EMERGE

WP4 Learning and Evolutionary Awareness

D4.1 First version of the ACS

Version: 1.0

Date: 29/03/2024

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Document control

Project title	Emergent awareness from minimal collectives
Project acronym	EMERGE
Call identifier	HORIZON-EIC-2021-PATHFINDERCHALLENGES-01-01
Grant agreement	101070918
Starting date	01/10/2022
Duration	48 months
Project URL	http://eic-emerge.eu
Work Package	WP4 Learning and Evolutionary Awareness
Deliverable	D4.1 First version of the ACS
Contractual Delivery Date	M18
Actual Delivery Date	M18
Nature[1]	OTHER
Dissemination level[2]	PU
Lead Beneficiary	UNIPI
Editor(s)	Claudio Gallicchio
Contributor(s)	Andrea Ceni, Andrea Cossu, Claudio Gallicchio, Vincenzo Lomonaco, Valerio De Caro, Jingyue Liu, Maximilian Stölzle
Reviewer(s)	Cosimo Della Santina, TUD
Document description	This document presents the first version of the EMERGE learning framework built on the archetype concept, the so-called Archetype Computing System (ACS). We first introduce the current state-of-the-art in related deep learning areas, including Reservoir Computing, Physics-inspired neural networks, and learning algorithms beyond backpropagation. We then present our main result, which is given by the introduction of the Random Oscillators Network (RON), a new computational model based on an internal dynamical layer of untrained archetype oscillators.

[1]R: Document, report (excluding the periodic and final reports); DEM: Demonstrator, pilot, prototype, plan designs; DEC: Websites, patents filing, press & media actions, videos, etc.; DATA: Data sets, microdata, etc.; DMP: Data management plan; ETHICS: Deliverables related to ethics issues.; SECURITY: Deliverables related to security issues; OTHER: Software, technical diagram, algorithms, models, etc.

[2]PU – Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page); SEN – Sensitive, limited under the conditions of the Grant Agreement; Classified R-UE/EU-R – EU RESTRICTED under the Commission Decision No2015/444; Classified C-UE/EU-C – EU CONFIDENTIAL under the Commission Decision No2015/444; Classified S-UE/EU-S – EU SECRET under the Commission Decision No2015/444

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

	We also introduce our early results on networks of archetype-inspired neural systems, leveraging novel neural-inspired units' connectors, and our preliminary analysis on lifelong learning. Finally, we introduce the software framework for the ACS.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Version control

Version[3]	Editor(s) Contributor(s) Reviewer(s)	Date	Description
0.1	Claudio Gallicchio	05.02.2024	Document Created. TOC proposed.
0.3	Andrea Cossu, Andrea Ceni, Davide Bacciu	21.02.2024	First draft of the document.
0.4	Andrea Cossu, Andrea Ceni, Davide Bacciu	04.03.2024	Second draft of the document.
0.7	Andrea Cossu, Andrea Ceni, Vincenzo Lomonaco	08.03.2024	Third draft of the document, with revisions in all sections.
0.8	Claudio Gallicchio	12.03.2024	Document finished. Added Document Description, Abstract, Introduction and Conclusions sections.
0.85	Cosimo Della Santina	19.03.2024	Document reviewed.
0.9	Claudio Gallicchio	26.03.2024	Document revised by the editor.
1.0	Davide Bacciu	29.03.2024	Document released.

[3] 0.1 – TOC proposed by editor; 0.2 – TOC approved by reviewer; 0.4 – Intermediate document proposed by editor; 0.5 – Intermediate document approved by reviewer; 0.8 – Document finished by editor; 0.85 – Document reviewed by reviewer; 0.9 – Document revised by editor; 0.98 – Document approved by reviewer; 1.0 – Document released by Project Coordinator.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Abstract

This document introduces the first version of the Archetype Computing System (ACS), a computational framework that employs archetypes to facilitate computation implemented via dynamical systems, embodying a novel computational paradigm where intelligence and awareness emerge from the complex interplay of minimal, dynamically interacting units, mirroring the adaptability and emergent properties akin to neural processes in biological systems. This initial iteration of ACS is grounded in the principles of designing novel intelligent information processing units, paving the way to life-long and evolutionary adaptation, and including early results on networks and connectors.

The core of the document elaborates on the introduction of Random Oscillators Networks (RON) a novel computational model inspired by the physical behaviour of oscillator archetypes (D3.1, Sections 1.2 and 1.3). RON provides an innovative strategy for creating efficiently trainable recurrent neural networks in the field of deep learning, marking a significant shift from traditional design approaches.

We show the fundamental principles of information processing via RON systems, both in theory and in benchmarking applications, highlighting the natural ability of RON to effectively propagate information on long temporal ranges, and the exceptional trade-off between predictive performance and cost of training.

The document presents an exhaustive overview of the state-of-the-art techniques in related deep learning areas, including Reservoir Computing, Physics-inspired Neural Networks, and alternative learning strategies that circumvent the limitations of traditional backpropagation algorithms. Furthermore, this deliverable presents a detailed account of the preliminary results obtained from implementing Euler State Networks, Residual Recurrent Neural Networks, and Continuously Deep Recurrent Neural Networks. It also examines the application of relevant neural architectures in scenarios requiring lifelong learning capabilities, showcasing their potential in recalibration and rejection learning tasks.

Finally, this document introduces the preliminary version of the ACS Python library, which is made publicly available[4].

Consortium

The EMERGE consortium members are listed below.

Organization	Short name	Country
Università di Pisa	UNIPI	IT
TU Delft	TUD	NL
University of Bristol	UOB	UK
Ludwig Maximilian University of Munich	LMU	DE
Da Vinci Labs	DVL	FR

[4] https://github.com/EU-EMERGE/archetype-computing-adaptive-system

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Disclaimer

This document does not represent the opinion of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA), and neither the European Union nor the granting authority can be held responsible for any use that might be made of its content.

Neither the EMERGE consortium as a whole, nor a certain party of the EMERGE consortium warrant that the information contained in this document is capable of use, nor that use of the information is free from risk and does not accept any liability for loss or damage suffered by any person using this information.

Acknowledgement

This document is a deliverable of EMERGE project. This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement Nº 101070918.

Table of Contents

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

List of abbreviations

GA	Grant Agreement
CA	Consortium Agreement
IPR	Intellectual Property Rights
WP	Work Package
RON	Random Oscillators Network
ESN	Echo State Network
ACS	Archetype Computing System
ADS	Archetype aDaptye System
RNN	Recurrent Neural Network
RC	Reservoir Computing
V/E	Vanishing/Exploding Gradient
ESP	Echo State Property
GNN	Graph Neural Network
NN	Neural Network
SSM	State Space Model
GPU	Graphical Processing Unit
LNN	Lagrangian Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
CORNN	Coupled Oscillatory Recurrent Neural Network
HCORNN	Heterogeneous Coupled Oscillatory Recurrent Neural Network
ODE	Ordinary Differential Equation
ML	Machine Learning
NRMSE	Normalized Root Mean Squared Error
pRCON	Physically-implementable Randomly Coupled Oscillator Networks

European Innovation Council Funded by the European Union
Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

ISS	Input State Stability
EuSN	Euler State Networks
LTM	Long Term Memorization
ResNet	Residual Network
ResESN	Residual Echo State Network
CDESN	Continuously Deep Echo State Network
MSE	Mean Squared Error
ECE	Expected Calibration Error

List of tables

List of figures

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

1. Introduction

Within the EMERGE project, Work Package 4 (WP4), titled "Learning and Evolutionary Awareness - Adaptive Non-linear Dynamical Systems for Awareness", seeks to advance the development of adaptive computing systems based on the archetypes concepts. This is done by conceptualizing and implementing an Archetype Computing System (ACS) alongside an Archetypes Adapting System (ADS), which are fundamentally engineered to leverage neural-inspired dynamics for computation.

The Archetypes units, connectors, and networks introduced in D3.1 provide a (yet to be completed) list of non-linear dynamical systems from the combination of which we expect the emergence of high-level representations of the self, the collective, and the environment. All the Archetypes of D3.1 are defined by a rigorous mathematical language that fully characterizes their behaviour. The mathematical language of the Archetypes enables their implementation by means of other "languages", like the analogical language of the physical world, for example through mechanical/soft robots and swarms.

The ACS introduced in this deliverable is a computational framework that allows to implement the Archetypes from D3.1 through the language of artificial deep neural networks. We find artificial neural networks to be a convenient and flexible tool to realize Archetypes, due to their ability of modelling complex non-linear dynamical systems through the composition of simple entities. This allows us to use Archetypes to compute and to process input signals. The objective is to enable a new class of computational systems capable of adapting their processing capabilities in response to environmental changes. The elements in the ACS are anticipated to exhibit emergent behaviours and facilitate efficient information processing across various scales.

Due to the different languages used to define the Archetypes (mathematical models) and the ACS (artificial neural networks) this deliverable intentionally uses a different notation with respect to D3.1. This choice allows us to distinguish the Archetypes from their implementation in the ACS. In particular, with respect to D3.1, we denote the external input with $\boldsymbol{u}$ instead of $\boldsymbol{v}$, and the (hidden) state of the system with $\boldsymbol{h}$ instead of $\boldsymbol{y}$. Moreover, whenever needed by the involved computational scheme, we use discretization of continuous time dynamics by means of Euler methods.

The ADS further extends the ACS by endowing its computational elements with the ability to adapt and learn through lifelong and evolutionary learning algorithms.

In this deliverable, we focus on oscillator units that are both harmonic and multistable, described in D3.1, respectively in Sections 1.2 and 1.3. We also consider the neuron-like connector from D3.1, Section 2.2, as it can be easily employed in artificial neural networks models, like the ones in the ACS. Section 2 of this deliverable describes how the neuron-like connector can be used in artificial neural network models by providing examples from existing architectures. The novel Random Oscillators Network (RON) Archetype network, introduced in Section 3 of this deliverable, leverages the harmonic oscillator unit and the neuron-like connector. The resulting Archetype network is the ACS version of the Archetype network described in D3.1, Section 3.2. Moreover, Section 3.5 in this deliverable is dedicated to preliminary stability results on the Archetype network composed of multistable oscillator units coupled with the hyperbolic potential coupling connector, as explained in D3.1, Section 3.1. In Section 4 of this deliverable, we report further preliminary results on artificial neural network models based on linear units coupled with variations of the neuron-like connector that account

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

for skew-symmetric coupling of the units in the network (Section 4.1), augmented with residual connections (Section 4.2), an encode local connectivity patterns between units (Section 4.3).

Overall, this deliverable, D4.1, introduces a first version of the ACS, delineating its theoretical underpinnings, architectural design, and initial outcomes. It emphasizes the role of RON and other innovative neural models in achieving the project's goals. Moreover, this deliverable discusses the initial steps toward developing a Python library for the ACS and ADS, intending to facilitate its application and further development. Although this document is primarily focused on the development and study of new neural models based on archetype units, we are able to anticipate some relevant preliminary findings on learning strategies in the context of lifelong learning.

This document is structured into several sections, as described below.

- Section 2, "State-of-the-Art," offers an in-depth analysis of the current advancements and theoretical foundations in artificial intelligence (AI) and machine learning (ML) that are key to the development of the ACS and the ADS. It provides an overview of Reservoir Computing, Physics-inspired Neural Networks, and non-traditional learning algorithms that surpass backpropagation in efficiency and adaptability. This section situates the work conducted in WP4 within the related AI research landscape, highlighting how these cutting-edge concepts and methodologies inform the design and implementation of adaptive, evolutionary-aware computing systems.
- Section 3, "Random Oscillators Networks," details the conceptualization, implementation, and initial findings related to Random Oscillators Networks (RON) within the framework of the EMERGE project. This section delves into the theoretical underpinnings and practical applications of RON as a novel neural model inspired by the dynamic behaviour of oscillators. It is in this regard important to note that the RON model represents the first neural network based on neurons whose behaviour is directly following the oscillator archetypes analysed in synergy with WP3 activities and described in D3.1, Section 1.
  The content of Section 3 is mostly based on the following papers: [1] and [2], which present the fundamental concepts of the RON model, the theoretical analysis, and the experimental assessment [3]; which introduces the analysis of the sparse topologies for physical implementations.
- Section 4, "Preliminary Results on Archetype Networks," offers insights into the initial achievements in modelling Archetype Networks using neuron-like connectors. It explores the broader concept of integrating neural-inspired architectures for enhancing the system's computational and adaptive learning capabilities. Specifically, the section delves into the properties and the performances of Euler State Networks (published in [4]), Residual Recurrent Neural Networks (with preliminary concepts published in [5]), and Continuously Deep Recurrent Neural Networks (preliminarily described in [6]).
- Section 5, "Preliminary analysis on the Archetype Adapting System" presents our early results on the development and analysis of ADS methodologies. It focuses on two key aspects: learning to reject and calibration, exploring how these mechanisms enhance the neural system's ability to adapt and evolve over time. The calibration study is based on [7].
- Section 6, "Software Framework: Design Considerations of the ACDS Python Library," outlines the development and architectural choices behind the ACS/ADS Python library, a key component for facilitating the exploration and application of the methodologies developed within WP4. It discusses the library's modular design, intended to provide researchers and practitioners with a flexible tool for implementing

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

and testing various configurations of the ACS. This section highlights the library's components, their functionalities, and how they collectively support the operationalization of the ACS for further research and practical applications.

- Section 7 presents some conclusion remarks for the deliverable.
- Beyond the main content, Sections 8 and 9 provide further details on the proof of the main mathematical results and experimental settings, respectively.

2. State of the art

This section provides an overview on the current state-of-the-art on AI research that falls within the scopes of WP4 in the project. The aim is to frame the ground for the technical content presented in the following sections of this document. More specifically, Section 2.1 introduces the fundamental concepts of Reservoir Computing, Section 2.2 provides an overview on physics-inspired design of neural networks architectures, with a focus on neural systems based on oscillators, and Section 2.3 presents the fundamental concepts around bio-inspired learning algorithms that are alternatives to backpropagation.

2.1. Reservoir Computing

Recurrent neural networks (RNNs) [8] are computational models designed to extract features from data with temporal structures. Applications ranges from speech recognition to classification of time series. The most common way of training RNNs is via stochastic gradient descent methods, usually via the backpropagation through-time algorithm [9]. Unfortunately, these methods come with a significant computational effort. Modern hardware unleashed the power of parallel computing techniques, allowing to reduce the computational time of training deep learning models as RNNs. However, the price to pay is a massive energy consumption. Moreover, a fundamental limitation of theoretical nature prevents RNNs to be fully exploited, namely the vanishing/exploding (V/E) gradient issue [10]. In this regard, an appealing alternative is represented by Reservoir Computing (RC) [11], [12], a different paradigm of training RNNs dodging the V/E while being computationally fast, and energy efficient. The flexibility of the RC paradigm offers a suitable theoretical framework for computing with physical substrates [13], [14], [15], [16], for fast and scalable graph neural networks models [17], [18], and for implementing digital twins of real-world nonlinear dynamical systems [19].

The key idea of RC is to inject the input signal into a large random untrained recurrent layer, the *reservoir*. The reservoir provides a high dimensional and heterogeneous set of activations. Hence, this set is used as input for a readout layer which is optimised to fit the desired target signal. The Echo State Network (ESN) [20], [21] provides a popular discrete-time class of RC machines. The name ESN recalls the imagery of the input signal echoing and reverberating within the pool of neuronal activations in the reservoir, which in turn serves as a high-dimensional representation of the past history of the input. Although gradients are not backpropagated in the RC paradigm, the ESN's forward dynamics are ruled by the very same equation of conventional RNNs.

Thus, ESNs inherit a problem which closely relates to the V/E gradient problem of plain RNNs, namely the degradation of memory [22]. As revealed by previous studies [23], [24], the degradation of memory is linked to the nonlinearity of the system in an inherent trade-off. Nonlinear computation and short-term memory are two fundamental aspects of neural systems. Therefore, the existence of a trade-off between them compels to design nonlinear

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

RC systems able to retain as much memory as possible. In fact, the memory capacity is a key feature to reach desirable results in certain learning tasks [25].

As a fundamental baseline, we introduce the Echo State Network (ESN) model, which in its vanilla formulation includes a fixed non-linear reservoir layer and a trainable readout. We consider, in particular, the general case with leaky-integrator recurrent neurons [26]. The state transition equation of the reservoir is given as follows:

$$\mathbf{h}(t) = (1-\tau)\mathbf{h}(t-1) + \tau\phi(\mathbf{W_h}\mathbf{h}(t-1) + \mathbf{W_u}\mathbf{u}(t) + \mathbf{b}), \quad (1)$$

where $\mathbf{y}(t) \in \mathbb{R}^{N_\mathrm{h}}$ and $\mathbf{u}(t) \in \mathbb{R}^{N_x}$ respectively denote the state and input at time t, $\mathbf{W_h} \in \mathbb{R}^{N_\mathrm{h} \times N_\mathrm{h}}$ and $\mathbf{W_u} \in \mathbb{R}^{N_x \times N_h}$ are the reservoir and the input weight matrices, $\mathbf{b} \in \mathbb{R}^{N_h}$ is the bias vector, $\phi(-)$ is an element-wise applied non-linearity (we use $\tanh(\cdot)$), and $\tau \in (0,1]$ is a leakage hyperparameter. The reservoir is typically initialized in the origin, i.e., $\mathbf{h}(0) = \mathbf{0}$. The values in $\mathbf{W}_\mathrm{h}$ are randomly chosen and then re-scaled to have a specific value of the spectral radius (i.e., the maximum length of an eigenvalue), a crucial hyperparameter denoted as ρ. $\mathbf{W_u}$ and $\mathbf{b}$ are randomly initialized from uniform distributions over $(-\omega_x, \omega_x)$, and $(-\omega_b, \omega_b)$, where ω_x and ω_b act respectively as input and bias scaling hyperparameters. The value of ρ is important as it practically determines the dynamic regime of the reservoir layer, and in applications it is often controlled to values not exceeding too much 1.

The ESN architecture also comprises a tunable readout layer, which is typically linear and trained in closed-form by ridge regression. This efficient design, where only the readout layer is trainable, contrasts with fully trainable systems and contributes to the computational advantages of ESNs and RC in general, including simplified and fast training [3], [13], as well as physical implementability in neuromorphic substrates [14]. For time series classification problems, the reservoir is run on each input sequence, and the state calculated for the last time step is used to feed the readout classifier.

ESNs work under the foundational Echo State Property (ESP). An ESN with the ESP, when driven by an input signal, will become entrained by the input and develop a unique internal response signal. Such an internal response is a high-dimensional, nonlinear, unique transform of the input with memory content on top of which we perform regression or classification based on the specific task at hand. This property is crucial for the design and training of RNNs as it ensures that the network's response to inputs is stable, making it possible to train the network to produce desired outputs. The easiest condition to ensure the ESP is to set a reservoir such that the following condition holds:

$$\|\mathbf{W}_\mathrm{h}\| < 1,$$

where $\|\mathbf{W_h}\|$ denotes the matrix norm induced by the Euclidean norm in $\mathbb{R}^{N_h}$, or equivalently the maximum singular value of $\mathbf{W}_\mathrm{h}$. For the rest of this document, we will always use $\|\cdot\|$ to denote the matrix norm induced by the Euclidean norm on $\mathbb{R}^{N_b}$.

In coarse terms, the ESP guarantees the ESN to possess a unique input-driven solution such that all the trajectories originating from different initial conditions (in the infinite past) synchronise with it (in present time). Following the imagery, such a unique input-driven solution would represent the echo of the input signal from the infinite past. An ESN such that $\|\mathbf{W}_\mathrm{h}\| < 1$ is characterised by straight contraction in the phase space at each time step. Therefore, any two different internal states of the RNN, when driven by the same input sequence, will get closer and closer to each other as time flows ahead. Although stable, such

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

a dynamical system risks to have little margin to exploit the transient dynamics for computational purposes, if the contraction is too strong. An ideal situation would be for the RNN to stay in a regime of balance between stable contractive dynamics and unstable chaotic dynamics, i.e. along the *edge of chaos* [27], [28]. This led the RC community to adopt the rule of thumb of setting the reservoir matrix to have spectral radius approximately one [20], [29], [30]. Thus, in the RC literature, when introducing a new model, it is common practice to study the eigenspectrum of the model to get insights on the stability properties. For continuous-time models, the stability region for the zero-input case is represented by the left complex plane, i.e. all eigenvalues with non-positive real parts. While for discrete-time models, the stability region for the zero-input case is represented by the unitary disk, i.e. all eigenvalues with modulus not greater than 1. Hence, the edge of stability results the imaginary axis, for continuous-time models, and the unit complex circle, for discrete-time models.

2.2. Physics-inspired Neural Networks

Neural networks have been proved to be universal approximators [31]. However, the space of possible functions is so vast that usually the optimisation procedure ends in a sub-optimal solution of the problem. Constraining the problem often makes easier to find more educated guesses and so better solutions. In recent years, physics-inspired neural network (PINN) have emerged as a potential solution to address the challenges faced by traditional neural networks [32]. These architectures draw inspiration from the principles and laws of physics, utilising concepts such as conservation laws, symmetries, and local interactions to improve network performance. These novel approaches aim to enhance the capabilities of traditional neural networks by leveraging insights from the field of physics.

Neural networks have revolutionised fields such as computer vision, natural language processing, and speech recognition. However, in certain applications where physical laws and constraints play a crucial role, standard neural networks may fail to capture the underlying physics accurately. To address this limitation, various techniques have been proposed to incorporate physical knowledge into neural networks.

