Minimal Euler State Networks

Anonymous Author

***Abstract*—The Euler State Network (EuSN) is a recently proposed Reservoir Computing (RC) model where the fixed state dynamics are obtained by discretizing an ordinary differential equation under stability and non-dissipative conditions. As a result, the model is able to effectively propagate input information over time, hugely improving the performance of RC models in tasks requiring long-term memory.**

Aiming at both reducing the complexity of the reservoir structure and further improving its efficiency, in this paper we propose a *minimalistic* EuSN architecture where the reservoir is constrained to a fixed bi-directional chain structure. We explore progressive simplifications where the recurrent and the input connections of the reservoir are fully described by a single weight value. While reducing the complexity of the base EuSN, the proposed minimal EuSN approach shows comparable performance on several tasks on time-series classification, thus offering considerable potential advantages, especially in embedded applications and physical implementations.

***Index Terms*—Reservoir Computing, Euler State Networks, Recurrent Neural Networks**

I. Introduction

Reservoir Computing (RC) [1], [2] is a powerful methodology for efficiently training Recurrent Neural Networks (RNNs). The approach basically exploits the architectural bias of stable neural dynamics, with a fixed (i.e., untrained) recurrent reservoir layer, and a trainable readout for the output computation. Over the years, RC has proven to be particularly effective in applications, in particular for the implementation of distributed learning functionalities in embedded systems [3], [4], and as a reference paradigm for neuromorphic hardware implementations of recurrent models [5], [6].

Crucial to the operating principles of RC is that the untrained reservoir needs to develop stable dynamics according to a global asymptotic stability property, that – in the context of the widely known Echo State Network (ESN) [7], [8] – is called the Echo State Property. According to this characterization, the reservoir develops stable dynamics, but at the same time it is constrained to have fading memory and a consequent state space structure that limits the capability of transmitting the input information through several time-steps.

To overcome these inherent limitations, a new RC approach has been recently proposed in which the reservoir is obtained by discretizing an Ordinary Differential Equation (ODE) under appropriate constraints of stability and non-dissipation [9]. As the resulting reservoir dynamics are computed as the forward Euler solution of an ODE, the model has been called the *Euler State Network* (EuSN). Differently from standard ESNs, EuSNs dynamics are neither lossy nor unstable, making it possible to effectively preserve input information over time, and thus being more suitable for addressing tasks involving long-term memorization. The concrete advantages of the EuSN approach on time-series classification tasks have been already pointed out in [9], in which it has been shown that EuSNs, significantly exceeding the accuracy of ESNs, are able to reach comparable levels of performance to fully trainable state-of-the-art RNN models, while still maintaining the striking RC efficiency advantage. Interestingly, a key aspect of the EuSN model is to make use of antisymmetric recurrent matrices. However, the study of the pattern of connectivity among the reservoir neurons in EuSNs remains until now unexplored.

In this paper, we go deeper in investigating the architectural organization of the reservoir layer in EuSNs. Taking inspiration from the works in [10], [11], we present progressive simplifications, both in the recurrent and in the input structure, that aim at lowering the overall reservoir complexity. Our analysis addresses the resulting efficiency benefits and, through experiments on several real-world datasets, comparatively assesses the efficacy of the proposed minimal EuSNs in applications.

The rest of this paper is organized as follows. In Section II, after a brief recap on ESN, we present the key concepts of EuSN processing. We introduce our architectural simplifications to EuSN in Section III. In Section IV we provide our experimental analysis. Finally, Section V concludes the paper.

II. Echo and Euler State Networks

Reservoir Computing (RC) [1], [2] indicates a class of recurrent neural models in which the recurrent hidden layer, i.e., the *reservoir*, is featured by fixed connections, and only a *readout* output layer is trained. In the following, within such a class, we consider the widely popular Echo State Network (ESN) [7], [8], and the recently introduced Euler State Network (EuSN) [9].

We fix our notation by considering N reservoir neurons and X input units. The reservoir state and the external input at time-step t are denoted, respectively, by $\mathbf{h}(t) \in \mathbb{R}^N$, and $\mathbf{x}(t) \in \mathbb{R}^H$.

Echo State Network (ESN) – Reservoir dynamics are ruled by the following state transition function, where we consider the general formulation with leaky-integrator neurons [12]:

$$\mathbf{h}(t) = (1-\alpha)\mathbf{h}(t-1) + \alpha \tanh\left(\mathbf{W}_h\mathbf{h}(t-1) + \mathbf{W}_x\mathbf{x}(t) + \mathbf{b}\right), \tag{1}$$

in which $\alpha \in (0, 1]$ is the leaking rate, $\mathbf{W}_h \in \mathbb{R}^{N\times N}$ is the recurrent weight matrix, $\mathbf{W}_x \in \mathbb{R}^{X\times H}$ is the input weight matrix, $\mathbf{b} \in \mathbb{R}^N$ is a bias vector, and $\tanh(\cdot)$ indicates the element-wise application of the hyperbolic tangent nonlinearity.

