\gamma_{\max} - C \geq -1, \tag{47}$$
$$1 - \tau\varepsilon_{\min} + C \leq 1, \tag{48}$$
$$1 - \tau\varepsilon_{\max} - C \geq -1. \tag{49}$$

The above inequalities can be rewritten as follows

$$C \leq \tau^2\gamma_{\min} \leq \tau^2\gamma_{\max} \leq 2 - C, \tag{50}$$
$$C \leq \tau\varepsilon_{\min} \leq \tau\varepsilon_{\max} \leq 2 - C. \tag{51}$$

We deduce that a necessary condition for eqs. (50)-(51) to hold is that $C \leq 1$. In fact, if $C > 1$, then $2 - C < C$, and eqs. (50)-(51) are never satisfied. □

We use Lemma F.1 to deduce necessary conditions on the hyperparameter σ for an input-free hcoRNN and RON to be asymptotically stable.

Proposition. *If the input-free hcoRNN and RON models are asymptotically stable, then either one of the two cases must hold true*

- *if $\sigma > \xi - \gamma_{max}$, then $\sigma \leq \dfrac{1 - \tau\gamma_{max}}{\tau + \tau^2}$.*
- *if $\sigma \leq \xi - \gamma_{max}$, then $\sigma \leq \dfrac{1 - \tau\xi}{\tau^2}$.*

Proof. If there is asymptotic stability then Lemma F.1 implies that $C \leq 1$.
Let's first assume that $\sigma > \xi - \gamma_{\max}$. The constant C, in such case, reads $C = \tau^2\sigma + \tau(\gamma_{\max} + \sigma)$. Imposing $C \leq 1$ translates in the condition $\sigma \leq \dfrac{1 - \tau\gamma_{\max}}{\tau + \tau^2}$.
Now let's assume that $\sigma \leq \xi - \gamma_{\max}$. In this case the constant reads $C = \tau^2\sigma + \tau\xi$. Imposing that $C \leq 1$ gives us the upper bound $\sigma \leq \dfrac{1 - \tau\xi}{\tau^2}$. □