There are several approaches to inform a neural network with physics-based knowledge. A first method is to modify the loss function of the neural network to enforce physical constraints [33]. For example, in the field of fluid dynamics, where the conservation of mass and momentum are fundamental principles, a loss function can be designed to penalize predictions that violate these principles [34] This approach ensures that the neural network learns to respect the laws of physics and produces physically meaningful results. An alternative way is to modify the optimization algorithm to incorporate physical knowledge [35]. For instance, gradient-based optimization algorithms can be augmented with physical constraints [36]. Another popular approach is to incorporate physical knowledge into the architecture of the neural network [37]. The latter approach has been a particularly effective strategy in machine learning long before PINN became popular. For example, instead of using a standard fully connected architecture, a common approach used in computer vision is to use a convolutional architecture that considers the spatial structure of the problem and incorporates physical symmetries. In the field of image segmentation, a physics-inspired neural network can be designed with convolutional layers that have shared weights to enforce translation invariance, ensuring that the network maintains spatial consistency and respects the inherent symmetries in the data. As another example, RNN provides time-scale invariance, and fading memory, both important in the processing of sequential data. Furthermore, Graph Neural Networks

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

(GNNs) [38] are inherently biased to process graph-structured data like social networks, molecule structures, etc.

Now, from a RC perspective the loss function and the optimisation algorithm do not play an important role, since simple linear regression techniques are used. Therefore, we can restrict our search to PINN architectures. Some examples in the context of sequential data processing are given below.

Oscillators-based neural network architectures utilise the behaviour of oscillators, which are physical systems that exhibit rhythmic patterns or oscillations, to mimic the functioning of biological neurons and create a network of interconnected units that can perform computation. By harnessing the properties of oscillators, such as their ability to synchronise and exhibit resonance, these neural network architectures can effectively model and analyse complex phenomena. Few examples of deep learning models leveraging on oscillators can be found in [39] [40]. In this category we can include wave-based models as [41] which enforce spatiotemporal oscillatory motion.

Residual models enforce a bias towards the identity mapping [42]. This architectural bias is achieved building skip connections that interconnect spatially distant layers together, a feature observed also in cortical circuits of the brain. This strategy has been proved very effective [43], and it's currently implemented in almost every deep neural network architecture.

Convolutional models enforce spatial localisation and translational invariance. These models proved to be effective both in computer vision [44] and time series processing [45].

Linear state space models are machine learning models based on State Space Models (SSMs) enforcing linearity as an architectural bias. SSMs have recently raised interest in the deep learning community due to the seminal paper [46]. There, authors established a link between convolutional models and Linear SSMs that allows to train them in parallel via GPU. Other interesting variants of Linear SSMs include DSS [47] and S5 [48], where different structures of the state space matrices are explored. More recently, a Linear SSM incorporating an attention mechanism on the input signal has been proposed [49]. Other than allowing fast training, the linearity enforced in all these models equip them with the capability of long-term propagation of information. A feature that is not easy to gain in strongly nonlinear models.

Recurrent memory cells have been used to process data sequentially exploiting alternative paths of the information flow enforcing an input-dependent selective memory. These models include LSTM [50] and GRU [51]. Recently, other models proposed the construction of memory cells exploiting orthogonal polynomials (e.g. Legendre polynomials) to get expressive representation of the input signal [52].

Lagrangian neural networks (LNNs) [53] are a novel approach that parameterises arbitrary Lagrangians using neural networks. These models enforce symmetries corresponding to conservation laws, e.g. energy and momentum, naturally present in physical systems. LNNs produce energy-conserving models.

In the next section we focus on Oscillators-based NNs since they fit perfectly into the archetype zoo provided in Deliverable D3.1.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

2.2.1 Oscillators-based Neural Networks

In this section we trace a path from the classical harmonic oscillator to a recurrent neural network model composed of many oscillator-based units. We provide a gentle introduction to the damped harmonic oscillator, framing it in the context of fading memory systems. Then we introduce the coRNN model [39].

Damped harmonic oscillator

Harmonic oscillators are at the core of classical mechanics. They describe simple oscillatory motions around an equilibrium point without experiencing any dissipation of energy. The equation of an harmonic oscillator reads as follows:

$$\ddot{y} = -\gamma y. \tag{2}$$

The general solution of the above equation is given by

$$y(t) = A\cos\left(\sqrt{\gamma}t + \psi\right), \tag{3}$$

where the amplitude A, and phase ψ, are uniquely determined by the initial conditions. Here different initial conditions give rise to different solutions.

A key feature required for a dynamical system to be exploited for computational purposes is the fading memory. In rough terms, we aim for a stable dynamical system which can wash out in the long term any dependencies from the initial conditions. Therefore, exploiting a network of randomised oscillators of the form of eq. (5) for computational purposes is impractical. Fading memory can be brought by a damping term into eq. (4), thus introducing a source of energy dissipation. The equation of a damped harmonic oscillator reads as follows:

$$\ddot{y} = -\gamma y - \varepsilon\dot{y}. \tag{4}$$

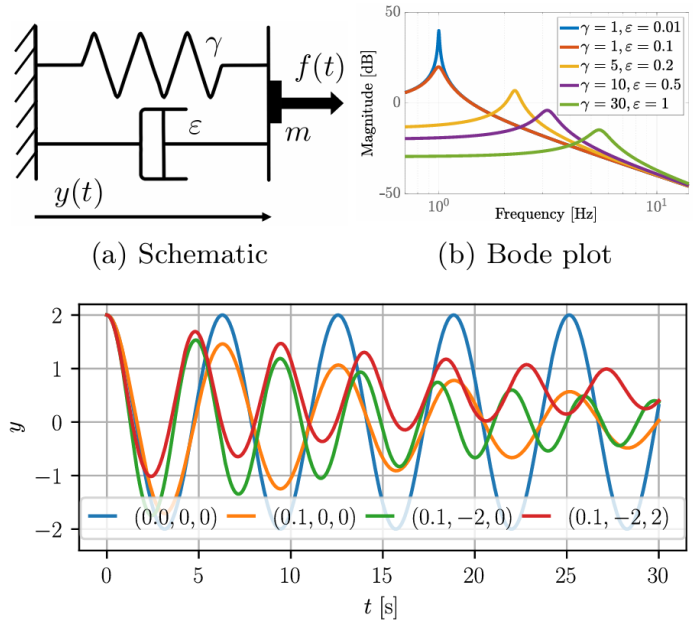
In eq. (6), the γ scalar term relates with the intrinsic frequency of the underlying harmonic oscillator, while the ε scalar term refers to the strength of the damping force (also called friction) exerting against the harmonic oscillator. Any value of $\varepsilon > 0$ induces the system of eq. (6) to converge towards the resting state of $y = 0$. According to the value of ε, two main behaviours can be observed: overdamped dynamics (when $\varepsilon > 2\sqrt{\gamma}$) characterised by exponential decay towards $y = 0$ without any oscillations, or underdamped dynamics (when $0 < \varepsilon < 2\sqrt{\gamma}$) characterised by an oscillatory behaviour with decreasing amplitude in time. In this sense, the damped harmonic oscillator possesses a fading memory property. Often, an external time-varying force $f(t)$ drives the damped oscillator giving rise to the following equation:

$$\ddot{y} = f(t) - \gamma y - \varepsilon\dot{y}. \tag{5}$$

The driven damped harmonic oscillator of eq. (7) emerges in many physical, engineering and biological systems. Such a simple system, sketched in Figure 1(a) for the case of unitary mass, has already several properties that make it ideal to serve as a fundamental cell of our learning strategy.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.



(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 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 stiffness γ = 1 driven by f(t) = tanh(Wy+b) for various values of ε, W, and b.*

First, different choices of γ and ε give rise to quite different transient behaviors, as shown in Figure 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 Figure 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 $E = \dot{y}^2/2 + \gamma y^2/2$. 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 (6) 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 Figure 1(c)). This small change substantially enriches the dynamic spectrum of this simple unit, without changing its fundamental character. Considering a bias term $b \neq 0$ 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, as described in deliverable D3.1.

Coupled oscillatory RNN

The next step is to build a network of input-driven damped harmonic oscillators and use this physically-inspired neural network model to perform computations. Let us denote with vectors $\boldsymbol{\gamma}, \boldsymbol{\varepsilon} \in \mathbb{R}^N$, the characteristic frequencies and damping ratios of each oscillator in the network. Then, the following equation describes a network of heterogeneous driven damped harmonic oscillators:

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

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

where $\odot$ denotes the point-wise multiplication of vectors. 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

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

where $\mathbf{u}(t)$ is the external input driving the network. Being $\mathbf{f}$ a function of all configurations $\mathbf{y}$, this forcing term has the effect of nonlinearly connecting the decoupled oscillators into a network.

In Random Oscillators Networks (see Section 3) we will build a network of many interconnected Archetype Units of the oscillatory type described here, thoroughly studying its theoretical properties and evaluating empirically its performance in time series processing tasks.

2.3. Learning beyond backpropagation

Learning in archetype units and networks requires to update a set of *adaptive* parameters to fit a given objective. The adaptive parameters control the behavior of the archetype. One of the most popular approaches for learning is the stochastic gradient descent update [54]: given a loss function computing the error of the model prediction on a given input, stochastic gradient descent updates each adaptive parameter towards the direction of steepest loss descent. By iterating this process multiple times on a given set of data, the model moves towards a region of parameters where the loss is low, hence where the model predictions are accurate. The steepest descent direction is computed by the gradient of the loss with respect to the adaptive parameters. The backpropagation algorithm [55] allows to compute the gradient for any differentiable function. Stochastic gradient descent and backpropagation are the main tools used nowadays to learn with artificial neural networks, due to their effectiveness on a wide range of tasks [56].

One of the key ideas behind archetypes is that they can be implemented on different physical substrates: from computer simulations to soft robots and swarms. Unfortunately, while the backpropagation algorithm can be efficiently run on a computer simulation, it cannot be easily adapted to work on a physical substrate, where computation is performed by the physical system itself [57]. On one side, backpropagation requires the learning model to have two separate “circuits”: one circuit computes the evolution of the model over time for any external input stimuli, and the other circuit computes the gradient for each parameter. The circuits need to be the same, both in structure and in connections strength, and they cannot interfere with each other. When one circuit is operating, the other needs to be shut down. These assumptions [58], [59] make backpropagation difficult or even impossible to be implemented on most physical systems.

Backpropagation-free learning algorithms exist, and they are designed to overcome the aforementioned limitations of backpropagation [57], [59], [60], [61]. Direct feedback alignment [62], [63] employs a random update circuit, bypassing the issue in backpropagation of circuits with identical connections strength. Unlike feedback alignment, direct feedback alignment allows projections of the error signal to layers that are not directly connected with the output layer. The forward-forward algorithm [64] removes the need for a separate update circuit and only keeps the “forward” circuit that computes the state evolution. The circuit is used in two

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

distinct phases to compute the parameters update. Similarly, Equilibrium Propagation [61], [65] does not require two separate circuits. During a first phase, Equilibrium Propagation lets the system freely evolve under a given external input until it reaches a steady state (the fixed, or equilibrium point). During the second phase, the steady state is used as a starting point to nudge the system towards a state where the prediction for the same external input is the correct one. The second phase requires the target answer for the input and a loss function to measure the accuracy of the prediction. The system eventually reaches a second steady state. The difference between the first steady state and the second steady state is used to update the parameters of the model. Interestingly, the update is local to each parameter, provided that the first steady state can be accessed later after the second phase.

We plan to leverage backprop-free algorithms to adapt the parameters of the archetypes. When possible, mostly in computer simulations, we will also leverage backpropagation. Equilibrium Propagation is a promising approach to learning with archetypes since i) it does not require two separate circuits, ii) the steady states can be easily obtained in a physical system by letting the system itself evolve, iii) the update is local and does not require sharing global information across the whole system/archetype.

3. Random Oscillators Networks

We present a recurrent neural network made of oscillators. The oscillators are very flexible archetypes (presented in D3.1, Section 1) that can be easily connected into networks of archetypes.

In line with the Archetype Networks of oscillators presented in D3.1, Section 3, we build our Random Oscillators Network (RON) [1], [2] by starting from a continuous-time dynamical system composed by a set of damped oscillators. Introducing the variable $\mathbf{z} = \dot{\mathbf{y}}$, we get the following first-order system of ODEs:

$$\begin{aligned}\dot{\mathbf{y}} &= \mathbf{z}, \\ \dot{\mathbf{z}} &= \tanh\left(\mathbf{W}\mathbf{y} + \mathbf{V}\mathbf{u}(t) + \mathbf{b}\right) - \boldsymbol{\gamma} \odot \mathbf{y} - \boldsymbol{\varepsilon} \odot \mathbf{z},\end{aligned}$$

In ML terms, the equations above 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.

We discretise the ODE with an implicit (the $\dot{\mathbf{y}}$ equation), and an explicit (the $\dot{\mathbf{z}}$ equation) Euler numerical scheme, via discretisation time step $\tau > 0$.
Definition 2.1. Random Oscillators Network (RON). The RON (Figure 2) is a discrete-time RNN model whose update reads as follows:

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

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

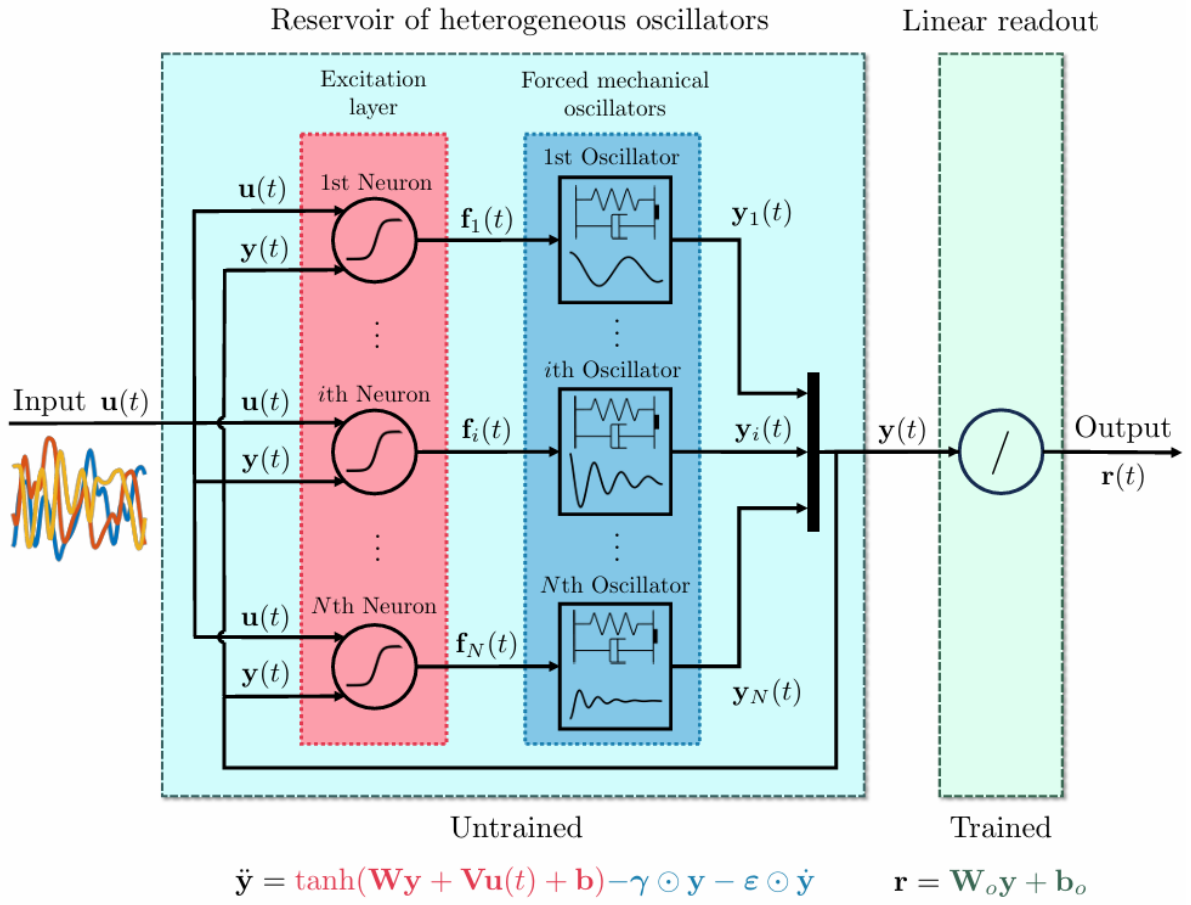


Figure 2. The Random Oscillators Network (RON) consists of N harmonic oscillators forced by coupled neurons with hyperbolic tangent activations. A linear output layer maps the states of the mechanical oscillators in the desired output. This layer is the only one that is adapted during learning.

RON is a component of the ACS that casts an Archetype Network of harmonic oscillators with neuron-like connectors towards a recurrent neural network implementation, where the hidden state is defined as $\mathbf{h}_k = \begin{pmatrix}\mathbf{y}_k \\ \mathbf{z}_k\end{pmatrix}$. The hidden states can be exploited as features encoding crucial temporal information for the processing of the discretised input time series $\mathbf{u}_k$. We use for simplicity only the position of the oscillators $\mathbf{y}_k$ to perform time series processing. Precisely, we stack a linear layer transforming $\mathbf{y}$ to an output state $\mathbf{r}$ as follows:

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

where $\mathbf{W}_o, \mathbf{b}_o$, are weights and biases of the output layer. RON maps input sequences $\mathbf{u}_k$ into output sequences $\mathbf{r}_k$. See Figure 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 [66] 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

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

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 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 but rather to investigate the expressiveness of the physically-inspired discrete-time RNN model. As a consequence, in the remainder, 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.

RON as Leaky-ESN. In the particular case of RON 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 [26], whose state-update equation reads as follows:

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

We can therefore interpret the hyperparameter τ 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 [26]. Remarkably, the Leaky-ESN with linear output layer can accurately learn the climate of chaotic attractors.

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.

3.1. Linear stability analysis of RON

In this section, we perform a linear stability analysis of the RON mode. All the proofs can be found in the Appendix A – Proof of mathematical results. Let us denote the hidden state of our model at time step k as $\mathbf{h}_k = \begin{pmatrix} \mathbf{y}_k \\ \mathbf{z}_k \end{pmatrix}$. Then, the RON model can be defined by the imput-driven state-update equation $\mathbf{h}_{k+1} = \boldsymbol{G}(\mathbf{h}_k, \mathbf{u}_{k+1})$, where $\boldsymbol{G}: \mathbb{R}^{2N} \times \mathbb{R}^I \to \mathbb{R}^{2N}$ is the RON state-update equation. The Jacobian of the $\boldsymbol{G}$ map computed on $(\mathbf{h}_k, \mathbf{u}_{k+1})$, denoted with $\mathbf{J}_k$, reads:

European Innovation Council

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

$$\mathbf{J}_k = \begin{bmatrix} \frac{\partial \mathbf{y}_{k+1}}{\partial \mathbf{y}_k} & \frac{\partial \mathbf{y}_{k+1}}{\partial \mathbf{z}_k} \\ \frac{\partial \mathbf{z}_{k+1}}{\partial \mathbf{y}_k} & \frac{\partial \mathbf{z}_{k+1}}{\partial \mathbf{z}_k} \end{bmatrix} = \\ = \begin{bmatrix} \mathbf{I} + \tau^2 \mathbf{A}_k & \tau(\mathbf{I} - \tau \mathrm{diag}\,(\boldsymbol{\varepsilon})) \\ \tau \mathbf{A}_k & \mathbf{I} - \tau \mathrm{diag}\,(\boldsymbol{\varepsilon}) \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{A}_k &= \mathbf{S}_k \mathbf{W} - \mathrm{diag}\,(\boldsymbol{\gamma}), \\ \mathbf{S}_k &= \mathrm{diag}\left(1 - \tanh^2\left(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b}\right)\right). \end{aligned}$$

A widely known stability condition for RC systems [20] 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. 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 based on the following quantities

$$\begin{aligned} \xi &= \max_j \left|1 - \tau \varepsilon_j\right|, \\ \eta &= \max_j \left|1 - \tau^2 \gamma_j\right|, \\ \sigma &= \| \mathbf{W} \|, \end{aligned}$$

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

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

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

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

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 $\mathbf{y}$-dynamics becomes decoupled from the $\mathbf{z}$-dynamics, as observed in the previous section; we provide sufficient conditions for contractivity for such a particular case in the Appendix A – Proof of mathematical results. In the general case, imposing the upper bound of Theorem 3.1 to be less than 1, we obtain sufficient conditions for a contractive RON, thus a uniformly

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

asymptotically stable RON in particular. For sake of conciseness, these sufficient conditions for a contractive RON are reported in the Appendix A – Proof of mathematical results.

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 [67], [68].

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

$$|\mu - \lambda| \leq C,$$

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

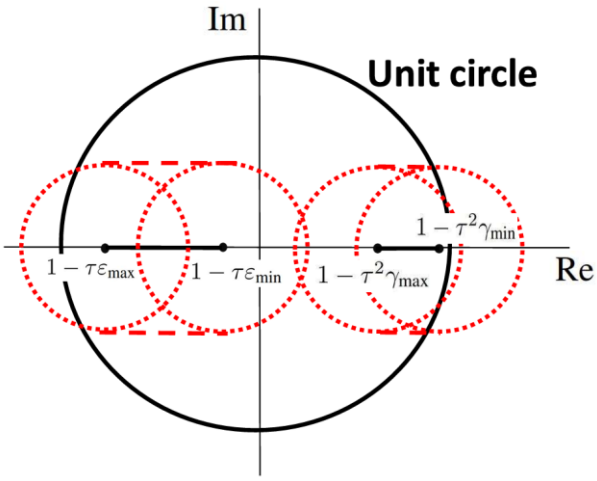


Figure 3. Depiction of the eigenspectrum's bound given by Theorem 3.2 for the Jacobian of an RON model.