The weight values in $\mathbf{W}_h$, $\mathbf{W}_x$ and $\mathbf{b}$ are kept fixed after initialization in accordance with a global asymptotic stability

condition known as the Echo State Property (ESP) [13]. In practice, the recurrent weights in $\mathbf{W}_h$ are randomly drawn from a uniform distribution, e.g., over $(-1, 1)$, and then re-scaled in order to control the resulting spectral radius[1] $\rho(\mathbf{W}_h)$, typically limiting it to values smaller than 1. The value of $\rho(\mathbf{W}_h)$ is treated as a crucial hyper-parameter of the reservoir, and takes a major role in determining the nature of the developed state dynamics. Similarly, the input weight matrix and the bias vector are initialized randomly, e.g., from uniform distributions over $(-\omega_x, \omega_x)$ and $(-\omega_b, \omega_b)$, respectively. The values of ω_x and ω_b play the role of input and bias scaling coefficients, respectively, and are considered as further hyper-parameters of the model.

The network's architecture comprises a trainable dense read-out layer that, in the case of time-series classification tasks, is fed by the last reservoir state computed in correspondence to each input time-series.

As mentioned above, crucial to the operation of the ESN model is the ESP stability property that is used to constrain the reservoir dynamics. Essentially, when this property holds and the network is driven by a long input time-series, then the initial state conditions (or equivalently, input perturbations far in the past) are progressively forgotten, and the state encoding developed by the reservoir is stable. However, this characterization is related to the properties of fading memory and suffix-based Markovian organization of the reservoir state space (see, e.g., [14]–[16]), which make it difficult to effectively propagate information across several time-steps, thereby intrinsically limiting the efficacy of ESNs in tasks requiring long-term memorization abilities.

Euler State Network (EuSN) – Reservoir dynamics are obtained by discretizing an ODE subject to conditions of being stable and non-dissipative. Specifically, we start from the continuous-time dynamics given by:

$$\mathbf{h}'(t) = \tanh(\mathbf{W}_h\mathbf{h}(t) + \mathbf{W}_x\mathbf{x}(t) + \mathbf{b}), \tag{2}$$

and require that the Jacobian of the above eq. 2 has eigenvalues with ≈ 0 real parts. This, on the one hand, entails stable dynamics and, on the other, that the dynamical system is not lossy, making it able to effectively propagate the input information across the time-steps [17], [18]. A simple *architectural* way of meeting such a critical condition on the Jacobian consists in using an antisymmetric recurrent weight matrix, i.e., $\mathbf{W}_h = -\mathbf{W}_h^T$. In fact, in this case, the eigenvalues of both $\mathbf{W}_h$ and of the Jacobian of eq. 2 are purely imaginary [18]. Crucially, as noted in [9], such a property is not required to be learned from the data. Hence, taking an RC approach, we can consider all the weights in $\mathbf{W}_h$, $\mathbf{W}_x$ and $\mathbf{b}$ to be fixed, provided that the antisymmetric property of $\mathbf{W}_h$ is satisfied.

Discretizing the system in eq. 2 using the forward Euler method, we finally get the reservoir state transition equation:

$$\mathbf{h}(t) = \mathbf{h}(t-1) + \varepsilon \tanh\Big((\mathbf{W}_h - \gamma\mathbf{I})\mathbf{h}(t-1) + \mathbf{W}_x\mathbf{x}(t) + \mathbf{b}\Big), \tag{3}$$

[1]The maximum among the eigenvalues in modulus.

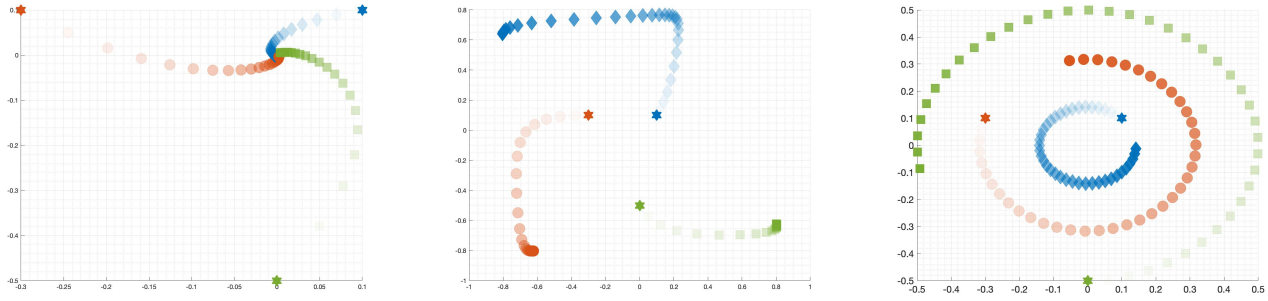
where $\mathbf{W}_h$, $\mathbf{W}_x$ and $\mathbf{b}$ are as in eq. 2, and $\mathbf{W}_h$ is anti-symmetric. Moreover, ε and γ are both small positive values that represent, respectively, the step size of the discretization, and the diffusion coefficient for stabilizing the forward Euler solution. Both ε and γ are treated as model's hyper-parameters.

Denoting by $\mathbf{J}(t)$ the Jacobian of the reservoir in eq. 3, we can derive that

$$\mathbf{J}(t) = \mathbf{I} + \varepsilon\mathbf{D}(t)\mathbf{W}_h - \varepsilon\gamma\mathbf{D}(t), \tag{4}$$

where $\mathbf{D}(t)$ is a diagonal matrix with entries given by the elements of $\tanh'\big((\mathbf{W}_h - \gamma\mathbf{I})\mathbf{h}(t-1) + \mathbf{W}_x\mathbf{x}(t) + \mathbf{b}\big)$. Relevantly, recalling that both ε and γ take small positive values, we can observe that the right-hand side of eq. 4 is dominated by the $\mathbf{I}$ term. This means that all the eigenvalues of $\mathbf{J}(t)$ are intrinsically ≈ 1, hence the local Lyapunov exponents of the reservoir [1], [19], [20] are all ≈ 0, and the system is biased towards edge-of-stability dynamics [21]. Moreover, the reservoir operation tend to preserve all the components of the state, while performing infinitesimal rotations due to the antisymmetric structure of $\mathbf{W}_h$. The resulting qualitatively different behavior between an ESN and an EuSN is illustrated in Fig. 1, which shows examples of state trajectories developed by 2-dimensional autonomous reservoirs (i.e., with zero input and bias) starting from different initial conditions around the origin. As it can be seen, for ESNs the origin is an attractor (Fig.1a) or a repeller (Fig.1b), depending on whether the ESP is valid or not. Instead, for EuSN (Figure 1c) the states rotate around the origin without being attracted or repelled, and the distances between the trajectories are conserved, reflecting the differences between the initial conditions. In other words, external input perturbations to the state are preserved without either exploding or vanishing, thereby enabling an effective propagation of the information over time.

From eq. 4, we can notice that the actual weight values of $\mathbf{W}_h$ only have a minor influence on the eigenvalues of $\mathbf{J}(t)$, hence on the dynamical behavior of the reservoir. In fact, the $\mathbf{W}_h$ contribution to $\mathbf{J}(t)$ is modulated by ε, which is small by definition, and by $\mathbf{D}(t)$, whose diagonal entries are in $(0, 1)$ due to the contractive properties of $\tanh$. Moreover, when the system is studied under common autonomous RC settings and linearized around origin (thereby setting $\mathbf{D}(t) = \mathbf{I}$ in eq. 4), we can see that the effective spectral radius of the reservoir is given by $\rho((1 - \varepsilon\gamma)\mathbf{I} + \varepsilon\mathbf{W}_h)$, which is naturally confined to a neighborhood of 1. At initialization, we can therefore avoid re-scaling the weights of $\mathbf{W}_h$ to control its spectral radius, in contrast to what happens with ESNs. As proposed in [9], we can limit ourselves to simply considering a scaling factor of the weights to take into account the modulation of the different contributions in eq. 3 (previous state, new input, and bias). $\mathbf{W}_x$ and $\mathbf{b}$ are initialized as described above for the case of ESNs. To initialize $\mathbf{W}_h$, we start with a random matrix $\mathbf{W}$ with values drawn from a uniform distribution over $(-\omega_r, \omega_r)$, and then set $\mathbf{W}_h = \mathbf{W} - \mathbf{W}^T$, which satisfies the antisymmetric constraint by construction. Here, ω_r denotes a recurrent weight scaling coefficient, which – together with ω_x and ω_b – is treated as a hyper-parameter.



(a) ESN with ESP. (b) ESN without ESP. (c) EuSN.

Fig. 1: Examples of 2-dimensional autonomous reservoir dynamics visualized around the origin for instances of: (a) ESN with ESP, (b) ESN without ESP, and (c) EuSN. The same three initial conditions are used in all the cases, shown as full stars with different colors: $[0.1, 0.1]^T$ in blue, $[-0.3, 0.1]^T$ in red, and $[0, -0.5]^T$ in green. Trajectories are shown by points of different shapes, with more transparent colors indicating earlier time-steps. (a): $\mathbf{W}_h = \left[[0.8, 0.1]^T, [-0.1, 0.8]^T\right]$. (b): $\mathbf{W}_h = \left[[1.3, 0.1]^T, [-0.1, 1.3]^T\right]$. (c): $\mathbf{W}_h = \left[[0, 1.4]^T, [-1.4, 0]^T\right]$, with $\varepsilon, \gamma = 0.001$.

The network is complemented by a readout layer whose settings are as in the case of ESN.