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 Figure 3 for a visual representation of this fact, and Figure 4 for few eigenspectrum configurations of a 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.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

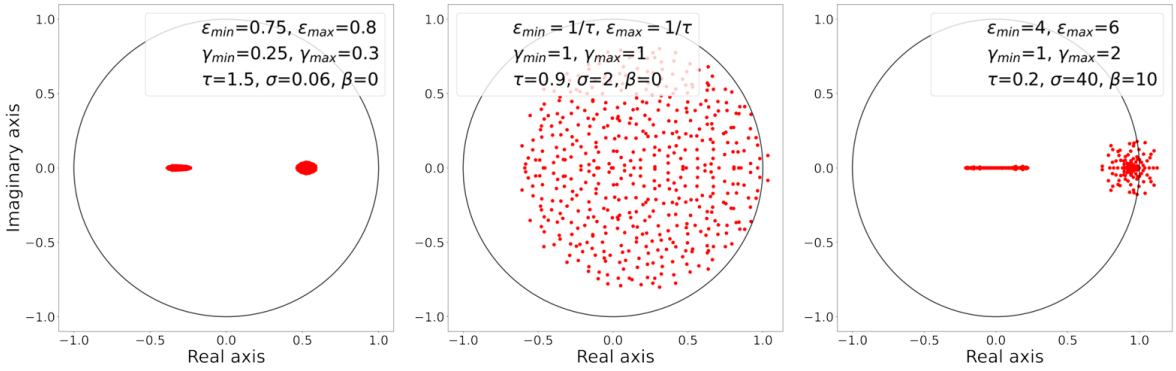


*Figure 4. Jacobian's eigenspectrum eq. (9) for few hyperparameter's combinations. Bias vector has β in all its components, input-to-hidden matrix **V** is zero matrix. **Left:** a case satisfying the sufficient conditions for a contractive RON. **Centre:** a case of RON coinciding with Leaky ESN. **Right:**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

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

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

$$\begin{aligned} C \leq \quad \tau\varepsilon_i \quad &\leq 2 - C, \\ C \leq \quad \tau^2\gamma_i \quad &\leq 2 - C, \end{aligned}$$

for all $i = 1, \dots, N$, where C is defined in Theorem 3.2. The necessary conditions of Proposition 3.3 are derived by imposing that $C \leq 1$.

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

$$\begin{aligned} \varepsilon_{\min} &\geq 0, \\ \gamma_{\min} &\geq 0, \\ \varepsilon_{\max} &\leq 2/\tau, \\ \gamma_{\max} &\leq 2/\tau^2. \end{aligned}$$

Selecting τ values small enough 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 [28], and lately also in typical deep learning applications [69].

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

3.2. Experiments

coRNN. We compare the performance of RON against the coupled oscillatory RNN (coRNN) from [39]. coRNN also builds on a network of oscillators. A coRNN has unique scalar values $\gamma \equiv \gamma, \varepsilon \equiv \varepsilon$, and it is defined by the following equation:

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

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.

Our empirical evaluation focuses on two key RON properties, discussed in the theoretical analysis of Linear stability analysis of RON:

1 We study the impact of weight randomisation by comparing the performance of an RON against fully-trained hcoRNN, coRNN, and LSTM [50], 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 Linear stability analysis of RON.

To guarantee a fair comparison, we adopt the experimental setup of [39], 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 Appendix B – Details on experimental settings. The code to reproduce the experiments is available at https://github.com/AndreaCossu/.

Sequence classification benchmarks. We use 6 classification benchmarks, sMNIST, psMNIST, npCIFAR-10, FordA, Adiac and uWaveGesture. FordA, Adiac and uWaveGesture are from timeseriesclassification.com. 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 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). uWaveGesture is a benchmark that represents different gestures measured from accelerometers.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Chaotic systems forecasting. In [39], 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. 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 [70]. Similar to [21], 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.

3.3. 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 [39], while for FordA, Adiac, uWaveGesture and Mackey-Glass we explore a smaller parametrisation regime (Table 2).

Table 1 Average accuracy (and standard deviation) of RON and other recurrent models.

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

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.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 2 Number of trainable parameters and total training time (in minutes) for each benchmark and model.

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

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 outperforms fully-trained models by two orders of magnitudes in the Lorenz96 task, while surpassing state-of-the-art RC models. 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 the best RON configurations (Table 13 in the Appendix A – Proof of mathematical results) satisfy the guideline values and the necessary conditions for linear stability we found in our analysis. We observed 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 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.

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.
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.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

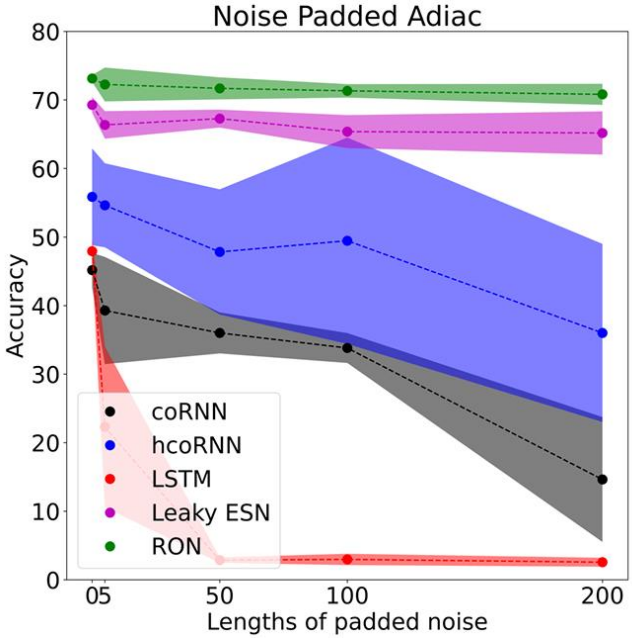


Figure 5. Accuracy on the npAdiac dataset. The x-axis shows the padded sequence length.

Comparison with expRNN. We compare RON against the expRNN model [71], a fully-trainable RNN with orthogonal recurrent weights. Results are shown in Table 3. 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.

Table 3 Accuracy/loss for the expRNN model and RON

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

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. 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.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

3.4. Sparse reservoir topologies for physical implementations of Random Oscillator Networks

Here, we further extend the analysis on the RON approach, by introducing a pool of diverse strategies to sparsify the pattern of connectivity among the elementary units in the reservoir layer of a RON (the interested reader can find further details in [3]).

In deep learning, sparse computation is often still implemented without relying on sparse data structures, but rather by performing dense matrix multiplications on very sparse matrices. This is mainly due to the advantage provided by co-processors like GPUs that do not really thrive on sparse multiplications, yet. However, computational efficiency is only one motivation behind the use of sparse architectures in deep learning. Implementing densely connected neural layers in physical substrates can be expensive and difficult to scale for a large number of units. We focus on the RON model. Such networks can hardly be implemented in a physical substrate (e.g. mechanical oscillators) when they require hundreds or thousands of fully connected oscillators. Such a system is far too complex to be built in the real world. Scaling down the number of oscillators and designing sparse topologies, where each unit needs not to be connected with all others, would solve all the aforementioned issues. But at what cost? Sparsity would give us the flexibility to create a variety of different architectures and to adapt to the constraints coming from the real world, but would the predictive performance of the resulting model still be comparable with its fully-connected counterpart? We answer positively to this question by designing and studying 6 sparse topologies for RON. We chose topologies with very different connection patterns, to cover a wide range of possibilities. Some of them, like the ring topology, are well established in the reservoir computing literature, while some others, like the Toeplitz and the sparse orthogonal, have not been extensively investigated in reservoir computing. Our experiments prove that sparse topologies in RON are effective across different sequence processing tasks and even with a small number of hidden units. We tested different levels of sparsity and we show that, with respect to its fully-connected counterpart, sparse topologies in RON do not show any substantial decrease in predictive performance, not even with $80\%/90\%$ of sparsity in the reservoir connections. Our main finding is that RON is an ideal candidate for the realization of a recurrent network in hardware, mainly because of its computational efficiency due to the untrained recurrent component and its effectiveness when combined with very sparse topologies.

Sparse RON Topologies. We designed 6 different sparse reservoir topologies that we applied to the RON model, as well as the Leaky ESN. In all the following, we assume a recurrent state-update matrix $W \in \mathbb{R}^{N\times N}$. We set a sparsity level of $P\%$, where $P = 0$ corresponds to a fully-connected reservoir matrix. In building sparse topologies of RON, we prioritized structures that are more prone to physical implementations. A fullyconnected network of N oscillators has N^2 interconnections. This prevents RON from mechanical implementations at a large scale. Moreover, often in nature neural networks are characterised by local connections forming relatively sparse topologies. One solution to model local connections while tuning the sparsity of the network is to promote a band-like diagonal structure in the recurrent matrix W, suppressing farfrom-diagonal connections. For this reason, apart from the Sparse Orthogonal case (as discussed below), in all the other cases we sparsify the recurrent matrix $\boldsymbol{W}$ zeroing-out diagonal elements starting with the most far from the main diagonal and proceeding towards the main diagonal until the desired sparsity level is reached. The spectral radius of W is then scaled accordingly.
a) Sparse Orthogonal: (or simply orthogonal). We randomly initialize $\boldsymbol{W}$ uniformly in $[0,1]$.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Then, we compute the QR factorization of $\boldsymbol{W}$ and we take the matrix $\boldsymbol{Q} \in \mathbb{R}^{N \times N}$ to be our orthogonal matrix (note that the elements of $\boldsymbol{Q}$ will also be negative). However, $\boldsymbol{Q}$ is a dense, fully-connected matrix. To enforce sparsity, we randomly zero-out $\left\lfloor \frac{P \times N}{100} \right\rfloor$ rows of $\boldsymbol{Q}$. In practice, we perform the matrix multiplication $\boldsymbol{W} = \boldsymbol{I}\boldsymbol{Q}$, where $\boldsymbol{I}$ is a $N \times N$ identity matrix with P elements of the diagonal set to 0. The elements of $\boldsymbol{I}$ to be zeroed-out are randomly selected. b) Lower triangular (Figure 6): we randomly initialize $\boldsymbol{W}$ as a lower-triangular matrix where each element is sampled uniformly in $[-1,1]$. According to the sparsity level P, and starting from the bottom, we remove a certain number of "lower" diagonals, where $j < i$ (j indexes the columns of $\boldsymbol{W}$ and i indexes its rows). A lower triangular matrix has already a base sparsity level of roughly 50%, as half of its value are zeroed-out. Further removing diagonals increases its sparsity. For example, for $\boldsymbol{W} \in \mathbb{R}^{100 \times 100}$ and $P = 90\%$, the resulting matrix will only have 10 active diagonals in the lower part (including the main diagonal). We zero-out $\left\lfloor \frac{P \times N}{100} \right\rfloor$ lower diagonals, starting from the last one (which only includes the first value of the last row of $\boldsymbol{W}$) and going up. Lower triangular matrices are associated with feed-forward architectures.

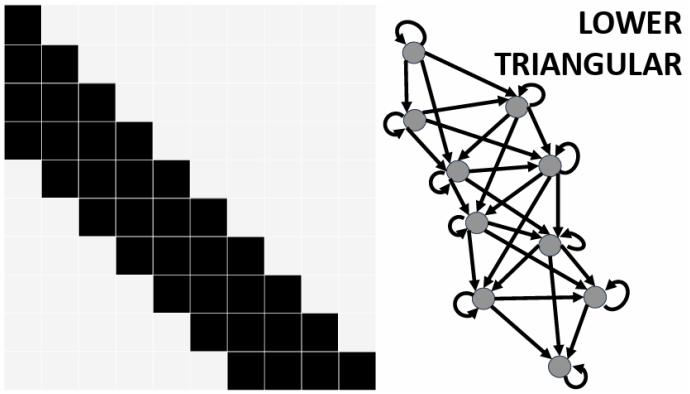


Figure 6. A 10 × 10 lower triangular matrix. In black the non-zero entries. The corresponding topology is a feed-forward architecture with the addition of self-loops corresponding to the main diagonal entries. The connections are i.i.d. uniformly sampled in (−1,1). Then the matrix is rescaled to a desired spectral radius. The sparsity level in the picture is 66%.

c) Band (Figure 7): we randomly initialize $\boldsymbol{W}$ uniformly in $[-1,1]$. Then, starting from the top and bottom and proceedings towards the main diagonal, we remove the upper $(j > i)$ and lower $(j < i)$ diagonals, up until reaching the desired level of sparsity. The resulting matrix has zero everywhere except that in a bandwidth spread around the main diagonal. The width of the bandwidth is inversely proportional to the sparsity level.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

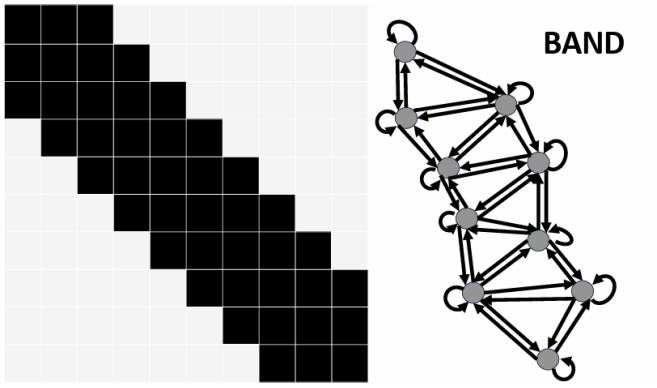


Figure 7 A 10 × 10 band matrix. In black the non-zero entries. The corresponding topology is characterised by bidirectional flow, and self loops corresponding to the main diagonal entries. The connections are i.i.d. uniformly sampled in (−1,1). Then the matrix is rescaled to a desired spectral radius. The sparsity level in the picture is 56%.

d) Toeplitz (Figure 8): we first generate a band matrix as defined above. Then, we constraint the entries on each diagonal to have the same value. Toeplitz matrices describe linear convolutional operators.

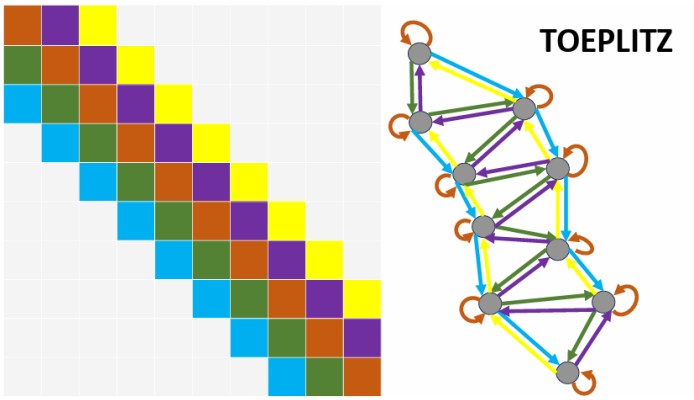


Figure 8 A 10×10 (band) Toeplitz matrix. The non-zero entries are coloured. Same colours represent same real values. In the case depicted there are effectively just 5 different real values, i.i.d. uniformly sampled in (−1,1), disposed in 5 different diagonals. Then the matrix is rescaled to a desired spectral radius. The corresponding topology is similar to the band topology with the additional constraints given by the diagonals. The sparsity level in the picture is 56%.

e) Ring (Figure 9): we create a simple cycle (or ring) matrix by filling with 1s the main sub-diagonal of $\boldsymbol{W}$ as well as the last element of the first row. The rest of the elements are zero. The simple cycle matrix has a fixed sparsity: the only nonzero elements are $N-1$ ones in the main sub-diagonal and the one in the last place of the first row, for a total of N nonzero elements. Therefore, the sparsity level is $100\frac{N^2-N}{N^2}\% = 100\left(1-\frac{1}{N}\right)\%$. As the name suggests, in a ring topology each unit is connected only with one other unit and the connections form a ring across the units.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

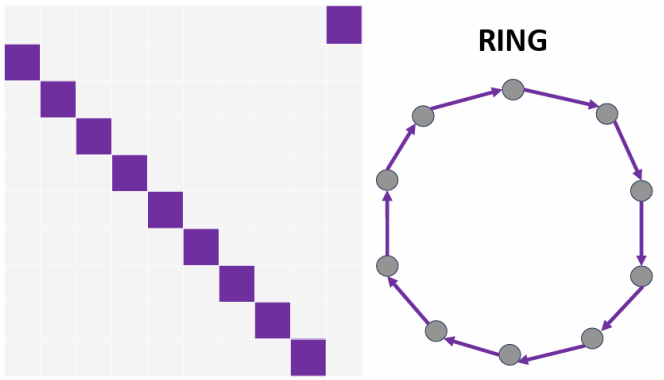


Figure 9 . A 10×10 simple ring matrix. The only non-zero entries are those in the subdiagonal and on the top right corner, and all of them are the same real value. Such unique real value is rescaled to a desired spectral radius. This connection matrix forms a ring topology. The sparsity level in the picture is 90%.

f) Circulant (Figure 10): we randomly initialise C real numbers in $[-1,1], r_1, r_2, \dots, r_C$. We define a circulant matrix with sparsity $100\left(1-\frac{C}{N}\right)\%$ by defining $(\boldsymbol{W})_{i+s,i} = r_s$, for $i = 0, \dots, N-1$, and $s = 1, \dots, C$, where the index $i+s$ is intended modulo N. Thus, the circulant structure defined in this way is a natural generalisation of the ring topology that we recover as particular case when $C = 1$. In fact, when $C = 1$, then we have $(\boldsymbol{W})_{i+1,i} = r_1$ for $i = 0, \dots, N-2$, i.e. the entire subdiagonal, and $(\boldsymbol{W})_{0,N-1} = r_1$, i.e. the top right corner. A schematic example of the sparse circulant topology is depicted in Figure 10.

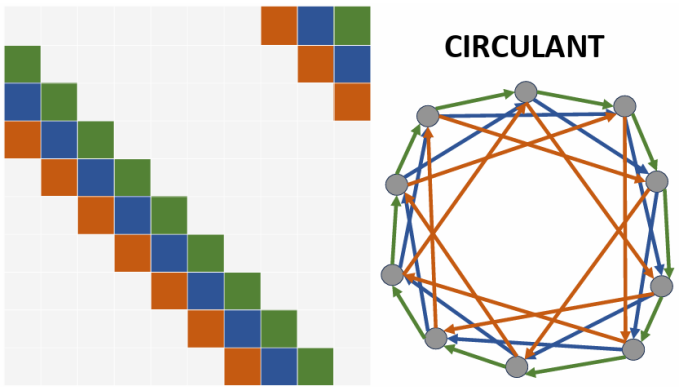


Figure 10 A 10×10 circulant matrix. The non-zero entries are coloured. Same colours represent same real values. In the case depicted there are effectively just 3 different real values, i.i.d. uniformly sampled in (−1,1). Then the matrix is rescaled to a desired spectral radius. The corresponding topology can be represented as an annular network. Note the lack of self-loops in this topology. This architecture generalises the ring topology. The sparsity level in the picture is 70%.

g) Full: for comparison, we also use a fully-connected $\boldsymbol{W}(P = 0\%)$ randomly initialized in $[-2,2]$.

We study in detail the performance of the aforementioned topologies at different levels of sparsity and for different datasets.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

3.4.1 Experiments

We chose 3 different benchmarks that are widely used for sequence processing tasks. Two of them, sequential MNIST and Adiac, are sequence classification tasks. The other, Mackey-Glass, is a chaotic time-series forecasting task.
For classification tasks, we compute the output of the models by feeding the last hidden state (i.e., the hidden state computed on the final time step of each sequence) to the linear readout layer. The readout outputs a probability for each candidate class and the class with the largest probability is selected as the final prediction. For the time-series forecasting task, the readout returns an output for each hidden state computed by the reservoir at each time step.

We found that all RON configurations surpass the performance of Leaky ESN on all three benchmarks by a significant margin, across all levels of sparsity and over different numbers of hidden units.

Table 4 Accuracy over sMNIST dataset.