III. Minimal EuSN Architectures

In this section, we introduce our proposed EuSN architectures, specifically aimed at reducing the complexity of reservoir topology, network parametrization, and state computations. We progressively introduce two types of simplifications. First, we restrict the reservoir topology to have a deterministic bi-directional chain structure. Then, to further minimize the architectural design, we add a deterministic structure also to the input weights.

Simple Antisymmetric Reservoir – We start by noticing that, due to the antisymmetric constraint, the recurrent weight matrix of an N-dimensional EuSN reservoir is determined by $\frac{N(N-1)}{2}$ parameters, i.e., the number of entries below (or above) the main diagonal. Therefore, if we want to transmit the reservoir over a network (e.g., in embedded applications), we need to transfer a number of $\mathcal{O}(N^2)$ recurrent weights. Moreover, due to the dense reservoir structure, the time complexity of state computation in eq. 3 also scales as $\mathcal{O}(N^2)$.

To reduce the quadratic costs, we introduce a simple organization of the recurrent topology, which is both deterministic and sparse. Specifically, we consider a bi-directional chain reservoir architecture, such that every internal neuron i has only two outgoing recurrent connections, one to the preceding neuron $i-1$ and one to the subsequent neuron $i+1$, and two incoming connections from the same neurons. Moreover, we impose to use a single absolute weight value, say $\omega_r > 0$, that is shared among all neurons, and is used to describe the strength of all the recurrent connections. Finally, to respect the antisymmetric constraint, we assume that all the connections pointing to the preceding neuron have negative sign (i.e., $-\omega_r$), while those pointing to the subsequent neuron have positive sign (i.e., $+\omega_r$). The achieved architectural organization of the reservoir is illustrated in Fig. 2. Correspondingly,

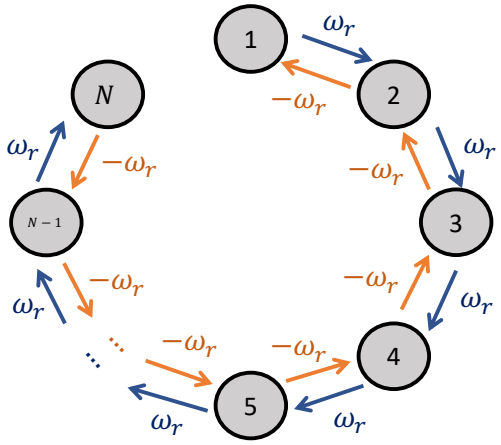


Fig. 2: Simple antisymmetric reservoir topology.

the recurrent weight matrix $\mathbf{W}_h$ has a peculiar antisymmetric structure with non-zero elements only on the sub and the super diagonals, as follows:

$$\mathbf{W}_h = \begin{pmatrix} 0 & -\omega_r & 0 & 0 & \dots & 0 \\ \omega_r & 0 & -\omega_r & 0 & \dots & 0 \\ 0 & \omega_r & 0 & -\omega_r & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \omega_r & 0 & -\omega_r \\ 0 & \dots & 0 & 0 & \omega_r & 0 \end{pmatrix}. \quad (5)$$

Notice that, given the fixed reservoir architecture, the whole recurrent weight matrix $\mathbf{W}_h$ can be described by just 1 parameter, i.e., ω_r, as opposed to $\mathcal{O}(N^2)$ as for the base EuSN. Moreover, given the value of the recurrent scaling ω_r, the recurrent weight matrix is deterministically constructed, without requiring randomization. In addition, due to the high degree of sparsity in $\mathbf{W}_h$, the complexity of state computation reduces from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

TABLE I: Complexity of EuSN compared to M-EuSN$_H$ and M-EuSN$_{H,X}$. We report the number of parameters required to describe the recurrent weight matrix (#Par. $\mathbf{W}_h$), the input weight matrix (#Par. $\mathbf{W}_x$), and the bias vector (#Par. $\mathbf{b}$). The last line gives the time complexity of the state update (eq. 3).

	EuSN	**M-EuSN$_H$**	**M-EuSN$_{H,X}$**
#Par. $\mathbf{W}_h$	$\mathcal{O}(N^2)$	1	1
#Par. $\mathbf{W}_x$	$\mathcal{O}(N\,X)$	1	1
#Par. $\mathbf{b}$	$\mathcal{O}(N)$	1	1
State Update	$\mathcal{O}(N^2)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$

As a side observation, it is also interesting to see that when $\mathbf{W}_h$ follows the organization in eq. 5, it can be described as a tridiagonal Toeplitz matrix with constantly zero diagonal values. As a consequence, its eigenvalues are symmetrically arranged on the imaginary axis, such that $\forall i = 1, \ldots, N$, $\text{eig}_i(\mathbf{W}_h) = \iota\, 2\omega_r \cos \frac{i\pi}{N+1}$.

In the following, we refer to a minimal EuSN architecture featured by the simple antisymmetric reservoir structure described above as M-EuSN$_H$.