Accuracy (%)	sMNIST				
Full	86.2 ± 1.5				
Ring	85.3 ± 2.5				
Sparsity	**0%**	**20%**	**40%**	**60%**	**80%**
Orthogonal	86.3 ± 2.2	82.1 ± 2.1	79.7 ± 2.0	85.4 ± 2.0	81.4 ± 1.3
Band	85.4 ± 1.3	85.5 ± 1.2	86.2 ± 0.9	$\mathbf{85.4 \pm 1.7}$	84.6 ± 0.8
Toeplitz	$\mathbf{86.6 \pm 0.9}$	81.7 ± 0.5	85.4 ± 1.6	83.0 ± 1.7	81.3 ± 3.0
Circulant	84.9 ± 1.7	$\mathbf{85.6 \pm 1.0}$	$\underline{\mathbf{86.7 \pm 0.5}}$	83.6 ± 1.2	$\mathbf{86.6 \pm 0.6}$
Leaky ESN	79.8 ± 0.1	79.6 ± 0.1	$\underline{78.3 \pm 0.1}$	77.6 ± 0.1	74.3 ± 0.1
Sparsity	**50%**	**60%**	**70%**	**80%**	**90%**
Lower Triangular	84.3 ± 1.8	83.9 ± 1.1	84.2 ± 1.5	83.6 ± 3.0	84.1 ± 1.4

We observed that RON-based reservoir computing models retain a strong performance even when the number of randomized parameters is greatly reduced. Table 4 shows the average accuracy on sMNIST across multiple levels of sparsity. Even when pruning 80% of the connections in the hidden-to-hidden matrix $\boldsymbol{W}$, both RON and Leaky ESN achieve a good performance. However, Leaky ESN shows a monotonically decreasing accuracy trend as sparsity increases. This is not the case for RON with sparse topologies. Compared to the full RON, where W is a fully-connected matrix, all the sparse variants remain competitive. In particular, circulant RON and band RON do not show any performance decrease when pruning 80% of the hidden-to hidden weights. Interestingly, circulant RON and band RON with sparsity levels greater than or equal to 40% can surpass the fully connected RON variant. The other variants (the lower triangular, the orthogonal, and the Toeplitz RON) lose a few points for high sparsity levels $(60 - 80\%)$, but they still surpass the Leaky ESN. The ring RON, a popular topology borrowed from the ESN literature, performs on par with the full RON. This confirms the effectiveness of ring-like architectures in classification problems with hundreds of time steps.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 5 Accuracy on the Adiac dataset

Accuracy (%)	Adiac				
Full	$\mathbf{72.7 \pm 1.0}$				
Ring	71.6 ± 1.2				
Sparsity	**0%**	**20%**	**40%**	**60%**	**80%**
Orthogonal	68.1 ± 3.2	70.7 ± 1.3	71.2 ± 0.6	69.9 ± 1.0	$\mathbf{71.4 \pm 1.7}$
Band	67.5 ± 2.9	69.8 ± 1.8	68.9 ± 1.6	69.3 ± 1.0	69.8 ± 2.1
Toeplitz	$\mathbf{69.9 \pm 1.4}$	70.6 ± 0.6	69.3 ± 1.0	$\mathbf{71.4 \pm 1.8}$	69.3 ± 1.4
Circulant	69.8 ± 1.8	$\mathbf{71.6 \pm 1.1}$	$\mathbf{72.2 \pm 0.9}$	70.3 ± 1.7	70.2 ± 1.1
Leaky ESN	68.5 ± 2.7	67.1 ± 1.6	67.5 ± 1.8	66.0 ± 1.1	65.8 ± 2.8
Sparsity	**50%**	**60%**	**70%**	**80%**	**90%**
Lower Triangular	71.4 ± 1.7	71.4 ± 2.0	69.2 ± 0.6	71.3 ± 1.5	68.6 ± 1.7

Adiac is a challenging dataset due to the low number of examples. However, sequences are much shorter than sMNIST. Therefore, unlike sMNIST, the model is not required to show long-term memory properties but rather an effective input-output mapping. Table 5 shows that, on Adiac, the performance of Leaky ESN slowly decreases as the sparsity level increases. RON, instead, is more robust to different sparsity levels. The only exception is the lower triangular RON, whose performance decreases from around 71% with 50% sparsity to 68% with 90% sparsity. Interestingly, on the Adiac task the Orthogonal variant with 80% sparsity almost matches the full RON. Similarly to the sMNIST classification task, the circulant RON exhibits strong performance also on Adiac, suggesting this topology as promising for classification tasks in general. Although, the full RON obtains the best score on Adiac, the Toeplitz, Orthogonal, and Circulant topologies are able to closely approach the full RON performance even with sparsity levels greater or equal than 40%.
Up to now, we focused on sequence classification datasets, where the objective is to predict the target class given the entire class. Nevertheless, recurrent models and reservoir computing models in particular, are especially powerful in time-series forecasting task. Table 6 reports the NRMSE computed on the Mackey-Glass task.

Table 6 NRMSE on Mackey-Glass

NRMSE	Mackey-Glass				
Full	0.19 ± 0.06				
Ring	0.16 ± 0.03				
Sparsity	**0%**	**20%**	**40%**	**60%**	**80%**
Orthogonal	0.16 ± 0.09	0.23 ± 0.02	0.19 ± 0.12	0.15 ± 0.04	0.15 ± 0.03
Band	0.14 ± 0.03	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.12 \pm 0.01}$	$\mathbf{0.14 \pm 0.06}$	$\mathbf{0.11 \pm 0.02}$
Toeplitz	$\mathbf{0.13 \pm 0.01}$	0.15 ± 0.02	0.13 ± 0.02	0.15 ± 0.02	0.12 ± 0.02
Circulant	0.15 ± 0.02	$\mathbf{0.12 \pm 0.01}$	0.16 ± 0.03	0.15 ± 0.01	0.13 ± 0.02
Leaky ESN	0.37 ± 0.02	0.36 ± 0.03	0.38 ± 0.01	0.38 ± 0.01	0.37 ± 0.02
Sparsity	**50%**	**60%**	**70%**	**80%**	**90%**
Lower Triangular	0.16 ± 0.01	0.16 ± 0.02	0.17 ± 0.02	0.18 ± 0.02	0.19 ± 0.01

In Mackey-Glass, the gap between RON and the Leaky ESN is even larger than in the other benchmarks. The NRMSE of RON is smaller than half of the NRMSE of Leaky ESN across all sparsity levels. Again, the lower triangular RON shows a small decrease in performance as the sparsity level increases. The other sparse topologies enjoy a robust performance with low variation between one sparsity level and another. The band RON results the clear winner on the Mackey-Glass task, almost over all sparsity levels considered. Surprisingly, the band topology with 80% sparsity achieves an NRMSE score almost halved compared to the full RON while displaying a smaller standard deviation.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

All the results presented so far may be questioned by the fact that the reservoir computing models employ a relatively low number of hidden units. In turn, increasing the sparsity percentage may result in a relatively small number of additional dropped connections in $\boldsymbol{W}$. To account for this effect, we ran additional experiments by increasing the number of hidden units in RON and Leaky ESN. Beside the experiments presented above with 100 units, we report the performance of all models with 500,$1{,}000$ and 2,000 hidden units on sMNIST (Table 7) and Mackey-Glass (Table 8). We did not run the same set of experiments on Adiac, since the main challenge of Adiac is to learn from a limited number of examples. Therefore, reservoirs with thousands of units would only over-fit to the examples in the dataset. For this family of experiments, we always set the sparsity level to a large value, namely 80%, in order to be able to properly compare all the models.

Table 7 Accuracy on sMNIST for different numbers of hidden units

sMNIST Accuracy (%)	**100**	**500**	**1,000**	**2,000**
Full	86.2 ± 1.5	93.6 ± 0.4	$\mathbf{95.5 \pm 0.3}$	96.4 ± 0.2
Ring	85.3 ± 2.5	$\mathbf{94.0 \pm 0.1}$	95.3 ± 0.2	95.6 ± 0.1
Orthogonal	81.4 ± 1.3	92.3 ± 0.3	93.5 ± 0.5	93.2 ± 0.2
Band	84.6 ± 0.8	92.7 ± 0.3	95.2 ± 0.2	$\underline{\mathbf{96.4 \pm 0.1}}$
Toeplitz	81.3 ± 3.0	91.3 ± 0.4	93.4 ± 0.2	94.3 ± 0.1
Circulant	$\mathbf{86.6 \pm 0.6}$	92.7 ± 0.3	93.7 ± 0.2	94.4 ± 0.1
Leaky ESN	74.3 ± 0.1	83.3 ± 0.1	85.8 ± 0.1	88.1 ± 0.1

On sMNIST, we see that adding more units quickly increases the final performance of all models. Leaky ESN jumps from 74% of accuracy with 100 units to 88%. It is worth observing that RON models of just 100 units can surpass Leaky ESNs of $\times 10$ number of neurons on sMNIST. Albeit circulant RON demonstrated the strongest performance on sMNIST with 100 units, when scaling up the reservoir size the band RON architecture surpasses all the other considered models. Still, all RON models greatly surpass Leaky ESN: the best-performing RONs are the full RON and the band RON, scoring over 96% on test. All the other RON variants are all above 93% test accuracy, while Leaky ESN merely touches 88%. These results show that the small amount of hidden units used in our first set of experiments did not introduce a confounding factor on the sparsity effect. Even with 20 times as many units, sparse topologies with RON can achieve a comparable performance with respect to their fully-connected version.

Table 8 NRMSE for on Mackey-Glass for different numbers of hidden units.

MG NRMSE	**100**	**500**	**1,000**	**2,000**
Full	0.19 ± 0.06	0.15 ± 0.01	0.12 ± 0.01	0.08 ± 0.01
Ring	0.16 ± 0.03	0.07 ± 0.03	$\mathbf{0.05 \pm 0.02}$	$\underline{\mathbf{0.03 \pm 0.01}}$
Orthogonal	0.15 ± 0.03	0.08 ± 0.04	0.08 ± 0.07	0.13 ± 0.12
Band	$\mathbf{0.11 \pm 0.02}$	$\mathbf{0.06 \pm 0.03}$	0.15 ± 0.11	0.09 ± 0.01
Toeplitz	0.12 ± 0.02	0.07 ± 0.02	0.08 ± 0.04	0.07 ± 0.01
Circulant	0.13 ± 0.02	0.08 ± 0.02	0.07 ± 0.01	0.08 ± 0.02
Leaky ESN	0.37 ± 0.02	0.21 ± 0.01	0.19 ± 0.01	0.15 ± 0.01

This is also confirmed on Mackey-Glass, where the NRMSE of all models is halved when switching from 100 to 2,000 hidden units. Again, all RON models surpass the Leaky ESN by a large margin. Here the gap widens since RON models of just 100 units can abundantly surpass Leaky ESNs of $\times 20$ number of neurons on Mackey-Glass. Similarly to sMNIST, also on Mackey-Glass we observe the optimal topology to change when scaling up the reservoir size. In fact, although band RON demonstrated the strongest performance on Mackey-Glass with 100 units, when increasing the number of units the ring architecture surpasses all the other considered models. Remarkably, ring RON reaches a staggering $3 \cdot 10^{-2}$ NRMSE, five

European Innovation Council

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

times smaller than a Leaky ESN with the same number of 2000 units. Therefore, we can conclude that increasing the number of hidden units in RON causes a corresponding increase in the performance, preserving both the robustness against large sparsity levels and the advantage with respect to the Leaky ESN.

Our experiments suggest the circulant (including ring) and the band as the most promising topologies. Moreover, it emerges that the specific topology can significantly improve the performance while massively dropping the number of possible connections (even more than 99.9%), and that, in general, large sparsity levels are well tolerated by the RON model. This is a positive result indicating the possibility to physically implement RON, where the oscillators are coupled just with their neighbours, and having the chance to outperform fully connected digital alternatives. Another interesting aspect emerged is that for a predetermined sparsity level the optimal topology might depend on the reservoir size.

The opportunity to leverage on sparse topologies in oscillatory-based recurrent models like RON opens up the possibility of implementation of such networks in real-world applications. One of the common problems when considering neuromorphic / physical implementation of neural networks is that all units in a layer are fully connected to all units in the next layer. Recurrent neural networks with a single layer suffer from the same issue since the units within the same layer are fully-connected.
Developing sparse topologies allows to simplify the physical implementation of the model and allows for more flexibility in its development. Ad-hoc sparse connections can follow the constraints coming from the hardware substrate. For oscillatory-based models, it is unlikely to build a fully connected network of thousands of oscillators outside a computer simulation. With our work, we showed that: 1) oscillators-based models like RON do not always need thousands of units, but they can work well even with only 100 hidden units; 2) RON can prune the vast majority of the recurrent connections without affecting its predictive performance. Since RON follows the reservoir computing paradigm, we do not need to train the recurrent connections after the model is realised on a hardware substrate. This also gets rid of the on-device training issue, which still plagues all fully trainable models like the LSTM when it comes to their physical implementation.
Overall, RON looks like a compelling candidate for a physical implementation, due to both its computational efficiency (no training required on the recurrent component) and its effectiveness when equipped with very sparse topologies.

3.5. Physically-implementable Random Oscillators Networks (pRONs)

We provide a version of RON that is physically implementable and fully compatible with the archetype systems of D3.1.Due to its architectural properties (e.g., two separate nonlinear terms instead of only one as in RON) we are able to provide strong stability results for this new Archetype network.

We consider the continuous-time network dynamics

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\mathbf{y}} \\ \dot{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \tanh(\mathbf{V}\mathbf{u}(t)) - \mathbf{W}^T \tanh(\mathbf{W}\,\mathbf{y} + \mathbf{b}) - \boldsymbol{\Gamma}\,\mathbf{y} - \mathbf{E}\,\mathbf{z} \end{bmatrix},$$

where $\mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \in \mathbb{R}^{2N}$ represents the network's state, $\mathbf{u}(t) \in \mathbb{R}^I$ is the time-dependent network input, $\boldsymbol{\Gamma} = \mathrm{diag}(\gamma_1, \dots, \gamma_i, \dots, \gamma_N) \succ 0$ and $\mathbf{E} = \mathrm{diag}(\varepsilon_1, \dots, \varepsilon_i, \dots, \varepsilon_N) \succ 0$ describe the stiffness and damping coefficients of N harmonic oscillators, respectively. We assume that $\mathbf{W} \in \mathbb{R}^{N \times N}$ is *both full-rank and orthogonal* ($\mathbf{W}\mathbf{W}^{\mathbf{T}} = \mathbb{I}^N$). Finally, $\mathbf{b} \in \mathbb{R}^N$ is the bias term of the neuron-inspired potential force.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

As visualized in Figure 11, these dynamics can be interpreted as N harmonic oscillators of unit mass coupled by the neuron-inspired potential $U(\mathbf{y}) = \sum_{i=1}^{N} \log(\cosh(\sum_{j=1}^{N} \mathbf{W}_{ij}\mathbf{y}_j + \mathbf{b}_i))$ and excited through an input neuron that uses a hyperbolic tangent as the activation function. The output of the network $\boldsymbol{r}(t) \in \mathbb{R}^O$ is given by applying a linear layer to the oscillator positions: $\boldsymbol{r}(t) = \mathbf{W_o}\mathbf{y}(t) + \mathbf{b}_o$.

The system exhibits a unique, isolated equilibrium $\bar{\mathbf{x}} = \begin{bmatrix} \bar{\boldsymbol{y}} \\ \mathbf{0}^N \end{bmatrix}$ that is given by the roots of the equation $\mathbf{W}^T \tanh(\mathbf{W}\bar{\boldsymbol{y}} + \mathbf{b}) + \boldsymbol{\Gamma}\,\bar{\boldsymbol{y}} = 0$.

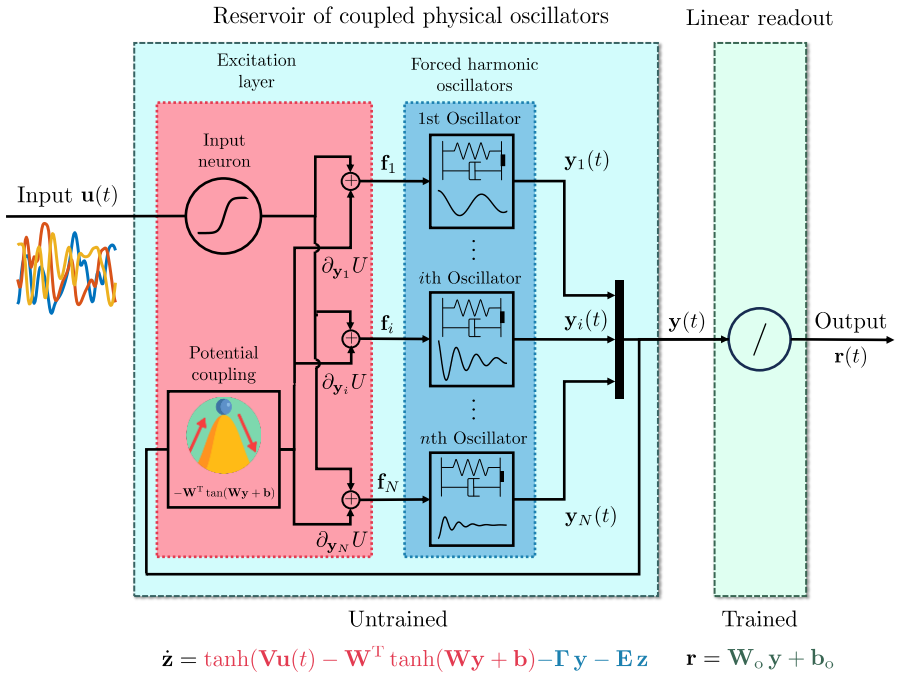


*Figure 11. Block diagram of the physically-implementable Randomly Coupled Oscillator Network (pRCON): we define a reservoir of N harmonic oscillators coupled by a neuron-inspired potential. An input neuron excites the network as a function of **u**(t). The recorded oscillator positions **y**(t) serve as an input to the trainable linear readout layer.*

We can define a transformation $\mathbf{y}_{\mathrm{w}} = \mathbf{W}\mathbf{y} \in \mathbb{R}^N, \mathbf{z}_{\mathrm{w}} = \mathbf{W}\mathbf{z} \in \mathbb{R}^N$ into $\mathcal{W}$-coordinates in which the state-space dynamics becomes

$$\dot{\boldsymbol{x}}_{\mathrm{w}} = \begin{bmatrix} \dot{\boldsymbol{y}}_{\mathrm{w}} \\ \dot{\mathbf{z}}_{\mathrm{w}} \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{\mathrm{w}} \\ \mathbf{W}^T \tanh(\mathbf{V}\mathbf{u}(\mathrm{t})) - \tanh(\mathbf{y}_{\mathrm{w}} + \mathbf{b}) - \boldsymbol{\Gamma}_{\mathrm{w}}\,\mathbf{y}_{\mathrm{w}} - \mathbf{E}_{\mathrm{w}}\,\mathbf{z}_{\mathrm{w}} \end{bmatrix}.$$

Here, $\boldsymbol{\Gamma}_{\mathrm{w}} = \mathbf{W}\,\boldsymbol{\Gamma}\,\mathbf{W}^{\mathrm{T}} \succ 0$ and $\mathbf{E}_{\mathrm{w}} = \mathbf{W}\,\mathbf{E}\,\mathbf{W}^{\mathrm{T}} \succ 0$ are now full matrices representing the stiffness and the damping coefficients, respectively. $\boldsymbol{r}(t) = \mathbf{W_o}\mathbf{y}_{\mathrm{w}}(\mathbf{t}) + \mathbf{b}_{\mathrm{o}}$ recovers the output of the network.

Furthermore, we can now, in the $\mathcal{W}$-coordinates, come up with a physical interpretation of the network topology: the network consists of N oscillators, each with the nonlinear elastic force profile $\tanh(\mathbf{y}_{\mathrm{w},i} + \mathbf{b}_i)$, where i denotes the index of the oscillator unit. Each oscillator additionally exhibits linear elasticity $\boldsymbol{\Gamma}_{\mathrm{w},ii}\,\mathbf{y}_{\mathrm{w},i}$ and damping $\mathbf{E}_{\mathrm{w},ii}\,\mathbf{z}_{\mathrm{w},i}$. A spring damper with stiffness $\boldsymbol{\Gamma}_{\mathrm{w},ij}$ and damping coefficient $\mathbf{E}_{\mathrm{w},ij}$ connects the *i*th and the *j*th oscillator units. Motors apply the force $\mathbf{f}_w = \mathbf{W}^T \tanh(\mathbf{V}\mathbf{u}(\mathrm{t})) \in \mathbb{R}^N$ on the oscillator units.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

We strive to show that the network exhibits bounded states $\mathbf{x}(\mathbf{t})$ for bounded inputs $\mathbf{u}(\mathbf{t})$**.** This can be achieved by proving Input-to-State (ISS) stability [72]. First, we introduce a change of coordinate w.r.t. the equilibrium state: $\tilde{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$**.** The delta dynamics are then given by

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \dot{\tilde{\mathbf{y}}} \\ \dot{\tilde{\mathbf{z}}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{z}} \\ \tanh(\mathbf{V}\mathbf{u}(\mathrm{t})) - \mathbf{W}^T \tanh(\mathbf{W}\bar{\mathbf{y}} + \mathbf{W}\tilde{\mathbf{y}} + \mathbf{b}) - \boldsymbol{\Gamma}(\bar{\mathbf{y}} + \tilde{\mathbf{y}}) - \mathbf{E}\tilde{\mathbf{z}} \end{bmatrix},$$

with the equilibrium at $\tilde{\mathbf{x}} = \mathbf{0}^{2N}$**.**

We now consider the strict Lyapunov candidate with skewed level sets [15], [73] $V_\mu(\tilde{\mathbf{x}}) = \frac{1}{2}\tilde{\mathbf{x}}^T \mathbf{P}_\mathrm{V}\, \tilde{\mathbf{x}} + V_{\tanh}(\tilde{\mathbf{y}})$ where lcosh(·)=log(cosh(·)), $\mathbf{P}_\mathrm{V} = \begin{bmatrix} \boldsymbol{\Gamma} & \mu\mathbb{I}^N \\ \mu\mathbb{I}^N & \mathbb{I}^N \end{bmatrix}$, $\mu > 0$ and

$$V_{\tanh}(\tilde{\mathbf{y}}) = \sum_{i=1}^{N} \left[\mathrm{lcosh}\left(\sum_{j=1}^{N} \mathbf{W}_{ij}\, \bar{\mathbf{y}}_j + \sum_{j=1}^{N} \mathbf{W}_{ij}\, \tilde{\mathbf{y}}_j + \mathbf{b}_i \right) - \mathrm{lcosh}\left(\sum_{j=1}^{N} \mathbf{W}_{ij}\, \bar{\mathbf{y}}_j + \mathbf{b}_i \right) - \tanh\left(\sum_{j=1}^{N} \mathbf{W}_{ij}\, \bar{\mathbf{y}}_j + \mathbf{b}_i \right) \sum_{j=1}^{N} \mathbf{W}_{ij}\, \tilde{\mathbf{y}}_j \right].$$

$V_\mu(\tilde{\mathbf{x}})$ is bounded by the $\mathcal{K}_\infty$ functions $\alpha_1(r)$ and $\alpha_2(r)$

$$\alpha_1(\|\tilde{\mathbf{x}}\|_2) \leq V_\mu(\tilde{\mathbf{x}}) \leq \alpha_2(\|\tilde{\mathbf{x}}\|_2),$$

$$\frac{1}{2}\lambda_\mathrm{m}(\mathbf{P}_\mathrm{V})\|\tilde{\mathbf{x}}\|_2^2 \leq V_\mu(\tilde{\mathbf{x}}) \leq \frac{1}{2}\lambda_\mathrm{M}(\mathbf{P}_\mathrm{V})\|\tilde{\mathbf{x}}\|_2^2 + 2\sqrt{N}\,\|\mathbf{W}\|_1\|\tilde{\mathbf{x}}\|_2,$$

where $\lambda_\mathrm{m}(\mathbf{A})$ and $\lambda_\mathrm{M}(\mathbf{A})$ are the smallest and largest Eigenvalues of the matrix $\mathbf{A}$**,** respectively. Then, the network is ISS stable such that

$$\|\tilde{\mathbf{x}}\|_2 \leq \beta(\|\tilde{\mathbf{x}}(t_0)\|_2, t - t_0) + \gamma\left(\sup_{t \geq t_0} \|\tanh(\mathbf{V}\mathbf{u}(\mathrm{t}))\|_2\right),$$

with $\beta(r, s)$ a class $\mathcal{KL}$ function, and $\gamma(r) \in \mathcal{K}$ given by

$$\gamma(r) = \sqrt{\frac{(1+\mu^2)\,\lambda_\mathrm{M}(\mathbf{P}_\mathrm{V})\, r^2 + 4\sqrt{N}\,\|\mathbf{W}\|_1 \sqrt{1+\mu^2}\,\lambda_\mathrm{m}(\mathbf{P}_{\dot{V}})\, r}{\theta^2 \lambda_\mathrm{m}(\mathbf{P}_\mathrm{V})\lambda_\mathrm{m}(\mathbf{P}_{\dot{V}})^2}}.$$

Here, $0 < \theta < 0$ ensures stability and $\mathbf{P}_{\dot{V}} = \begin{bmatrix} \mu\boldsymbol{\Gamma} & \frac{1}{2}\mu\,\mathbf{E} \\ \frac{1}{2}\mu\,\mathbf{E}^\mathrm{T} & \mathbf{E} - \mu\,\mathbb{I}^N \end{bmatrix}$. Therefore, if the network is unactuated with $\sup_{t \geq t_0} \|\mathbf{u}(\mathrm{t})\|_2 = \mathbf{0}$, the system is globally asymptotically stable and converge to the equilibrium $\bar{\mathbf{x}} = \begin{bmatrix} \bar{\mathbf{y}} \\ \mathbf{0}^N \end{bmatrix}$. Otherwise, the deviation of the system state $\tilde{\mathbf{x}}$ from the equilibrium will remain proportionally bounded w.r.t. $\gamma\left(\sup_{t \geq t_0} \|\tanh(\mathbf{V}\mathbf{u}(\mathrm{t}))\|_2\right)$.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

4. Preliminary results on Archetype Networks

In the following we report our preliminary results on Archetype Networks based on first order linear ODE units, therefore of the kind $h' + a\,h = u,$ with u being the neuron-like connector as defined in Section 2.2 of Deliverable D3.1 or slight variations of it that modify its connectivity structure. In Section 4.1 we consider a neuron-like connector with a skew-symmetric structure inspired by Neural ODEs (Euler State Networks). In Section 4.2 we explore neuron-like connectors augmented with residual connections (Residual Recurrent Neural Networks). Finally in Section 4.3 we enforce in the neuron-like connector a connectivity pattern inspired by the brain (Continuously Deep Recurrent Neural Networks). We frame all these three models again in the RC framework to simplify the analysis.