Deterministic Input Connections – Inspired by [10], we consider a further architectural simplification, this time applied to the input connections. In particular, we consider all weights in $\mathbf{W}_x$ to have the same absolute value ω_x, and, similarly, all weights in $\mathbf{b}$ to have the same absolute value ω_b. The sign of each weight is determined by the decimal expansion of π, where a digit ≥ 5 implies a positive sign, and a digit < 5 implies a negative one. If this simplification is applied, the number of parameters needed to describe $\mathbf{W}_x$ and $\mathbf{b}$ reduces from $\mathcal{O}(N\,X)$ and $\mathcal{O}(N)$, respectively, to 1. Moreover, the initialization process for both $\mathbf{W}_x$ and $\mathbf{b}$ is completely deterministic.

In the following, when in addition to the simple reservoir structure in eq. 5 and Fig. 2, our minimal EuSN architecture is featured by deterministically constructed input and bias weights, as illustrated here, we denote it by M-EuSN$_{H,X}$.

To wrap up, by the introduced architectural constraints, the complexity of the EuSN is progressively reduced in M-EuSN$_H$ and M-EuSN$_{H,X}$, both in terms of parametrization and time complexity for state computation, as summarized in Table I. Finally, notice that in the rest of the paper, we will use M-EuSN when generally referring to any of the minimal EuSN architectures introduced in this section.

IV. Experiments

In this section, we describe the datasets, the experimental settings, and the numerical results achieved in our analysis.

Datasets - We have considered 8 datasets for time-series classification, from diverse application areas and featured by a diverse nature of the input features. The first 7 come from the UEA & UCR time-series classification repository [22], namely: Adiac [23], CharacterTrajectories [24], HandOutlines [25], Handwriting [26], Libras [27], ShapesAll [28], and SpokenArabicDigits [29]. As a further dataset, we have considered the Reuters newswire classification dataset from UCI [30]. This last dataset was considered in the parsed version which is publicly available online[2]. In addition, we have applied a pre-processing phase to represent each sentence by a sequence of 32-dimensional words embeddings[3].

TABLE II: Information on the used datasets for time-series classification. Specifically, we report the size of the training (#Seq Tr) and of the test set (#Seq Ts), the length of the times-series (Length), the number of input features (Feat.)), and the number of class labels (Classes).

Name	**#Seq Tr**	**#Seq Ts**	**Length**	**Feat.**	**Classes**
Adiac	390	391	176	1	37
CharacterTrajectories	1422	1436	182	3	20
HandOutlines	1000	370	2709	1	2
Handwriting	150	850	152	3	26
Libras	180	180	45	2	15
Reuters	8982	2246	200	32	46
ShapesAll	600	600	512	1	60
SpokenArabicDigits	6599	2199	93	13	10

A summary of the relevant information for each dataset is reported in Table II. For each dataset, we have used the original data separation into training and test, as indicated in Table II, with a further stratified splitting of the original training data into training (66%) and validation (33%) sets.

Experimental Settings - We assessed the performance achieved by the introduced M-EuSN models, comparatively with EuSN and standard ESN. For all models, the values of the hyper-parameters were chosen by model selection on the validation set, using KerasTuner[4] with Hyperband [31]. We explored reservoir configurations with ω_x and ω_b in $(0.01, 1.5)$ with linear sampling. For EuSN and M-EuSNs, we considered values of ω_r in $(0.01, 1.5)$, with linear sampling, and values of ε and γ in $(10^{-5}, 10^{-1})$ with logarithmic sampling. For ESN, we explored values of $\rho(\mathbf{W}_h)$ from $(0.01, 1.5)$, and of α from $(0.01, 1)$, with linear sampling. In all cases, the readout was implemented by a dense layer. Depending on the number of target classes, the readout contained units with sigmoid or softmax activation, and was trained using binary or categorical cross-entropy as loss function. We used the Adam optimizer for a maximum number of 500 epochs, and early stopping with patience 10. The value of the learning rate was treated as a further hyper-parameter, exploring values in the range $(10^{-5}, 10^{-1})$ with logarithmic sampling. We ran experiments

[2]https://keras.io/api/datasets/reuters/

[3]First, each sentence was represented by a sequence of words among the 10000 most frequent ones in the whole database, with truncation to the maximum length of 200. Then, word embeddings were obtained by training an MLP with a preliminary embedding layer with 32 units, a hidden layer containing 128 units with ReLU activation, and a final dense output layer with 46 units and softmax activation. The MLP architecture has been trained with RMSProp for 100 epochs, using categorical cross-entropy as loss function, and early stopping with patience 10 on a validation set containing the 33% of the original training data. The output of the embedding layer on the sentences in the dataset is then used to obtain the input features for our experiments.

[4]https://keras.io/keras_tuner/

with increasing reservoir sizes $N = 25, 50, 75, 100$. For every model and reservoir size, after model selection, 10 instances (i.e., random guesses) of the network with the best hyper-parameters configuration were trained on the training set and evaluated on the test set. The presented results were then obtained by averaging (and computing the std on) the test set accuracy over the 10 guesses. As an additional comparative step, we also performed a final model selection phase (on the validation set) to fine-tune the reservoir size individually for each model.