4.1. Euler State Networks

As reported in Section 1.1 of Deliverable D3.1, an archetypical unit is an ODE. Here we propose an Archetype Network where a single unit is of the kind $h' = u$ where u is the neuron-like connector defined in section 2.2 of Deliverable D3.1. In formula, we have in compact form the following ODE describing both units and connectors:

$$\begin{aligned}\mathbf{h}'(t) &= \boldsymbol{f}\big(\mathbf{h}(t), \mathbf{u}(t)\big) \\ &= \tanh\left(\mathbf{W}_{\boldsymbol{h}}\mathbf{h}(t) + \mathbf{W}_{\mathbf{u}}\mathbf{u}(t) + \mathbf{b}\right)\end{aligned}$$

To implement the continuous-time system above we choose the Euler discretisation scheme. We call the resulting architecture the Euler State Network (EuSN). We aim to design reservoir dynamics that are both stable and non-dissipative by construction, so to provide an Archetype Network biased to long-term propagation of information. The key mathematical insight is to parametrise the recurrent matrix as follows:

$$\mathbf{W}_h = \mathbf{S}_h - \gamma\mathbf{I},$$

where $\mathbf{S}_h$ is a skew-symmetric matrix, $\mathbf{I}$ the identity, and γ a diffusion term, so that the EuSN state update dynamics (discretised with time step ε) reads

$$\begin{aligned}\mathbf{h}(t) = &\ \boldsymbol{F}(\mathbf{h}(t-1), \mathbf{u}(t)) \\ = &\ \mathbf{h}(t-1) + \\ &\ \varepsilon\tanh\left((\mathbf{S}_h - \gamma\mathbf{I})\mathbf{h}(t-1) + \mathbf{W}_{\mathbf{u}}\mathbf{u}(t) + \mathbf{b}\right).\end{aligned}$$

We refer to [74] for the details of the mathematical derivation. Experiments on long-term memory tasks show the clear superiority of the proposed approach over standard RC models in problems requiring effective propagation of input information over multiple time steps. In the next section, we briefly report an experiment showing the effectiveness of EuSN in long-term memorization.

Long-term memorization

We test EuSN on a set of time-series classification tasks to assess the long-term memorization (LTM) capabilities of recurrent layers, basically padding with noise time series to make more challenging the classification. The general aim is to exercise the ability of neural networks to correctly classify an increasingly long input time-series based on the presence of specific

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

patterns injected into the sequence at arbitrary points in time. The recurrent layer must be able to effectively latch input information into the state representations over long time spans, as the information relevant to the target becomes increasingly distant from the input suffix, posing a relevant challenge for RC-based systems.

In the following we report the results of 4 tasks. We refer to [74] for the details of the experimental setting.

Results

The results on the LTM tasks are reported in Figure 12, which shows the test set accuracy achieved by EuSN, ESN and and R-ESNs [75], for increasing values of the total length of padded sequences. Each plot in Figure 12 corresponds to one of the LTM tasks considered in this study. As it is evident, the performance shown by EuSN is consistently better than that of ESN and R-ESN. In general, it can be observed that the level of accuracy obtained by all models decreases as T increases. Crucially, the accuracy of EuSN decreases slowly and remains high even for high values of T, indicating an effective propagation process of taskrelevant information even after hundreds of time-steps. In contrast, consistent with its fading memory characterization, ESN (and R-ESN) shows a rapidly degrading performance. For example, in the case of the Synthetic task, ESN's accuracy is close to the chance level (i.e., ≈ 0.5) after a few hundred time-steps.

Overall, results in Figure 12 indicate that EuSN can propagate the input information effectively in tasks requiring long-term memorization abilities, overcoming the fading memory limitations that are typical of ESN variants.

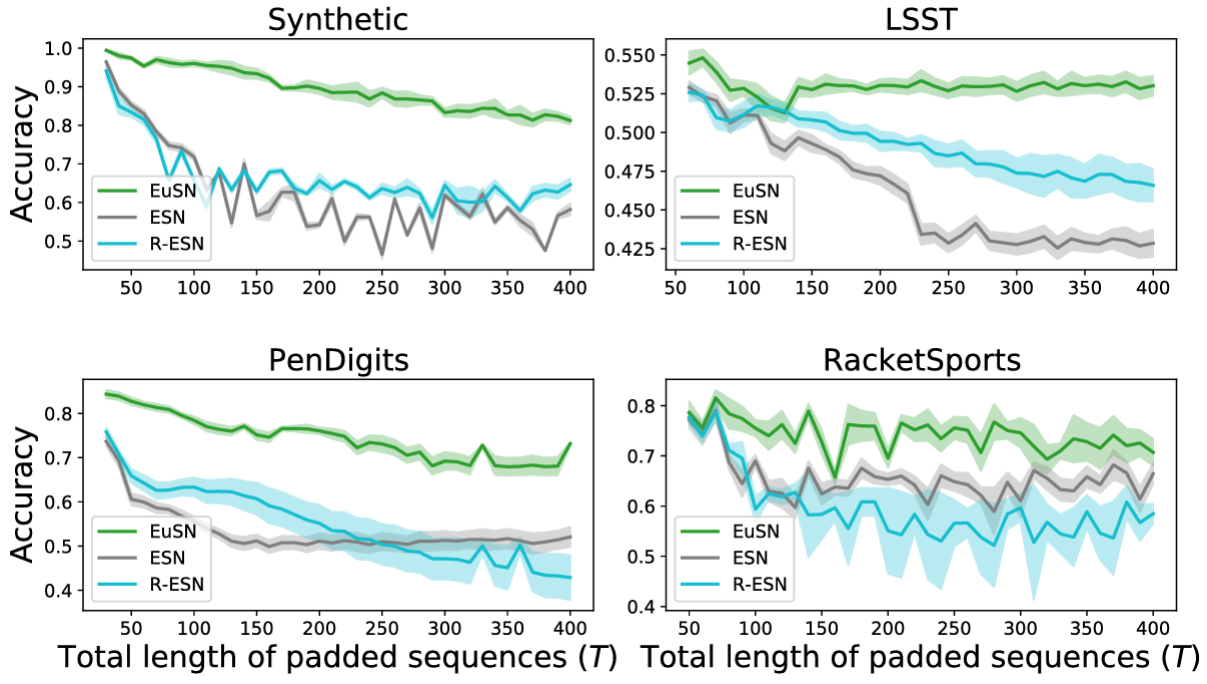


Figure 12. Accuracy achieved on the LTM tasks by EuSN for increasing the length of the total length of padded sequences T (higher is better). Results are compared to standard ESN and R-ESN. For each value of T, the plots

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

report the accuracy values averaged over the 10 random instances and the corresponding standard deviation as a shaded area.

4.2. Residual Recurrent Neural Networks

Similarly to the Archetype Network of the previous section (Euler State Networks), here we explore an Archetype Network based on linear units of the kind $h' + ah = u$, where u is the neuron-like connector augmented with residual connections. Residual networks (ResNets) are a type of deep neural network that uses skip connections [43]. Skip connections allow the network to learn the residual mapping between the input and output of a layer, instead of the direct mapping. This makes the network easier to optimise and enables the training of very deep architectures. Despite their success, little has been explored in the context of RNNs. We investigate the architectural bias of residual connections in the temporal dimension of RNNs, considering the simplest case of a single layer residually connected with itself in the temporal dimension at the previous time step. We address the problem of effective information propagation considering skip connections that linearly propagate the network state to the next time step (an idea also supported by biological plausibility [76]). To maximise the information content of the time-propagated state memory, the residual recurrent connections are modulated by an untrained orthogonal weight matrix, thus exploiting, in an RNN context, the optimal memory properties of this type of dynamic neural systems.

Precisely, we introduce a class of RC models based on linear skip connections in the state processing and called Residual Echo State Network (ResESN) [5].

The state transition function of the residual network is given as follows:

$$\mathbf{h}(t) = \alpha \mathbf{O}\mathbf{h}(t-1) + \beta \tanh(\mathbf{W_h}\mathbf{h}(t-1) + \mathbf{W_u}\mathbf{u}(t) + \mathbf{b}). \tag{13}$$

where $\mathbf{O}$ is a randomly generated orthogonal matrix, and α, β are scaling coefficients that we treat as hyperparameters. The matrix $\mathbf{O}$ can be generated by taking the orthogonal matrix resulting from the QR decomposition of a random matrix $\mathbf{M}$ of dimension $N_h \times N_h$ with i.i.d. entries in (-1,1). The values of α and β can be used to adjust the dynamic behaviour of the reservoir. All other terms in eq. (13) above are the same as in eq. (1) of the standard Leaky ESN model. Notably, the proposed model inherits the advantage of being fast to train from the RC paradigm.

As already stated above, eq. (13) of a ResESN can be interpreted as a generalisation of the neuron-like connector of EuSN, with the additional freedom of tuning the trade-off between linearity and nonlinearity via the two hyperparameters α and β. An alternative interpretation is to view the ResESN as a linear Archetype Network of continuous-time equations:

$$\mathbf{h}'(t) = \mathbf{A}\mathbf{h}(t) + \mathbf{U},$$

where

$$\mathbf{A} = (\alpha \mathbf{O} - \mathbf{I})/\beta,$$

with $\mathbf{O}$ an orthogonal matrix and $\mathbf{I}$ the identity, and $\mathbf{U}$ a neuron-like connector of the standard following form

$$\mathbf{U} = \tanh(\mathbf{W_h}\mathbf{h}(t-1) + \mathbf{W_u}\mathbf{u}(t) + \mathbf{b}),$$

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

where $\mathbf{u}(t)$ is the external input. This ODE, when discretised with the Euler scheme with time step β, brings to eq. (13).

Intuitively, considering the application of the recurrent layer over time, the introduction of the skip connections in the additive part of Eq. (13) (first term on the right-hand side) allows the creation of a path for the long-term propagation of the input information. This concept is illustrated graphically in Figure 13 below.

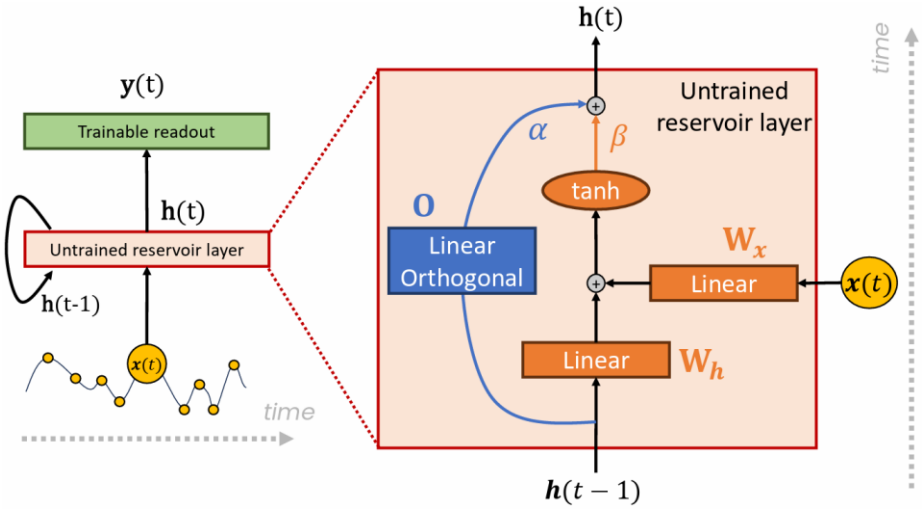


Figure 13. ***Left:*** *representation of a single layer reservoir computing architecture. In orange is the recurrent computational cell, and in green is the trainable readout.* ***Right:*** *scheme of the recurrent computational cell of a ResESN. For simplicity, the bias vectors are not shown in the figure.*

Here we just report a summary of the theoretical results that we derived for ResESN. We notice that due to the orthogonal structure of $\mathbf{O}$, the eigenvalues of the $\alpha\mathbf{O}$ term of eq. (13) are distributed in the complex plane over a circle of radius α centered at the origin. Therefore, we can locate the position of the eigenvalues of the complete Jacobian as being at maximum distance $\beta\|\mathbf{W_h}\|$ from those of $\alpha\mathbf{O}$, i.e., within a $\beta\|\mathbf{W_h}\|$-tube around the circle of radius α. This remark allows for a precise estimation of the ResESN spectrum, which can be made as close as we want to the edge of stability represented by the unitary circle, see Figure 14 below.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

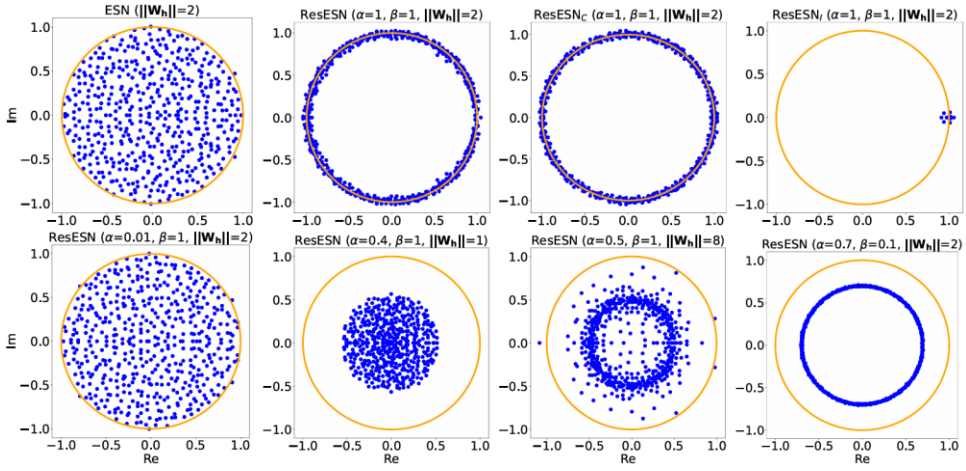


Figure 14. Eigenvalues of the resulting Jacobian in reservoirs with $N_h = 500$ recurrent neurons driven by a random uniform input vector in (-1,1) for various hyperparameter's values $\alpha, \beta, \|\boldsymbol{W_h}\|$, and assuming zero bias $\omega_b = 0$, and $\omega_x = 1$. In orange the unitary circle.

The hyperparameter α, in this context, makes it possible to decide how close to be to this boundary, with a value of $\alpha = 1$ yielding the boundary condition, see middle plots of the first row of Figure 14. At the same time, the hyperparameter β, together with the magnitude of the weights of $\mathbf{W_h}$, determines perturbations to the eigenvalues of $\alpha\mathbf{O}$, allowing a wider range of dynamics to be covered. In the Figure 14 above ResESN_C represents a ResESN model with a (deterministc) simple cycle orthogonal matrix in place of a random orthogonal matrix, while ResESN_I is a ResESN with the identity matrix as orthogonal matrix.

Experiments

We test our proposed approach on few classification and regression benchmark tasks covering both synthetic and real-world datasets.

Memorisation capability

Here we measure how effectively the models can recall past inputs and exploit them for computational purposes. First we compute the memory capacity of the models (MemCap task), evaluating the ability to reconstruct a delayed random signal. Then we test the models on 2 more challenging tasks. The former requires reconstructing a sinusoidal transformation of the delayed input. Thus we call it SinMem task. The latter can be interpreted as a continuous extension of a XOR task with delayed inputs. Thus we call it CtXOR task.

Discussion. As evident from Table 9 below, in all three memory-based tasks considered, the architectural bias induced by random orthogonal and simple cycle variants of ResESN gives a substantial advantage. The MemCap task reveals the outstanding capacity of memory retention of the ResESN variants with a random orthogonal matrix, and even more the simple cycle matrix. Note that, for a reservoir of 100 neurons the memory capacity as defined in the MemCap task has been proved [77] to be at most 100, hence these models are extremely close to the theoretical maximum. Finally, ResESN outperforms standard ESN variants by 2 orders of magnitudes on the SinMem task.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 9. Results on the memorisation capability tasks. Mean and standard deviation over 10 different initialisations of RC models of 100 neurons. The arrows on the left inform whether larger (up) or smaller (down) values indicate better performance for each task. For each task, the best overall is underlined, while the top two results are highlighted in blue.

Memory-based	LeakyESN	LeakySCR	ES^{2}N	EuSN	ResESN	ResESN$_C$	ResESN$_I$
↑ **MemCap**	$24.1_{\pm 1.5}$	$31.4_{\pm 0.8}$	$97.7_{\pm 0.3}$	$19.3_{\pm 0.7}$	$97.8_{\pm 0.8}$	$99.0_{\pm 0.0}$	$24.4_{\pm 3.2}$
↓ **SinMem** $(\cdot 10^{-2})$	$35.7_{\pm 0.6}$	$35.2_{\pm 0.3}$	$13.7_{\pm 1.1}$	$80.9_{\pm 6.9}$	$0.2_{\pm 0.1}$	$0.2_{\pm 0.1}$	$35.8_{\pm 0.7}$
↓ **CtXOR** $(\cdot 10^{-1})$	$4.3_{\pm 0.3}$	$3.9_{\pm 0.1}$	$4.0_{\pm 0.2}$	$11.8_{\pm 1.9}$	$3.8_{\pm 0.1}$	$3.8_{\pm 0.1}$	$4.3_{\pm 0.3}$

Time series classification

We experimentally validate the proposed residual RC models on a set of diverse time series classification benchmarks from the UEA & UCR repository (www.timeseriesclassification.com) plus the popular sequential MNIST (sMNIST) classification task.

We perform the classification of time series by exploiting only the last reservoir state as input for the readout classifier. The results given in Table 10 show the achieved accuracy scores, averaged over the random guesses with their empirical standard deviations. In addition to the accuracy scores on the individual tasks, we provide the resulting average rankings of the assessed models on the whole set of classification tasks in the histograms in Figure 15. For each of the 17 classification tasks we rank all the seven RC models assigning 1 point to the model achieving the larger test accuracy, 2 points to the second, and so on up to the seventh. Then the resulting scores are divided by 17. The lower the better.

Discussion. The values shown in the Table 10 below highlight the striking advantage of the class of models proposed here over traditional RC. Moreover, the accuracy achieved by the models in the residual class is substantially similar to the level achieved by EuSN. In this context, note that the ResESNs (and variants) perform equally well in the memorisation, where EuSNs perform significantly worse. From the average rankings of Figure 15, we conclude that the residual models together with EuSN form a distinctive clique of best performing models, posing them at a superior level for classification scopes with respect to other RC approaches.

Table 10. Accuracy results achieved on the time series classification tasks (the higher the better in all cases). The table reports the mean accuracy and standard deviation over 10 different initialisations of reservoir models. For each task, the best overall is underlined, while the top two results are highlighted in blue.

Task	LeakyESN	LeakySCR	ES^{2}N	EuSN	ResESN	ResESN$_C$	ResESN$_I$
Adiac	$0.299_{\pm 0.021}$	$0.480_{\pm 0.038}$	$0.289_{\pm 0.027}$	$0.703_{\pm 0.015}$	$0.629_{\pm 0.017}$	$0.630_{\pm 0.018}$	$0.497_{\pm 0.015}$
Blink	$0.513_{\pm 0.028}$	$0.537_{\pm 0.022}$	$0.484_{\pm 0.054}$	$0.800_{\pm 0.031}$	$0.564_{\pm 0.032}$	$0.695_{\pm 0.073}$	$0.763_{\pm 0.031}$
CinCECG	$0.283_{\pm 0.001}$	$0.607_{\pm 0.044}$	$0.248_{\pm 0.000}$	$0.724_{\pm 0.009}$	$0.558_{\pm 0.068}$	$0.708_{\pm 0.042}$	$0.617_{\pm 0.039}$
ECG5000	$0.922_{\pm 0.001}$	$0.939_{\pm 0.002}$	$0.932_{\pm 0.005}$	$0.936_{\pm 0.002}$	$0.926_{\pm 0.004}$	$0.940_{\pm 0.001}$	$0.933_{\pm 0.002}$
FordA	$0.639_{\pm 0.015}$	$0.664_{\pm 0.021}$	$0.600_{\pm 0.013}$	$0.704_{\pm 0.011}$	$0.664_{\pm 0.031}$	$0.726_{\pm 0.029}$	$0.661_{\pm 0.017}$
FordB	$0.598_{\pm 0.016}$	$0.605_{\pm 0.010}$	$0.560_{\pm 0.012}$	$0.614_{\pm 0.005}$	$0.601_{\pm 0.033}$	$0.571_{\pm 0.025}$	$0.633_{\pm 0.014}$
Kepler	$0.335_{\pm 0.014}$	$0.376_{\pm 0.027}$	$0.348_{\pm 0.014}$	$0.450_{\pm 0.036}$	$0.522_{\pm 0.015}$	$0.518_{\pm 0.013}$	$0.447_{\pm 0.021}$
Libras	$0.611_{\pm 0.025}$	$0.656_{\pm 0.035}$	$0.524_{\pm 0.040}$	$0.771_{\pm 0.012}$	$0.806_{\pm 0.010}$	$0.717_{\pm 0.017}$	$0.692_{\pm 0.033}$
Lightning2	$0.623_{\pm 0.000}$	$0.623_{\pm 0.000}$	$0.620_{\pm 0.012}$	$0.600_{\pm 0.082}$	$0.602_{\pm 0.010}$	$0.639_{\pm 0.018}$	$0.618_{\pm 0.042}$
Mallat	$0.492_{\pm 0.068}$	$0.518_{\pm 0.007}$	$0.213_{\pm 0.055}$	$0.808_{\pm 0.005}$	$0.888_{\pm 0.020}$	$0.870_{\pm 0.046}$	$0.714_{\pm 0.028}$
OliveOil	$0.400_{\pm 0.000}$	$0.427_{\pm 0.042}$	$0.400_{\pm 0.000}$	$0.620_{\pm 0.078}$	$0.773_{\pm 0.036}$	$0.860_{\pm 0.029}$	$0.733_{\pm 0.077}$
ShapesAll	$0.543_{\pm 0.022}$	$0.622_{\pm 0.016}$	$0.353_{\pm 0.021}$	$0.764_{\pm 0.008}$	$0.715_{\pm 0.015}$	$0.692_{\pm 0.011}$	$0.723_{\pm 0.009}$
sMNIST	$0.438_{\pm 0.029}$	$0.710_{\pm 0.033}$	$0.819_{\pm 0.013}$	$0.835_{\pm 0.006}$	$0.820_{\pm 0.007}$	$0.891_{\pm 0.006}$	$0.810_{\pm 0.032}$
StarLight	$0.866_{\pm 0.003}$	$0.872_{\pm 0.007}$	$0.827_{\pm 0.011}$	$0.955_{\pm 0.002}$	$0.925_{\pm 0.006}$	$0.916_{\pm 0.015}$	$0.945_{\pm 0.002}$
UWave	$0.770_{\pm 0.006}$	$0.834_{\pm 0.028}$	$0.487_{\pm 0.023}$	$0.947_{\pm 0.001}$	$0.871_{\pm 0.012}$	$0.883_{\pm 0.016}$	$0.933_{\pm 0.002}$
Wafer	$0.989_{\pm 0.002}$	$0.992_{\pm 0.000}$	$0.942_{\pm 0.027}$	$0.989_{\pm 0.001}$	$0.994_{\pm 0.001}$	$0.989_{\pm 0.003}$	$0.993_{\pm 0.001}$
Yoga	$0.678_{\pm 0.005}$	$0.763_{\pm 0.016}$	$0.677_{\pm 0.006}$	$0.783_{\pm 0.018}$	$0.732_{\pm 0.027}$	$0.783_{\pm 0.015}$	$0.807_{\pm 0.015}$

European Innovation Council

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

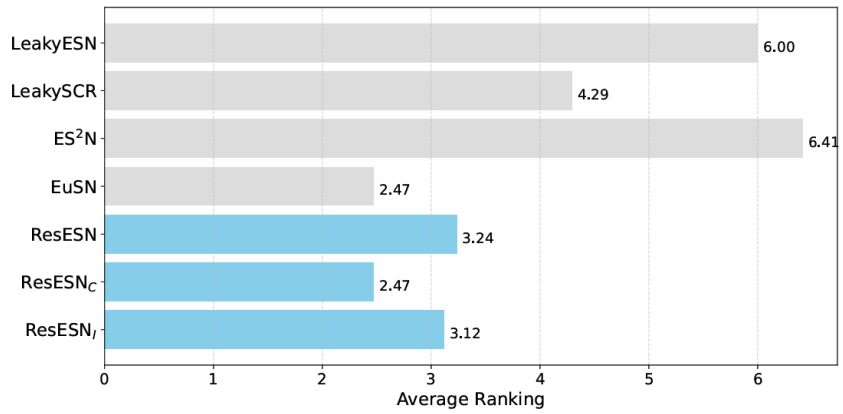


Figure 15. Average ranking across time series classification tasks (the lower the better). Our proposed residual models are highlighted in blue.

4.3. Continuously Deep Recurrent Neural Networks

In this preliminary work (preliminarily described in [6]) we explored further the role of the neuron-like connector promoting a spatial locality pattern of connectivity in a recurrent network. In fact, unlike feedforward neural networks, RNNs spatially encode temporal features of the input in their hidden state allowing for context-dependent computation, a key feature of prefrontal cortex microcircuits. Cortical networks in the brain exhibit hierarchies of temporal processing likely due to the structural organization of the cortex, which is dominated by neural connection probability diminishing exponentially with distance. This motivates the study of Archetype Networks whose connectivity topology is similarly constrained by such exponential distance rules.

We introduce a novel class of RC-based architectures called Continuously-Deep ESN (CDESN). Our proposal allows the design of biologically plausible ESNs characterized by local connections among neurons based on an exponentially decaying rule while modulating the depth of the internal information processing via a single hyperparameter of the C-DESN model. We show in the following section that we can interpret this local connectivity as the application of a particular connector on the archetypical RNN units. Unlike the deep architectures previously explored in the RC literature, as DeepESN [78], and fundamentally based on the concept of a pool of "discrete" recurrent layers nested one inside the other, the recurrent architecture of the model proposed here presents a more nuanced and "continuous" concept of layering. A graphical representation of CDESN architecture is shown in Figure 16.

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

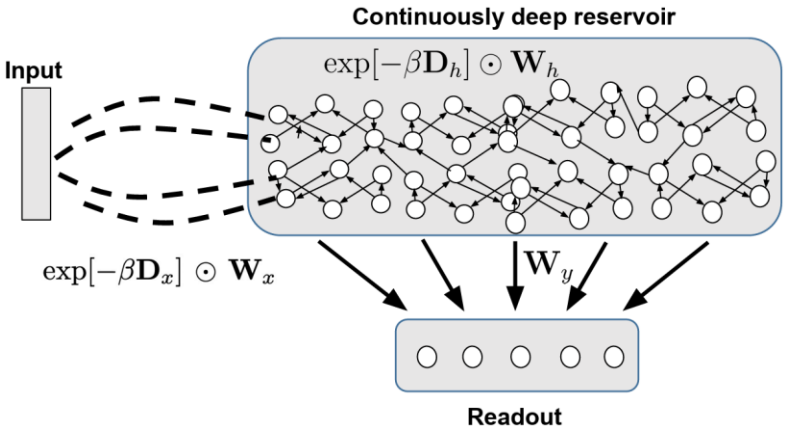


Figure 16. An example of C-DESN architecture. The external input drives the dynamics of the internal reservoir. Along the recurrent architecture, the distance between the hidden neurons and the external input source increases continuously. Only the readout connections are trained.

Our proposal is to consider a neuron-like connector of the following kind:

$$\mathbf{U} = \sigma\big(\widehat{\mathbf{W}}_{\boldsymbol{h}}(\beta)\mathbf{h}(t-1) + \widehat{\mathbf{W}}_{\mathbf{u}}(\beta)\mathbf{u}(t)\big),$$

where $\widehat{\mathbf{W}}_{\boldsymbol{h}}(\beta) = \exp[-\beta \mathbf{D}_{\boldsymbol{h}}] \odot \mathbf{W}_{\boldsymbol{h}}$, and $\widehat{\mathbf{W}}_{\mathbf{u}}(\beta) = \exp[-\beta \mathbf{D}_{\mathbf{u}}] \odot \mathbf{W}_{\mathbf{u}}$, and $\odot$ represents the component-wise multiplication of matrices (also called Hadamard product), i.e. $(\mathbf{A} \odot \mathbf{B})_{ij} = (\mathbf{A})_{ij}(\mathbf{B})_{ij}$. Here, $\exp[\mathbf{M}]$ denotes the component-wise exponentiation of all the entries of the matrix $\mathbf{M}$, i.e. $(\exp[\mathbf{M}])_{ij} = e^{(\mathbf{M})_{ij}}$, not to be confused with the matrix exponential. The two matrices $\mathbf{D}_{\boldsymbol{h}}$and $\mathbf{D}_{\mathbf{u}}$ are defined as follows:

$$\mathbf{D}_{\boldsymbol{h}} = \frac{1}{N_h - 1}\begin{bmatrix} 0 & 1 & 2 & \dots & N_h - 1 \\ 1 & 0 & 1 & \dots & N_h - 2 \\ 2 & 1 & 0 & \dots & N_h - 3 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ N_h - 1 & N_h - 2 & N_h - 3 & \dots & 0 \end{bmatrix},$$

$$\mathbf{D}_{\mathbf{u}} = \frac{1}{N_h - 1}\begin{bmatrix} 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \dots & \vdots \\ N_h - 1 & N_h - 1 & \dots & N_h - 1 \end{bmatrix}.$$

The structure of these two matrices encodes a local connectivity patter among the units of the network. In fact, the entries $(\mathbf{D}_h)_{ij}$ represent the discrete distance between the i-th neuron and the j-th neuron, normalized by the total number of neurons. Therefore, the mask $\exp[-\beta \mathbf{D}_h]$ applied on the matrix $\mathbf{W}_h$ has the effect of reducing the off-diagonal elements, giving values approaching 0 when pushing $\beta \to +\infty$. Large values of β force an exponentially weak coupling between i-th and j-th neurons if $|i - j|$ is large. This promotes the emergence of an effective depth in the recurrent architecture as a function of β. For this reason, we dub the hyperparameter β the continuous depth (or just depth) of the model. Similarly, the mask $\exp[-\beta \mathbf{D}_{\mathbf{u}}]$ for the input-to-hidden matrix $\mathbf{W}_{\mathbf{u}}$, produces an architecture in which the units are located at progressively greater distances from the point where the input signal is injected. Note that we use the same hyperparameter β for masking both the hidden-to-hidden and the input-to-hidden matrices. The element-wise multiplication with the two matrices $\exp[-\beta \mathbf{D}_h]$

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

and $\exp[-\beta \mathbf{D}_{\mathbf{u}}]$ produce a specific neuron-like connector (as defined in section 2.2 of deliverable D3.1) that takes into account a notion of distance between the units of the network. This connector weakens exponentially the couplings among the units that are far from each other according to a predefined order of the units in the network.

In the next section, we report a preliminary experiment on time series reconstruction for an Archetype Network defined as follows:

$$\mathbf{h}(t) = \sigma\big(\widehat{\mathbf{W}}_h(\beta)\mathbf{h}(t-1) + \widehat{\mathbf{W}}_{\mathbf{u}}(\beta)\mathbf{u}(t)\big),$$

that is, directly through the neuron-like connector discussed above.

Time series reconstruction

In this set of experiments we consider four real-world collection of sequences. Arabic [79] is a collection of audio sequences spoken in Arabic from different speakers. Blink [80] is a collection of 4-channels EEGs recorded on different subjects during eye blinking. Epilepsy [81] is a collection of accelerometer data on the dominant wrist collected on different subjects during normal activities and during seizures. Finally, Phoneme [82] is a collection of segmented audio phonemes by different speakers.

We evaluate the ability of C-DESNs to reconstruct previous inputs $\mathbf{u}(T-\tau)$ at different delays $\tau > 0$ from the final state $\mathbf{h}(T)$. In the Figure 17 below we report curves of the test set mean squared error (MSE) of the reconstructed inputs for different delays and different values of β. In all four tasks we notice a valley around the optimal value of β, evidenced with the red line in the plots of Figure 17; the corresponding connectivity patterns are reported on the right. This behaviour is most pronounced on the Blink task, where reconstruction delays up to 50 are considered, and the optimal β is ≈ 21.55. In the remaining tasks in which we considered delays up to $8-12$, the valley around the optimum is less pronounced, and the optimal value is a slightly smaller $\beta \approx 6.00$. These choices of β correspond to a trade-off between depth and width in the connectivity patters of C-DESNs: (a) for larger values of β the units become poorly connected to each other, and in the limit completely disconnected for $\beta \to \infty$ as the recurrent matrix $\mathbf{W}_h$ becomes diagonal; while (b) for smaller values of β the connectivity looses depth, and thus the ability to hold longer history on the input sequence as observed in previous

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

sections.

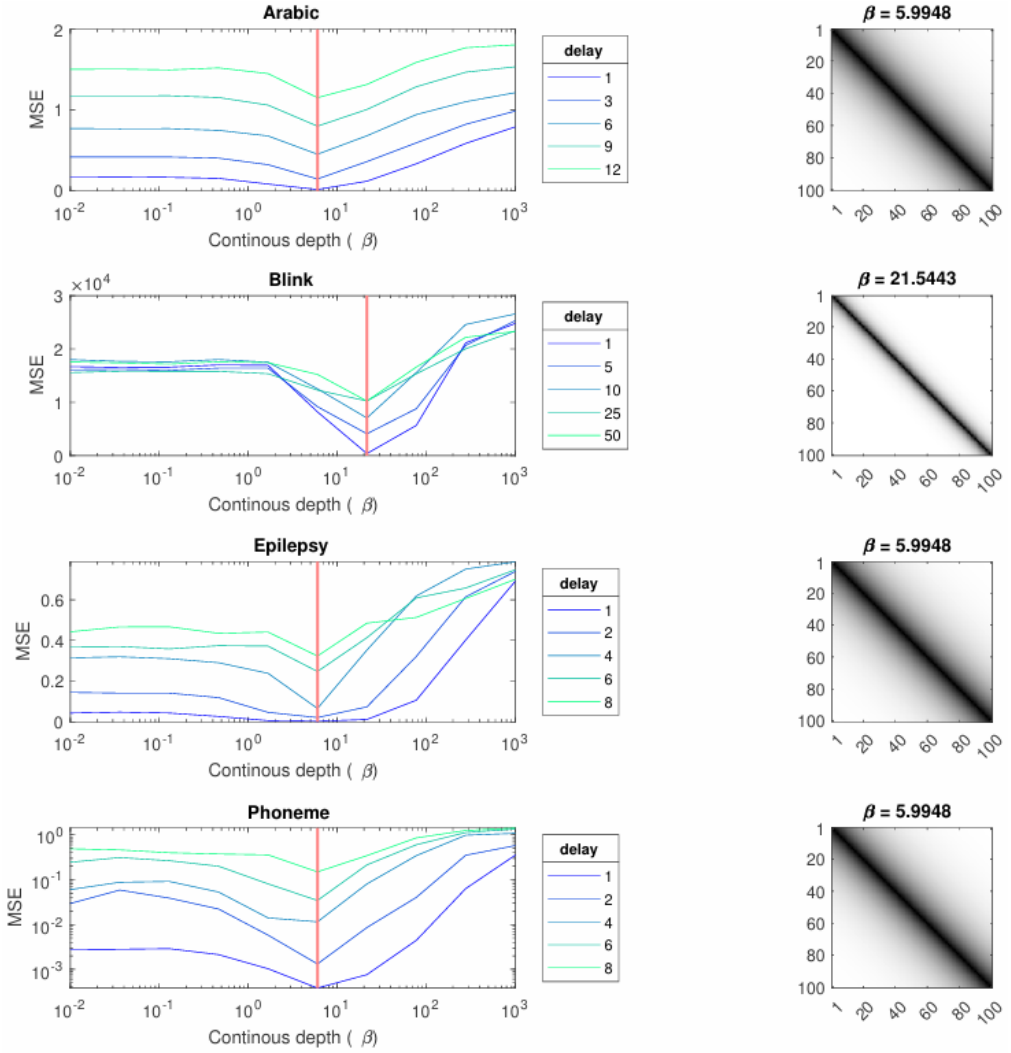


Figure 17. Reconstruction of delayed inputs on four real-world tasks by C-DESN with 100 units. The connectivity of the best β highlighted with the red bar on left plots is displayed on the right of each task.

5. Preliminary analysis on the Archetype Adapting System

The archetype computing system provides the theoretical and methodological framework to compute with archetypes. The archetype adapting system allows to train archetype units and networks to reach a predefined goal.

While the archetype adapting system is planned to be fully developed later in the project, we already started to explore some of its related challenges.

5.1. Learning to reject

Maximizing the predictive accuracy of a model is one of the main objectives pursued during training of machine learning models. However, in all interesting cases models never reach a perfect predictive accuracy. We are interested in designing models that can tell whether they are likely to provide a good prediction [83]. Otherwise, the model can refuse to make a prediction, since it would likely lead to a mistake [84]. This ability to *learning to reject* or to

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

selectively classify is a useful property both for real-world applications and for the study of metacognition, where a model is aware of its inner state.

Given a model f parameterized by θ, its output on an input x is represented by $\hat{y} = f_\theta(x)$. The model can *reject* an example if its confidence c is smaller than a threshold: $c < c^*$. The confidence can be computed in different ways. We experimented with two methods: the maximum probability and the Deep Gamblers methods [85].

We trained a ResNet 101 to recognize a given set of objects from images. We chose the CIFAR-100 dataset and we trained the ResNet with cross-entropy loss and gradient descent.

For the maximum probability method, the ResNet output $\hat{y}$ is a vector of probabilities associated with each possible object class. The class that shows the largest probability is taken as the final prediction. We selected the maximum probability as the model confidence [84]. We then compute the coverage curve: we sorted all the examples in the test set of CIFAR-100 in descending order based on the confidence computed by the model. Then, we took the first X% of the ordered dataset and we compute the average accuracy on all the examples, for different values of X. The coverage curve is monotonically decreasing, since as X grows, the model is considering examples which it is less confident on.

We compare the coverage curve of the maximum probability confidence with the same curve computed by the Deep Gamblers method [85]: a method which is specifically designed to learn accurate confidence values. Deep Gamblers modifies the cross-entropy loss by computing:

$$L(\hat{y}, y) = \sum_{i=1}^{m} y \log(\hat{y}_i + o\, \widehat{y_{m+1}})$$

where $\hat{y}$ is the probability vector computed by the model for all m classes and y is the target class corresponding to the input x fed to the model. The output layer in Deep Gamblers has $m + 1$ units. In fact, Deep Gamblers reserves a special unit for the rejection phase: the value of the unit represents the rejection probability. Hence, we take the confidence to be $1 - \widehat{y_{m+1}}$. The scalar hyper-parameter o is chosen by model selection on a held-out validation set and it represents the preference of the network towards predicting instead of rejecting. Setting $o = 0$ recovers the cross-entropy loss.

The maximum probability method reached an accuracy of 64% on the test set and an accuracy of 97% on the training set. Deep Gamblers reported a training accuracy of 61% and a test accuracy of 43%. Both models show over-fitting and Deep Gamblers struggle to learn properly. We report the coverage curve obtained by the best models on the test set.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

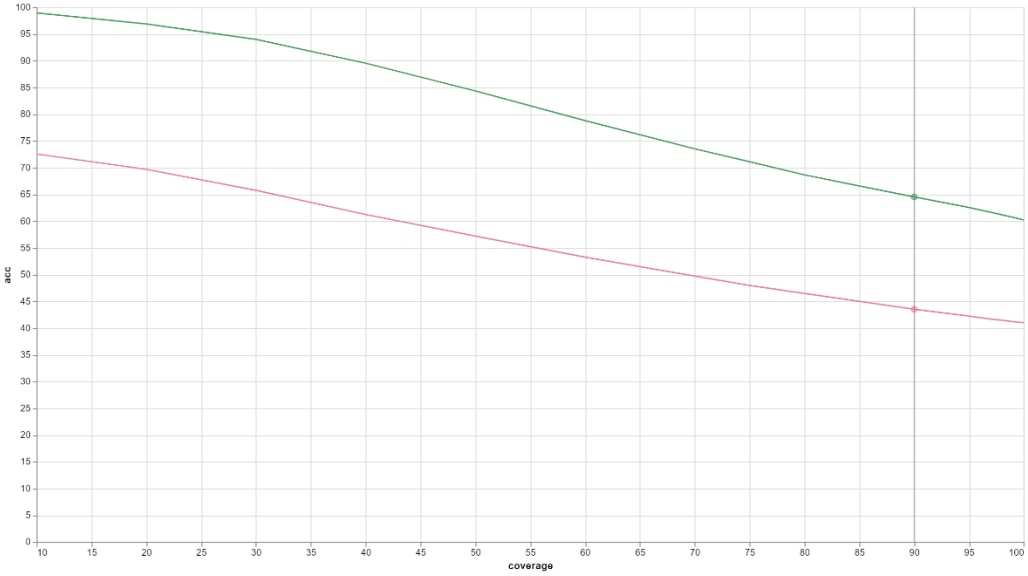


Figure 18. Validation coverage curve. Red curve = Deep Gamblers. Green curve = maximum probability.