The code for our experiments is written in Keras, and is publicly available online[5].

Results - The achieved results are given in Fig. 3, which shows the test accuracy on the time-series classification tasks (different plots) obtained by the considered models at increasing values of the reservoir size (horizontal axis in each plot). Results clearly show the advantages of the EuSN approach over the standard ESN. In fact, all versions of EuSNs, including the M-EuSNs, generally outperform ESNs on all tasks and for all reservoir dimensions. Interestingly, in some cases (e.g., Adiac and HandOutlines) the performance of ESNs never approaches that one achievable by EuSNs, even for the maximum number of reservoir units considered. In other cases, when the performance of the ESNs grows up to approach that one reachable with the models within the EuSN class (e.g., Reuters, ShapesAll, and SpokenArabicDigits), the latter are able to obtain a performance in line with the highest achievable by the ESNs, already with the smallest reservoir sizes.

Relevantly, comparing the results of EuSN with those of M-EuSNs in Fig. 3, we can observe that the minimal variants achieve a comparable level of accuracy, showing only a minor decrease in performance. This is confirmed by the results in Table III, which show the test accuracy after the final step of model selection on the reservoir size. As it can be seen, the class of M-EuSN variants reaches a performance that is generally in line with that of the base EuSNs, in some cases even reaching the top accuracy level overall. Narrowing the focus to the minimal EuSN architectures, we can observe that in most cases M-EuSN$_H$ leads to a better performance than M-EuSN$_{H,X}$. We can speculate that this can be interpreted as an effect of the variability of input weights in M-EuSN$_H$, which is absent in M-EuSN$_{H,X}$.

Overall, our experimental analysis pointed out the efficacy of minimal EuSN architectures, which were able to reach a comparable level of accuracy to base EuSNs, at the same time granting the efficiency advantages presented in Section III, both in terms of parametrization and time complexity of state computations (see Table I).

V. Conclusions

In this paper, we delved into the architectural construction of the reservoir layer of an Euler State Network (EuSN), a recently introduced class of RC models whose dynamics are obtained by discretizing an ODE subject to stability and non-dissipation constraints.

Aiming at reducing the complexity of base EuSNs in terms of topology, parametrization and state computations, we have introduced two minimal EuSN architectures. The first one presents a simple antisymmetric reservoir in which the recurrent neurons are arranged according to a bi-directional chain. All the non-zero recurrent connections share the same absolute weight value, leading to a deterministic construction of the recurrent weight matrix. The second proposed architecture further extends the simplified structure to the input weight matrix and the bias vector, which are constructed in a deterministic fashion, using a single weight value and signs from a decimal expansion of the irrational number π. Overall, we were able to reduce the number of parameters required to fully describe each weight matrix involved in the reservoir state computation to just 1 number, leading to evident efficiency advantages. Experiments performed on several real-world time-series classification tasks highlighted that, despite the introduced simplifications, minimal EuSNs are able to achieve a comparable performance to base EuSNs, and a neat improvement in comparison to standard ESNs.

In conclusion, the study presented in this paper showed the potentialities of simple and efficient EuSN architectures, aiming to pave the way for future exploitations of this new paradigm, especially in embedded applications and implementations in neuromorphic hardware.

[5]A GitHub repository with the code will be made public after acceptance. A preliminary anonymized version is available at https://www.dropbox.com/sh/rico9vtvt3rhiju/AAC3TI7i6O0bSpvmPAP63JFha?dl=0.

References

[1] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[2] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[3] M. Dragone, G. Amato, D. Bacciu, S. Chessa, S. Coleman, M. Di Rocco, C. Gallicchio, C. Gennaro, H. Lozano, L. Maguire *et al.*, "A cognitive robotic ecology approach to self-configuring and evolving aal systems," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 269–280, 2015.

[4] D. Bacciu, P. Barsocchi, S. Chessa, C. Gallicchio, and A. Micheli, "An experimental characterization of reservoir computing in ambient assisted living applications," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1451–1464, 2014.

[5] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, 2019.

[6] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576, 2017.

[7] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks - with an erratum note," GMD - German National Research Institute for Computer Science, Tech. Rep., 2001.

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

[9] C. Gallicchio, "Reservoir computing by discretizing odes," in *Proceedings of ESANN*, 2021.

[10] A. Rodan and P. Tiňo, "Minimum complexity echo state network," *IEEE transactions on neural networks*, vol. 22, no. 1, pp. 131–144, 2010.

[11] T. Strauss, W. Wustlich, and R. Labahn, "Design strategies for weight matrices of echo state networks," *Neural computation*, vol. 24, no. 12, pp. 3246–3276, 2012.