From Figure 18 we see how the Deep Gamblers is unable to maintain a good performance for different confidence level. This hints at the fact that learning a separate confidence unit like Deep Gamblers does is much more challenging than leveraging a built-in confidence value, like with the maximum probability method.

We leverage these preliminary findings and turn our attention to the calibration challenge. To build calibrated models, we always rely on the maximum probability confidence.

5.2. Lifelong Calibration

Learning to reject methods like Deep Gamblers can optimize the accuracy for a given coverage by learning ad-hoc confidence values. However, the confidence value is not guaranteed to be *calibrated* [83]. A model is calibrated when its confidence also represents the average prediction accuracy scored by the model on the examples with that confidence. Formally, a model is calibrated when on any given input-target pair (x, y)

$$p(\hat{y} = y | \, c = \hat{c}) = \hat{c}$$

When a model is calibrated, the confidence represents a true probability of correct predictions. Since the confidence c is a continuous random variable, the probability cannot be exactly computed by any finite number of examples. Calibration approaches approximates the calibration objective in different ways.

We follow the reliability diagram approach. The reliability diagram groups the model predictions on a given set of data according to M interval bins based on the confidence. Then, it computes the average accuracy for each bin. A model is then considered calibrated if the average accuracy for each bin is equal to the middle bin value. We can concisely express calibration through the Expected Calibration Error (ECE):

European Innovation Council
Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

$$ECE = \sum_{i=1}^{b} \frac{|B_i|}{n} |acc(B_i) - conf(B_i)|$$

where $|B_i|$ is the number of examples contained in the i-th bin of the reliability diagram, n is the total number of examples in the dataset and $acc(B\_i), conf(B\_i)$ are the average accuracy and confidence over the examples contained in the i-th bin, respectively. The ECE is zero only when the model is perfectly calibrated on the dataset. Since both the accuracy and the confidence are between zero and one, the ECE (worse possible calibration) is always smaller than 1. The ECE is usually reported as a percentage by multiplying it by 100.

Calibrating models on a given dataset is a well-known field of research. However, calibration is under-explored with artificial neural networks and there is no work focusing on lifelong calibration. This is a very promising topic, since a model that learns continuously over time needs also to adjust its calibration according to the dynamics of the environment.

We adopt a lifelong learning setup [86] where the model is trained on a sequence of experiences $e_1, e_2, \ldots$. Each experience e_i contains a training dataset D_i^{tr} and a validation dataset D_i^{vl}. The test dataset associated to each experience is always available, while the training and validation datasets are only available when training on that experience. The objective is to train the model on all experiences and to preserve the predictive performance while keeping the model calibrated.

We started exploring lifelong calibration by selecting some of the calibration methods for artificial neural networks and some of the most effective lifelong learning strategies.

Table 11. Accuracy of calibration methods with CL strategies on a set of CL benchmarks. J=Joint, E=entropy regularization, TS=temperature scaling, MS/VS=matrix/vector scaling, R=replay, N=Naïve, MD=mixed data.

	SplitMNIST	*SplitCIFAR100*	*EuroSAT*	*Atari*
J	97.85 ± 0.51	65.13 ± 6.90	90.42 ± 3.07	55.10 ± 0.61
J + E	98.30 ± 0.33	65.43 ± 1.32	95.02 ± 2.96	54.82 ± 0.71
J + TS	98.73 ± 0.13	67.57 ± 5.79	91.80 ± 2.15	55.38 ± 0.42
J + VS	98.59 ± 0.39	65.17 ± 4.82	92.36 ± 1.47	55.32 ± 0.30
J + MS	95.97 ± 1.13	64.30 ± 4.10	96.37 ± 2.09	39.27 ± 2.04
R	81.25 ± 2.06	39.95 ± 0.75	81.14 ± 1.02	29.30 ± 0.66
R + E	83.80 ± 1.49	39.01 ± 3.21	81.14 ± 2.33	29.87 ± 0.50
R + TS	84.77 ± 1.13	33.94 ± 7.14	78.67 ± 3.44	28.29 ± 0.51
R + VS	83.38 ± 2.40	33.44 ± 2.18	81.64 ± 0.69	28.45 ± 0.19
R + MS	83.09 ± 3.05	33.06 ± 0.72	80.37 ± 3.05	28.98 ± 0.38
R + TS + MD	87.48 ± 0.60	36.53 ± 8.51	78.27 ± 2.67	29.03 ± 0.42
R + VS + MD	92.54 ± 0.42	47.18 ± 1.28	82.39 ± 1.55	27.83 ± 0.93
R + MS + MD	92.71 ± 0.44	46.76 ± 0.58	81.94 ± 2.42	28.34 ± 0.76
N	20.08 ± 0.37	7.57 ± 0.37	19.40 ± 0.44	20.57 ± 1.76
N + E	19.90 ± 0.02	7.51 ± 0.59	18.94 ± 1.46	19.95 ± 0.49
N + TS	20.52 ± 0.52	8.19 ± 0.41	19.51 ± 0.78	20.96 ± 0.94
N + VS	19.98 ± 0.02	7.84 ± 0.19	19.90 ± 0.09	20.66 ± 0.62
N + MS	19.89 ± 0.06	8.36 ± 0.32	19.53 ± 0.49	19.49 ± 0.25
N + TS + MD	19.95 ± 0.03	8.31 ± 0.13	19.59 ± 0.35	21.40 ± 0.02
N + VS + MD	35.38 ± 12.17	11.04 ± 0.51	17.36 ± 3.77	20.87 ± 1.01
N + MS + MD	32.00 ± 2.73	10.71 ± 1.23	19.21 ± 0.84	21.96 ± 1.06

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Entropy calibration. The entropy calibration method [87] regularizes the loss by adding a component that controls the entropy of the output layer of the network. The penalization term reads:

$$-\lambda \sum_{i=1}^{m} \hat{y}_i \log(\hat{y}_i)$$

where the hyper-parameter λ controls the regularization strength.

Temperature scaling. The temperature scaling method [83] is a post-processing method that is applied at the end of each experience in continual learning. The probabilities computed by the output layer of the network are obtained through a softmax function. If we divide the input to the softmax (the logits) by a factor T (called the temperature), we can control the smoothness of the resulting probability distribution. Peaked probability distributions often corresponds to uncalibrated models. The temperature scaling method learns via backpropagation the optimal temperature value. The optimization is performed on the validation set of the current experience and the loss is the Negative Log Likelihood loss.

Matrix/vector scaling. These two approaches are post-processing approaches that, like the temperature scaling, are applied at the end of each experience on the validation set of the current experience [83]. The approaches learn an additional linear layer after the output layer of the network. The matrix W_{clb} and the bias b_{clb} are optimized with respect to the Negative Log Likelihood loss. To avoid a quadratic dependence on the number of classes, the vector scaling variant learns a vector w_{clb} instead of a full matrix.

Mixed data. We specifically designed a novel calibration approach for lifelong learning. Instead of applying existing post-processing calibration approaches on the current validation set, we couple them with a replay strategy that stores a subset of previous validation sets into an external memory. In this way, the calibration approach also tunes the temperature or the additional linear layer on previous data distributions, thus improving the stability of the calibration when learning multiple data distributions over time.

We combine the aforementioned approaches with a naïve finetuning, that simply trains the model continuously on all experiences, and a replay approach that leverages a fixed-size external memory on which it stores a subset of previous examples. At training time, the examples in the memory are combined with the examples from the current training set, to mitigate forgetting.
We also provide results for the Joint Training, that simulates offline learning by concatenating all the experiences together into a single dataset.

We report the average accuracy (Table 11) and ECE (Table 12) on the test set of all experiences after training on all experiences.

We adopt 4 lifelong learning benchmarks.

Split MNIST/CIFAR-100. The object recognition datasets MNIST [88] and CIFAR-100 [89] are split into 5 and 10 experiences, respectively, by randomly selecting a subset of the classes for each experience (without replacement.

EuroSAT. EuroSAT is an object recognition dataset from satellite images. We built a lifelong learning stream composed by 5 experiences by randomly splitting the available classes into 5 non-overlapping subsets.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Atari. We selected environment transitions from 3 Atari games and we trained continuously a network to choose the best action given the transition (environment state, action to be predicted, reward after the action, next environment state). The output space is composed by all Atari actions, but on each game only a subset of the actions is available.

The preliminary results show that the ability of existing calibration strategies is seriously harmed in a lifelong learning environment. The ECE always increases with respect to the Joint Training performance if no ad-hoc approaches are employed.

Our mixed data approach helps in preserving a better calibration over the course of training, but sometimes this comes at the cost of a reduced predictive performance (lower average accuracy). This is an important trade-off that occurs in many experiments: prioritizing a calibrated model impacts negatively on the final accuracy.

In the future, we plan to focus on the design of novel calibration strategies for lifelong learning that can find a better accuracy-calibration trade-off than existing approaches.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 12. Expected Calibration Error (ECE) of calibration methods with CL strategies on a set of CL benchmarks. J=Joint, E=entropy regularization, TS=temperature scaling, MS/VS=matrix/vector scaling, R=replay, N=Naïve, MD=mixed data

	SplitMNIST	*SplitCIFAR100*	*EuroSAT*	*Atari*
J	1.30 ± 0.25	16.46 ± 3.80	4.72 ± 3.20	2.20 ± 1.83
J + E	1.06 ± 0.28	15.38 ± 1.88	2.93 ± 1.22	1.90 ± 0.79
J + TS	0.61 ± 0.20	7.32 ± 2.19	3.47 ± 1.06	1.52 ± 0.71
J + VS	0.49 ± 0.22	6.21 ± 2.51	4.47 ± 0.14	1.38 ± 0.20
J + MS	41.72 ± 2.77	29.50 ± 2.90	80.38 ± 2.21	23.50 ± 2.97
R	13.28 ± 2.24	39.34 ± 1.14	10.96 ± 2.77	48.94 ± 0.68
R + E	11.79 ± 1.25	36.54 ± 3.31	9.83 ± 3.96	48.68 ± 1.24
R + TS	10.28 ± 1.25	25.63 ± 1.68	13.05 ± 2.57	49.96 ± 2.28
R + VS	12.81 ± 1.91	48.21 ± 3.51	11.52 ± 0.42	34.78 ± 5.09
R + MS	12.62 ± 2.39	46.47 ± 1.26	12.34 ± 3.16	41.13 ± 2.62
R + TS + MD	7.61 ± 0.39	18.07 ± 5.42	9.50 ± 1.25	41.12 ± 1.87
R + VS + MD	3.35 ± 0.73	24.32 ± 1.49	10.26 ± 1.34	13.75 ± 1.23
R + MS + MD	2.30 ± 0.21	20.20 ± 1.06	10.02 ± 2.09	13.46 ± 3.40
N	76.76 ± 0.67	65.31 ± 4.81	74.85 ± 0.91	34.60 ± 5.35
N + E	76.78 ± 1.85	65.84 ± 1.62	74.19 ± 5.71	34.51 ± 4.20
N + TS	75.28 ± 1.74	67.49 ± 0.83	77.09 ± 1.49	47.10 ± 3.33
N + VS	73.04 ± 2.70	60.55 ± 4.14	77.79 ± 0.79	26.35 ± 8.56
N + MS	74.85 ± 1.77	65.21 ± 3.58	73.92 ± 4.02	35.77 ± 5.71
N + TS + MD	73.08 ± 0.56	18.07 ± 5.42	76.89 ± 0.61	42.03 ± 4.61
N + VS + MD	24.53 ± 6.05	24.32 ± 1.49	60.92 ± 12.10	29.20 ± 3.00
N + MS + MD	26.31 ± 4.20	20.20 ± 1.06	63.49 ± 1.14	13.87 ± 5.29

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

6. Software Framework: design considerations of the ACDS Python library

Figure 19. Archetype Computing and Adaptive System library from GitHub

The Archetype Computing and aDaptive System (ACDS) library (Figure 19) is a shared codebase that collects the utilities to build and assess the performance of Archetype Networks. The library is developed by following a modularity principle that i) fosters (re)usability, ii) allows fast prototyping and iii) ensure reproducibility.

Modularity. Separation of concerns is achieved by implementing different sets of functionalities into different, separate modules. The library is currently composed of 3 modules: 1) archetypes, 2) benchmarks, 3) experiments.
The archetypes module provides all the available archetypes units and networks. The module is independent from the rest of the library and can be used as a standalone component in external projects.
The benchmarks module allows to load datasets that can be later used to assess the performance of the archetypes. Like the archetypes module, the benchmarks module is also a standalone component that can be easily used outside the library.
The experiments module leverages on the previous two modules to build fully-fledged experiments. A typical experiment creates an instance of an archetype, loads a dataset and runs the archetype on the dataset. The experiment can include a learning phase in which the archetype's parameters are adapted to the specific dataset. The experiments module cannot be used as an independent component, since it heavily relies on the other modules of the ACDS library.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

(Re)usability. The modular architecture of the ACDS library allows to easily combine and reuse components in different experiments. Each archetype and each benchmark is implemented only once and then reused across different experiments. Also, modularity positively affects usability, since it is always clear where each component comes from.

Fast prototyping. The library allows to quickly experiment with new archetypes and new datasets. For example, as soon as a new archetype is defined, it can be implemented in the archetypes module and then added to any experiment. The same process works for a new dataset. Whenever a new archetype-benchmark combination is needed, one can create and run a new experiment.

Reproducibility. The library guarantees that any documented result involving archetypes can be easily verified and reproduced. In case an experiment requires some input parameter, it is sufficient to share the exact experiment configuration to get the same results. The code is released as an open-source project. As such, its changes are public and they can be easily monitored and inspected.

Preliminary implementation. The ACDS library is publicly available on GitHub, preliminarily via the following link: https://github.com/EU-EMERGE/archetype-computing-adaptive-system The ACDS library is being developed in Python. The library can already be used to run experiments with the RON archetype on a set of popular benchmarks, commonly used for sequence processing. The benchmarks include both time series forecasting and sequence classification datasets. The library also implements models that are not necessarily archetypes. This is useful to compare the performance of archetypes like RON with other existing models, like the Echo State Network and the Long Short-Term Memory network.

6.1. ACDS components

The library offers three packages: archetypes, benchmarks and experiments, as showed below.

```
├──acds: the core of the ACDS library
│       └──archetypes: module containing archetypes and baselines models
│              └──esn: the Echo State Network baseline
│              └──lstm: the Long Short-Term Memory baseline
│              └──ron: the Random Oscillators Network archetype
│              └──utils: utility functions for archetypes
│       └──benchmarks: module containing the main datasets and benchmarks
│              └──adiac: the Adiac dataset implementation
│                     └──dataset
│                     └──getter
│              └──mackey_glass: the Mackey Glass dataset implementation
│              └──mnist: the MNIST dataset implementation
│              └──rc_dataset: a dataset class for Reservoir Computing models
├──experiments: module that runs archetypes and baselines on a given benchmark
│       └──adiac: experiment on the Adiac dataset
│       └──mackey_glass: experiment on the Mackey Glass dataset
│       └──smnist: experiment on the sequential MNIST dataset
```

Archetypes. The archetype package implements the *Echo State Network model*, the *Long Short-Term memory model*, and the *Random Oscillators Network model*. The first two models are not Archetypes. Rather, they are used as baselines (established within the deep learning

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

community) for comparison with respect to the Archetypes (like RON). Each model is implemented as a Python class using the PyTorch library[5]. The archetypes package defines the way each archetype performs computation and it is therefore at the center of the Archetype Computing System. Moreover, the archetype package also provided utility functions that influence the archetype computation. For example, currently available functions can create sparsely connected Archetype Networks or control the topology of the connections (ring topology, Toeplitz topology and others).

Benchmarks. The benchmarks package allows to easily create an instance of popular datasets and benchmarks. Currently, the package features the *MNIST dataset*, the *Mackey-Glass chaotic dynamical system* and the *Adiac dataset* (a real-world time series processing task).

The datasets are currently implemented via helper functions. Each dataset has its own helper functions that returns the training set, the validation set and the test set randomly sampled from the dataset. Each set is a PyTorch compatible dataset or dataloader, that automatically manages multiple iterations with mini-batches. Each dataset can implement a custom way of splitting the examples in the three sets. From the point of view of the user, this does not impact neither on the archetypes nor on the experiment where the dataset is then used.

Experiments. The experiments package currently provides three experiments. Each experiment is a main Python script that can be executed by directly running it with a Python interpreter. The *Adiac experiment* uses the Adiac dataset and trains either an RON model or an ESN model. The choice of the model can be controlled by parameters provided as input to the main script. The experiments logs the results to a text file with the train and validation metrics. The test metrics can be computed when the corresponding flag is activated (to distinguish between model selection and model assessment). *The sMNIST experiment* follows the same approach, but it leverages the MNIST dataset, taken one pixel at a time (sequential MNIST) due to the use of recurrent archetypes. Finally, the *Mackey-Glass experiment* uses the Mackey-Glass dataset and trains an RON model or an ESN model to predict future states given the current one. The logging functionalities remain the same across the three experiments. Each experiment describes all the accepted parameters at the beginning of the code.

The library can easily be used by running the experiments contained in the experiments package. For ease of use, the library should be added to the PYTHONPATH environment variable, such that it behaves as any other installed Python libraries. Then, to run sMNIST with RON one simply runs (assuming the user is within the acds folder of the library):

python experiments/smnist.py --ron

Adapting the command to run other experiments is trivial, by just pointing to the desired experiment file and providing the desired parameters as input.

The ACDS library will be extended over time to include new features and new modules. For example, the Archetype Adapting System will be progressively added to the library with a dedicated module. The module will implement learning algorithms, even beyond back-propagation, to adapt the parameters of the archetypes.

The final version of the library will provide the full Archetype Computing and Adaptive System.

[5] https://pytorch.org/

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

7. Conclusions

This deliverable outlines the initial efforts and achievements of Work Package 4 within the EMERGE project, focusing on the development and preliminary analysis of the Archetype Computing System (ACS). The document presents the conceptual framework for the ACS, emphasizing the innovative approach of utilizing Random Oscillators Networks (RON) and other neural-inspired models to enhance adaptive and lifelong learning capabilities. Moreover, initial findings on lifelong learning strategies offer insights into their applicability for recalibration and rejection tasks, representing preliminary and useful work in perspective of development efforts geared toward learning algorithms in ADS. This deliverable also includes a software library written in Python, which collects the utilities to build and evaluate the methodologies progressively included in the ACS and ADS.

From a broader perspective, this deliverable contributes to fulfilling Milestone 2 of the project, by introducing the preliminary implementation of archetypes-based computing framework. At the same time, this deliverable is instrumental to set the stage for subsequent efforts in the project, including the development of the ADS as outlined in D4.2, the next deliverable of WP4.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

8. Appendix A – Proof of mathematical results

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

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

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

from which it follows that $\left\| \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \right\| \leq \max(\|\mathbf{M}\|, \|\mathbf{N}\|)$. 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

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

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

#

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

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

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

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

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

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

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

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

$$\|\mathbf{I}-\tau^2\mathrm{diag}\,(\gamma)+\tau^2\mathbf{S}_k\mathbf{W}\|\leq$$
$$\leq\|\mathbf{I}-\tau^2\mathrm{diag}\,(\gamma)\|+\tau^2\|\mathbf{S}_k\mathbf{W}\|\leq$$
$$\leq\max_j\left|1-\tau^2\gamma_j\right|+\tau^2\parallel\mathbf{W}\parallel=\eta+\tau^2\sigma,$$

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.

$$||\mathbf{I}-\tau\mathrm{diag}\,(\varepsilon)\parallel=\max_j\left|1-\tau\varepsilon_j\right|=\xi$$

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

$$\parallel\tau(\mathbf{I}-\tau\mathrm{diag}\,(\varepsilon))\parallel=\tau\max_j\left|1-\tau\varepsilon_j\right|=\tau\xi.$$

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

$$\|\tau(\mathbf{S}_k\mathbf{W}-\mathrm{diag}\,(\gamma))\|\leq\tau(\|\mathbf{S}_k\mathbf{W}\|+\|\,\mathrm{diag}\,(\gamma)\,\|)\leq$$
$$\leq\tau(\|\,\mathbf{W}\,\|+\gamma_{\max})=\tau(\sigma+\gamma_{\max}).$$

Putting together all the relations above, and Lemma B.1, we obtain

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

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

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

#

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

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

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

$$\mathbf{y}_{k+1} = (\mathbf{I} - \tau^2 \mathrm{dia\,g}(\gamma))\mathbf{y}_k + \tau^2 \tan \mathrm{h}(\mathbf{W}\mathbf{y}_k + \mathbf{V}\mathbf{u}_{k+1} + \mathbf{b})\,. \tag{38}$$

We provide the following sufficient conditions for contraction in the particular case of (38).
Proposition C.1. In the particular case of $\varepsilon \equiv \frac{1}{\tau}$, 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 $\mathbf{J}_k = \mathbf{I} + \tau^2\mathbf{A}_k = (\mathbf{I} - \tau^2 \mathrm{diag}\,(\gamma)) + \tau^2\mathbf{S}_k\mathbf{W}$. Therefore, it holds $\| \mathbf{J}_k \| \leq \| \mathbf{I} - \tau^2 \mathrm{diag}\,(\gamma) \| + \| \tau^2\mathbf{S}_k\mathbf{W} \| \leq \eta + \tau^2\sigma$. 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 $\tau^2\gamma_{\max} \leq 2$.