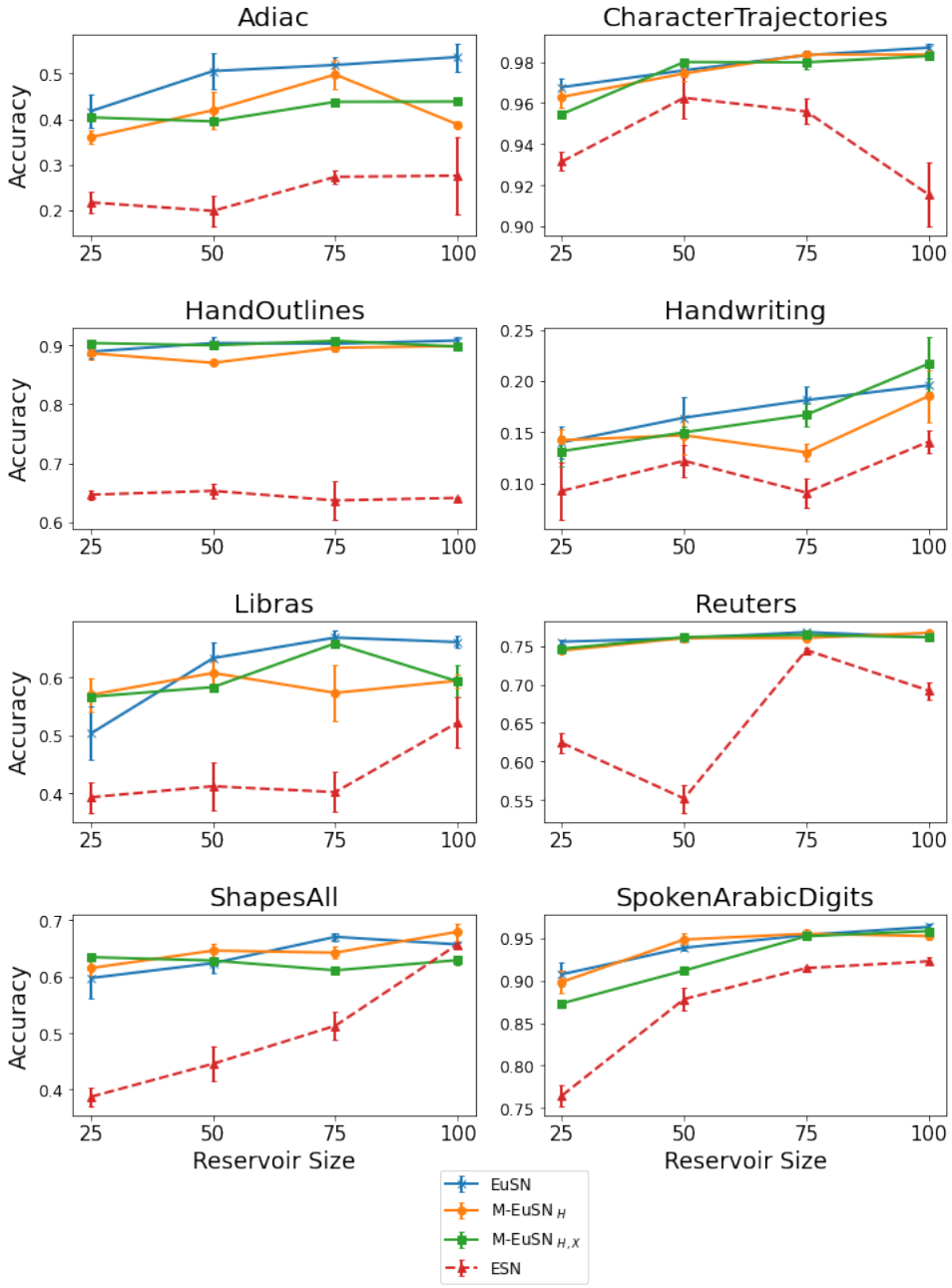


Fig. 3: Test set accuracy on the time-series classification tasks for increasing reservoir size. Results are reported for EuSN, M-EuSN variants, and ESN. M-EuSN$_H$ indicates minimal EuSN with simple antisymmetric reservoir. M-EuSN$_{H,X}$ indicates minimal EuSN with simple antisymmetric reservoir and deterministic input connections.

[12] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural networks*, vol. 20, no. 3, pp. 335–352, 2007.

[13] I. Yildiz, H. Jaeger, and S. Kiebel, "Re-visiting the echo state property," *Neural networks*, vol. 35, pp. 1–9, 2012.

[14] L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Neural Networks*, vol. 108, pp. 495–508, 2018.

[15] B. Hammer and P. Tiňo, "Recurrent neural networks with small weights implement definite memory machines," *Neural Computation*, vol. 15, no. 8, pp. 1897–1929, 2003.

[16] C. Gallicchio and A. Micheli, "Architectural and markovian factors of echo state networks," *Neural Networks*, vol. 24, no. 5, pp. 440–456, 2011.

[17] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks," *Inverse Problems*, vol. 34, no. 1, p. 014004, 2017.