#

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

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

European Innovation Council | Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

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 $\sigma \leq \min\left(\frac{\xi-\eta}{\tau^2}, \xi - \gamma_{\max}\right)$. 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 $\sigma \geq \max\left(\xi - \gamma_{\max}, \frac{\xi-\eta}{\tau^2}\right)$. 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.

#

Proof of Theorem 3.3

The proof is a straightforward application of the BauerFike's theorem [90] that we report here for ease of comprehension.

Theorem E.1 (Bauer-Fike). Let D be a diagonalisable matrix, and let $\mathbf{H}$ be the eigenvector matrix such that $\mathbf{D} = \mathbf{H}\boldsymbol{\Lambda}\mathbf{H}^{-1}$ where $\boldsymbol{\Lambda}$ 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

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

Let us denote

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

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

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

$$\|\mathbf{E}_k\| \leq C.$$

where C is defined as follows

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

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

$$|\mu - \lambda| \leq C,$$

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

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

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

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

#

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

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

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

$$\begin{aligned} 1 - \tau^2\gamma_{\min} + C \leq 1, \\ 1 - \tau^2\gamma_{\max} - C \geq -1, \\ 1 - \tau\varepsilon_{\min} + C \leq 1, \\ 1 - \tau\varepsilon_{\max} - C \geq -1. \end{aligned}$$

The above inequalities can be rewritten as follows

$$\begin{aligned} C \leq \tau^2\gamma_{\min} \leq \tau^2\gamma_{\max} \leq 2 - C, \\ C \leq \tau\varepsilon_{\min} \leq \tau\varepsilon_{\max} \leq 2 - C. \end{aligned}$$

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 hooRNN 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 $\sigma \leq \frac{1-\tau\gamma_{\max}}{\tau+\tau^2}$.
- if $\sigma \leq \xi - \gamma_{\max}$, then $\sigma \leq \frac{1-\tau\xi}{\tau^2}$.

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 $\sigma \leq \frac{1-\tau\gamma_{\max}}{\tau+\tau^2}$.
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 $\sigma \leq \frac{1-\tau\xi}{\tau^2}$.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

9. Appendix B – Details on experimental settings

9.1 Random Oscillators Network

We report the best hyperparameters configuration found by model selection for the experiments with RON.

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 [39]. 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 [39].

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].

9.2 Sparse RON topologies

Unless otherwise noted, we used 100 units in the reservoir. For the grid search on the validation set of each dataset, we considered $\gamma \in \{2,10,20\}, \gamma_r \in \{2,10\}, \epsilon \in \{2,10,20\}, \epsilon_r \in \{2,10\}, \rho \in \{0.9,9\}, \nu \in \{0.1,1,10\}$. The hyperparameters γ_r and ϵ_r denote, respectively, the ranges of heterogeneity of stiffness and dampening of the oscillators in the reservoir. In essence, once selected center values γ, ϵ and range values γ_r, ϵ_r for stiffness and dampening, we generate oscillators with random stiffness values i.i.d. uniformly sampled in $\left[\gamma - \frac{\tau_r}{2}, \gamma + \frac{\tau_r}{2}\right]$, and random dampening values i.i.d. uniformly sampled in $\left(\epsilon - \frac{\epsilon_x}{2}, \epsilon + \frac{\epsilon_e}{2}\right]$. The RON's time-scale τ has been selected in $\{0.042,0.42,4.2\}$ for sMNIST, in $\{0.01,0.1,1\}$ for Adiac and in $\{0.017,0.17,1.7\}$ for Mackey-Glass. For the Leaky ESN we considered $\rho \in \{0.9.0.99,0.999.9\}, \alpha \in \{0.001,0.01,0.1\}, \nu \in \{0.1,1,10\}$, while, each element of the input-to-hidden matrix W^{in} and the hidden-to-hidden matrix W is uniformly sampled in $[-2,2]$. The bias is uniformly sampled in $[-1,1]$. We repeated the same grid search when scaling up the number of hidden units. Mean and standard deviation is computed on test over 5 different random seeds.

Table 13 shows the optimal hyper-parameters for all experiments in Experiments of Random Oscillators Networks.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

Table 13. RON hyper-parameters

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	$\tau = \{0.05, \mathbf{0.01}\}, \rho = \{\mathbf{9}, 0.9\}, \nu = \{\mathbf{10}, 1, 0.1\}, \epsilon = \{\mathbf{5}, 0.5\} \pm \{5, 2.5, \mathbf{0.5}\}, \gamma = \{\mathbf{3}, 1\} \pm \{\mathbf{1}, 0.5, 0.25\}$

European Innovation Council

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

10. Bibliography

[1] A. Ceni *et al.*, “Randomly coupled oscillators for time series processing,” in *ICML workshop on new frontiers in learning, control, and dynamical systems*, 2023.
[2] A. Ceni *et al.*, “Random Oscillators Network for Time Series Processing,” in *To appear in AISTATS 2024*, 2024.
[3] A. Cossu, A. Ceni, D. Bacciu, and C. Gallicchio, “Sparse Reservoir Topologies for Physical Implementations of Random Oscillators Networks,” in *currently under review*, 2024.
[4] C. Gallicchio, “Euler State Networks: Non-dissipative Reservoir Computing,” *Neurocomputing*, vol. 579, p. 127411, Apr. 2024, doi: 10.1016/j.neucom.2024.127411.
[5] A. Ceni and C. Gallicchio, “Residual reservoir computing neural networks for time-series classification,” presented at the ESANN, 2023.
[6] A. Ceni, C. Gallicchio, D. Tortorella, L. Pedrelli, P. F. Dominey, and A. Micheli, “Continuously deep recurrent neural networks,” in *currently under review*, 2024.
[7] L. Li, E. Piccoli, A. Cossu, D. Bacciu, and V. Lomonaco, “Calibration of Continual Learning Models,” in *currently under review*, 2024.
[8] J. F. Kolen and S. C. Kremer, *A field guide to dynamical recurrent networks*. John Wiley & Sons, 2001.
[9] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.
[10] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.
[11] K. Nakajima and I. Fischer, *Reservoir computing*. Springer, 2021.
[12] M. Lukoševičius and H. Jaeger, “Reservoir computing approaches to recurrent neural network training,” *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, Aug. 2009, doi: 10.1016/j.cosrev.2009.03.005.
[13] X. Liang *et al.*, “Rotating neurons for all-analog implementation of cyclic reservoir computing,” *Nat. Commun.*, vol. 13, no. 1, p. 1549, 2022.
[14] H. Tan and S. van Dijken, “Dynamic machine vision with retinomorphic photomemristor-reservoir computing,” *Nat. Commun.*, vol. 14, no. 1, p. 2169, 2023.
[15] X. Wu, C. Ott, A. Albu-Schaffer, and A. Dietrich, “Passive Decoupled Multitask Controller for Redundant Robots,” *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 1, Jan. 2023, doi: 10.1109/TCST.2022.3162990.
[16] S. Wang *et al.*, “Echo state graph neural networks with analogue random resistive memory arrays,” *Nat. Mach. Intell.*, vol. 5, no. 2, pp. 104–113, 2023.
[17] C. Gallicchio and A. Micheli, “Fast and Deep Graph Neural Networks,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, Art. no. 04, Apr. 2020, doi: 10.1609/aaai.v34i04.5803.
[18] A. Cini, I. Marisca, F. M. Bianchi, and C. Alippi, “Scalable spatiotemporal graph neural networks,” in *Proceedings of the AAAI conference on artificial intelligence*, 2023, pp. 7218–7226.
[19] L.-W. Kong, Y. Weng, B. Glaz, M. Haile, and Y.-C. Lai, “Reservoir computing as digital twins for nonlinear dynamical systems,” *Chaos Interdiscip. J. Nonlinear Sci.*, vol. 33, no. 3, 2023.
[20] H. Jaeger, “The ‘echo state’ approach to analysing and training recurrent neural networks-with an erratum note,” *Bonn Ger. Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep.*, vol. 148, no. 34, p. 13, 2001.
[21] H. Jaeger and H. Haas, “Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication,” *Science*, vol. 304, no. 5667, pp. 78–80, Apr. 2004, doi: 10.1126/science.1091277.
[22] M. Inubushi and K. Yoshimura, “Reservoir computing beyond memory-nonlinearity trade-off,” *Sci. Rep.*, vol. 7, no. 1, p. 10199, 2017.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

[23] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, “Information processing capacity of dynamical systems,” *Sci. Rep.*, vol. 2, no. 1, p. 514, 2012.
[24] T. Schulte to Brinke, M. Dick, R. Duarte, and A. Morrison, “A refined information processing capacity metric allows an in-depth analysis of memory and nonlinearity trade-offs in neurocomputational systems,” *Sci. Rep.*, vol. 13, no. 1, p. 10517, 2023.
[25] T. L. Carroll, “Optimizing memory in reservoir computers,” *Chaos Interdiscip. J. Nonlinear Sci.*, vol. 32, no. 2, 2022.
[26] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, “Optimization and applications of echo state networks with leaky-integrator neurons,” *Neural Netw.*, vol. 20, no. 3, pp. 335–352, 2007.
[27] N. Bertschinger and T. Natschläger, “Real-time computation at the edge of chaos in recurrent neural networks,” *Neural Comput.*, vol. 16, no. 7, pp. 1413–1436, 2004.
[28] R. Legenstein and W. Maass, “What makes a dynamical system computationally powerful,” *New Dir. Stat. Signal Process. Syst. Brain*, pp. 127–154, 2007.
[29] M. Lukoševičius, “A practical guide to applying echo state networks,” in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 659–686.
[30] S. Lun, H. Hu, and X. Yao, “The modified sufficient conditions for echo state property and parameter optimization of leaky integrator echo state network,” *Appl. Soft Comput.*, vol. 77, pp. 750–760, 2019.
[31] K. Funahashi and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” *Neural Netw.*, vol. 6, no. 6, 1993, doi: 10.1016/s0893-6080(05)80125-x.
[32] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations.” arXiv, Nov. 28, 2017. doi: 10.48550/arXiv.1711.10561.
[33] R. Bischof and M. Kraus, “Multi-objective loss balancing for physics-informed deep learning,” *ArXiv Prepr. ArXiv211009813*, 2021.
[34] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, “Physics-informed neural networks (PINNs) for fluid mechanics: A review,” *Acta Mech. Sin.*, vol. 37, no. 12, pp. 1727–1738, 2021.
[35] J. Li, J. Chen, and B. Li, “Gradient-optimized physics-informed neural networks (GOPINNs): a deep learning method for solving the complex modified KdV equation,” *Nonlinear Dyn.*, vol. 107, pp. 781–792, 2022.
[36] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, “Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems,” *Comput. Methods Appl. Mech. Eng.*, vol. 393, p. 114823, 2022.
[37] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating physics-based modeling with machine learning: A survey,” *ArXiv Prepr. ArXiv200304919*, vol. 1, no. 1, pp. 1–34, 2020.
[38] D. Bacciu, F. Errica, A. Micheli, and M. Podda, “A gentle introduction to deep learning for graphs,” *Neural Netw.*, vol. 129, pp. 203–221, Sep. 2020, doi: 10.1016/j.neunet.2020.06.006.
[39] T. K. Rusch and S. Mishra, “Coupled Oscillatory Recurrent Neural Network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies,” presented at the International Conference on Learning Representations, Apr. 2023. Accessed: Apr. 18, 2023. [Online]. Available: https://openreview.net/forum?id=F3s69XzWOia
[40] S. Lanthaler, T. K. Rusch, and S. Mishra, “Neural Oscillators are Universal”.
[41] A. Momeni, B. Rahmani, M. Malléjac, P. del Hougne, and R. Fleury, “Backpropagation-free training of deep physical neural networks,” *Science*, vol. 382, no. 6676, pp. 1297–1303, Dec. 2023, doi: 10.1126/science.adi8474.
[42] K. He, X. Zhang, S. Ren, and J. Sun, “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*, Springer, 2016, pp. 630–645.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

[43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2012. Accessed: Mar. 30, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[45] S. Bai, J. Z. Kolter, and V. Koltun, “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.” arXiv, Apr. 19, 2018. doi: 10.48550/arXiv.1803.01271.

[46] A. Gu, K. Goel, and C. Re, “Efficiently Modeling Long Sequences with Structured State Spaces,” presented at the International Conference on Learning Representations, Jan. 2022. Accessed: Jun. 03, 2023. [Online]. Available: https://openreview.net/forum?id=uYLFoz1vlAC

[47] A. Gupta, A. Gu, and J. Berant, “Diagonal State Spaces are as Effective as Structured State Spaces,” *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 22982–22994, Dec. 2022.

[48] J. T. H. Smith, A. Warrington, and S. W. Linderman, “Simplified State Space Layers for Sequence Modeling.” arXiv, Mar. 03, 2023. doi: 10.48550/arXiv.2208.04933.

[49] A. Gu and T. Dao, “Mamba: Linear-Time Sequence Modeling with Selective State Spaces.” arXiv, Dec. 01, 2023. doi: 10.48550/arXiv.2312.00752.

[50] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[51] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.” arXiv, Dec. 11, 2014. Accessed: Sep. 05, 2022. [Online]. Available: http://arxiv.org/abs/1412.3555

[52] A. Voelker, I. Kajić, and C. Eliasmith, “Legendre Memory Units: Continuous-Time Representation in Recurrent Neural Networks,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019. Accessed: Mar. 06, 2024. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/952285b9b7e7a1be5aa7849f32ffff05-Abstract.html

[53] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, “Lagrangian Neural Networks.” arXiv, Jul. 30, 2020. doi: 10.48550/arXiv.2003.04630.

[54] S. Amari, “A Theory of Adaptive Pattern Classifiers,” *IEEE Trans. Electron. Comput.*, vol. EC-16, no. 3, pp. 299–307, Jun. 1967, doi: 10.1109/PGEC.1967.264666.

[55] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[56] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[57] M. Akrout, C. Wilson, P. Humphreys, T. Lillicrap, and D. B. Tweed, “Deep Learning without Weight Transport,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 976–984. Accessed: Apr. 19, 2020. [Online]. Available: http://papers.nips.cc/paper/8383-deep-learning-without-weight-transport.pdf

[58] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, “Backpropagation and the brain,” *Nat. Rev. Neurosci.*, pp. 1–12, Apr. 2020, doi: 10.1038/s41583-020-0277-3.

[59] G. Bellec, F. Scherr, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, “Biologically inspired alternatives to backpropagation through time for learning in recurrent neural nets,” *ArXiv190109049 Cs*, Feb. 2019, Accessed: May 19, 2020. [Online]. Available: http://arxiv.org/abs/1901.09049

[60] J. M. Murray, “Local online learning in recurrent networks with random feedback,” *eLife*, vol. 8, p. e43299, May 2019, doi: 10.7554/eLife.43299.

European Innovation Council

Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

[61] B. Scellier and Y. Bengio, “Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation,” *Front. Comput. Neurosci.*, vol. 11, 2017, Accessed: Nov. 16, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2017.00024

[62] A. Nø kland, “Direct Feedback Alignment Provides Learning in Deep Neural Networks,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., 2016, pp. 1037–1045. Accessed: May 18, 2020. [Online]. Available: http://papers.nips.cc/paper/6441-direct-feedback-alignment-provides-learning-in-deep-neural-networks.pdf

[63] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, “Random synaptic feedback weights support error backpropagation for deep learning,” *Nat. Commun.*, vol. 7, no. 1, Art. no. 1, Nov. 2016, doi: 10.1038/ncomms13276.

[64] G. Hinton, “The Forward-Forward Algorithm: Some Preliminary Investigations.” arXiv, Dec. 26, 2022. Accessed: Oct. 11, 2023. [Online]. Available: http://arxiv.org/abs/2212.13345

[65] A. Laborieux, M. Ernoult, B. Scellier, Y. Bengio, J. Grollier, and D. Querlioz, “Scaling Equilibrium Propagation to Deep ConvNets by Drastically Reducing Its Gradient Estimator Bias,” *Front. Neurosci.*, vol. 15, 2021, Accessed: Nov. 16, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.633674

[66] M. Lukoševičius, “A Practical Guide to Applying Echo State Networks,” in *Neural Networks: Tricks of the Trade: Second Edition*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., in Lecture Notes in Computer Science. , Berlin, Heidelberg: Springer, 2012, pp. 659–686. doi: 10.1007/978-3-642-35289-8_36.

[67] J.-J. E. Slotine, W. Li, and others, *Applied nonlinear control*, vol. 199. Prentice hall Englewood Cliffs, NJ, 1991.

[68] L. Kozachkov, M. M. Ennis, and J.-J. Slotine, “RNNs of RNNs: Recursive Construction of Stable Assemblies of Recurrent Neural Networks,” presented at the Advances in Neural Information Processing Systems, Oct. 2022. Accessed: Nov. 10, 2022. [Online]. Available: https://openreview.net/forum?id=2dgB38geVEU

[69] J. M. Cohen, S. Kaur, Y. Li, J. Z. Kolter, and A. Talwalkar, “Gradient descent on neural networks typically occurs at the edge of stability,” *ArXiv Prepr. ArXiv210300065*, 2021.

[70] M. C. Mackey and L. Glass, “Oscillation and Chaos in Physiological Control Systems,” *Science*, vol. 197, no. 4300, pp. 287–289, Jul. 1977, doi: 10.1126/science.267326.

[71] M. Lezcano-Casado and D. Martínez-Rubio, “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*, PMLR, May 2019, pp. 3794–3803. Accessed: Feb. 09, 2024. [Online]. Available: https://proceedings.mlr.press/v97/lezcano-casado19a.html

[72] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

[73] D. Calzolari, C. D. Santina, and A. Albu-Schäffer, “Exponential Convergence Rates of Nonlinear Mechanical Systems: The 1-DoF Case With Configuration-Dependent Inertia,” *IEEE Control Syst. Lett.*, vol. 5, no. 2, pp. 445–450, Apr. 2021, doi: 10.1109/LCSYS.2020.3004196.

[74] C. Gallicchio, “Euler state networks: Non-dissipative reservoir computing,” *ArXiv Prepr. ArXiv220309382*, 2022.

[75] A. Rodan and P. Tino, “Minimum complexity echo state network,” *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 131–144, 2010.

[76] Q. Liao and T. Poggio, “Bridging the gaps between residual learning, recurrent neural networks and visual cortex,” *ArXiv Prepr. ArXiv160403640*, 2016.

[77] H. Jaeger, “Short term memory in echo state networks. gmd-report 152,” in *GMD-German National Research Institute for Computer Science (2002), http://www. faculty. jacobs-university. de/hjaeger/pubs/STMEchoStatesTechRep. pdf*, Citeseer, 2002.

[78] C. Gallicchio, A. Micheli, and L. Pedrelli, “Deep reservoir computing: A critical experimental analysis,” *Neurocomputing*, vol. 268, pp. 87–99, 2017.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.

[79] N. Hammami and M. Sellam, “Tree distribution classifier for automatic spoken Arabic digit recognition,” in *2009 international conference for internet technology and secured transactions, (ICITST)*, 2009, pp. 1–4. doi: 10.1109/ICITST.2009.5402575.

[80] K. O. Chicaiza and M. E. Benalcázar, “A brain-computer interface for controlling IoT devices using EEG signals,” in *2021 IEEE fifth ecuador technical chapters meeting (ETCM)*, 2021, pp. 1–6. doi: 10.1109/ETCM53643.2021.9590711.

[81] J. R. Villar, P. Vergara, M. Menéndez, E. de la Cal, V. M. González, and J. Sedano, “Generalized models for the classification of abnormal movements in daily life and its applicability to epilepsy convulsion recognition,” *Int. J. Neural Syst.*, vol. 26, no. 06, p. 1650037, 2016, doi: 10.1142/S0129065716500374.

[82] H. Hamooni and A. Mueen, “Dual-domain hierarchical classification of phonetic time series,” in *2014 IEEE international conference on data mining*, 2014, pp. 160–169. doi: 10.1109/ICDM.2014.92.

[83] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, in ICML’17. Sydney, NSW, Australia: JMLR.org, Aug. 2017, pp. 1321–1330. Accessed: Nov. 27, 2023. [Online]. Available: https://proceedings.mlr.press/v70/guo17a.html

[84] X.-Y. Zhang, G.-S. Xie, X. Li, T. Mei, and C.-L. Liu, “A Survey on Learning to Reject,” *Proc. IEEE*, vol. 111, no. 2, pp. 185–215, Feb. 2023, doi: 10.1109/JPROC.2023.3238024.

[85] Z. Liu, Z. Wang, P. P. Liang, R. R. Salakhutdinov, L.-P. Morency, and M. Ueda, “Deep Gamblers: Learning to Abstain with Portfolio Theory,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2019. Accessed: Sep. 18, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/0c4b1eeb45c90b52bfb9d07943d855ab-Abstract.html

[86] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, “Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges,” *Inf. Fusion*, vol. 58, pp. 52–68, Jun. 2020, doi: 10.1016/j.inffus.2019.12.004.

[87] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. Hinton, “Regularizing Neural Networks by Penalizing Confident Output Distributions,” presented at the ICLR Workshop, 2017. Accessed: Nov. 27, 2023. [Online]. Available: https://openreview.net/forum?id=HyhbYrGYe

[88] G. M. van de Ven and A. S. Tolias, “Three scenarios for continual learning,” *Contin. Learn. Workshop NeurIPS*, 2018, Accessed: Feb. 05, 2020. [Online]. Available: http://arxiv.org/abs/1904.07734

[89] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental Classifier and Representation Learning,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010. Accessed: Jun. 15, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Rebuffi_iCaRL_Incremental_Classifier_CVPR_2017_paper.html

[90] F. L. Bauer and C. T. Fike, “Norms and exclusion theorems,” *Numer. Math.*, vol. 2, no. 1, pp. 137–141, 1960.

European Innovation Council Funded by the European Union

Funded by the European Union under Grant Agreement 101070918. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Innovation Council and SMEs Executive Agency (EISMEA). Neither the European Union nor the granting authority can be held responsible for them.