[18] B. Chang, M. Chen, E. Haber, and E. H. Chi, "Antisymmetricrnn: A dynamical system view on recurrent neural networks," *arXiv preprint arXiv:1902.09689*, 2019.

[19] D. Verstraeten and B. Schrauwen, "On the quantification of dynamics in reservoir computing," in *International Conference on Artificial Neural Networks*. Springer, 2009, pp. 985–994.

[20] F. Bianchi, L. Livi, and C. Alippi, "Investigating echo state networks dynamics by means of recurrence analysis," *arXiv preprint arXiv:1601.07381*, pp. 1–25, 2016.

TABLE III: Classification accuracy on the test set achieved by EuSN, M-EuSN variants, and ESN. The reported results correspond to the best overall configuration selected on the validation set. M-EuSN$_H$ indicates minimal EuSN with simple antisymmetric reservoir. M-EuSN$_{H,X}$ indicates minimal EuSN with simple antisymmetric reservoir and deterministic input connections. The best result for each task is highlighted in bold font.

Task	**EuSN**	**M-EuSN$_H$**	**M-EuSN$_{H,X}$**	**ESN**
Adiac	$\mathbf{0.536}_{(\pm 0.031)}$	$0.498_{(\pm 0.032)}$	$0.438_{(\pm 0.005)}$	$0.273_{(\pm 0.015)}$
CharacterTrajectories	$0.983_{(\pm 0.002)}$	$\mathbf{0.984}_{(\pm 0.002)}$	$0.980_{(\pm 0.003)}$	$0.963_{(\pm 0.010)}$
HandOutlines	$0.903_{(\pm 0.011)}$	$0.899_{(\pm 0.005)}$	$\mathbf{0.908}_{(\pm 0.008)}$	$0.637_{(\pm 0.032)}$
Handwriting	$0.181_{(\pm 0.013)}$	$0.185_{(\pm 0.025)}$	$\mathbf{0.217}_{(\pm 0.027)}$	$0.122_{(\pm 0.016)}$
Libras	$\mathbf{0.669}_{(\pm 0.013)}$	$0.608_{(\pm 0.021)}$	$0.583_{(\pm 0.004)}$	$0.522_{(\pm 0.044)}$
Reuters	$0.761_{(\pm 0.001)}$	$\mathbf{0.767}_{(\pm 0.001)}$	$0.761_{(\pm 0.000)}$	$0.744_{(\pm 0.003)}$
ShapesAll	$0.657_{(\pm 0.005)}$	$\mathbf{0.680}_{(\pm 0.015)}$	$0.611_{(\pm 0.008)}$	$0.657_{(\pm 0.005)}$
SpokenArabicDigits	$\mathbf{0.963}_{(\pm 0.004)}$	$0.955_{(\pm 0.006)}$	$0.958_{(\pm 0.000)}$	$0.923_{(\pm 0.004)}$

[21] J. Boedecker, O. Obst, J. Lizier, N. Mayer, and M. Asada, "Information processing in echo state networks at the edge of chaos," *Theory in Biosciences*, vol. 131, no. 3, pp. 205–213, 2012.

[22] A. Bagnall, J. L. W. Vickers, and E. Keogh, "The uea & ucr time series classification repository," www.timeseriesclassification.com.

[23] A. C. Jalba, M. H. Wilkinson, and J. B. Roerdink, "Automatic segmentation of diatom images for classification," *Microscopy research and technique*, vol. 65, no. 1-2, pp. 72–85, 2004.

[24] B. H. Williams, M. Toussaint, and A. J. Storkey, "Extracting motion primitives from natural handwriting data," in *International Conference on Artificial Neural Networks*. Springer, 2006, pp. 634–643.

[25] L. M. Davis, B.-J. Theobald, J. Lines, A. Toms, and A. Bagnall, "On the segmentation and classification of hand radiographs," *International journal of neural systems*, vol. 22, no. 05, p. 1250020, 2012.

[26] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, and E. Keogh, "Generalizing dtw to the multi-dimensional case requires an adaptive approach," *Data mining and knowledge discovery*, vol. 31, no. 1, pp. 1–31, 2017.

[27] D. B. Dias, R. C. Madeo, T. Rocha, H. H. Biscaro, and S. M. Peres, "Hand movement recognition for brazilian sign language: a study using distance-based neural networks," in *2009 international joint conference on neural networks*. IEEE, 2009, pp. 697–704.

[28] L. J. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 1. IEEE, 2000, pp. 424–429.

[29] N. Hammami and M. Bedda, "Improved tree model for arabic speech recognition," in *2010 3rd International Conference on Computer Science and Information Technology*, vol. 5. IEEE, 2010, pp. 521–526.

[30] C. Apté, F. Damerau, and S. M. Weiss, "Automated learning of decision rules for text categorization," *ACM Transactions on Information Systems (TOIS)*, vol. 12, no. 3, pp. 233–251, 1994.

[31] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6765–6816, 2017.