Practical Comparisons of Reservoir Topology Performance and Input Distribution in Digital Reservoir Computers

Extended Abstract

Lewis Thelen
University of Cincinnati
Cincinnati, United States of America
thelenlr@mail.uc.edu

Vikram Ravindra
University of Cincinnati
Cincinnati, United States of America
ravindvm@ucmail.uc.edu

ABSTRACT

Reservoir computing has seen renewed interest as a paradigm in autonomous single and multi-agent systems due to its lightweight training and ability to utilize a wide variety of digital and physical substrates as reservoirs. Reservoir topology plays a significant role in determining the performance and dynamics of a given reservoir computer. Previous work comparing different reservoir topologies has ignored the interaction between the distribution of input nodes into the reservoir and reservoir topology. This study provides a significant contribution by comparing effects of two different input node distributions. Our results demonstrate that by concentrating input into an arbitrary region in the reservoir one is able to nearly double the linear memory performance of ring reservoir based reservoir computers.

KEYWORDS

Reservoir Computing, Topologies, Input Distribution

ACM Reference Format:
Lewis Thelen and Vikram Ravindra. 2025. Practical Comparisons of Reservoir Topology Performance and Input Distribution in Digital Reservoir Computers: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

Reservoir computing is a lightweight approach to training Recurrent Neural Networks first introduced in [6] and then unified in [12]. Reservoir computing has seen a recent resurgence of interest as a paradigm in autonomous systems due to its low computational cost to train and operate and its substrate agnosticism allowing it to be instantiated at the level of multi-agent swarms [9], individual autonomous agents [2] and perhaps most interestingly in sub-components of a single agent's body [10] [3].

In its most orthodox form reservoir computer (RC) operation can be expressed as:

$$x(t) = f(W_{in}\,u(t) + W_{res}\,x(t-1)) \quad (1)$$

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

Where $x(t) \in \mathbb{R}^{N\times 1}$ is the reservoir state vector at time t, $f()$ is a chosen activation function (typically $tanh()$) applied element-wise, $u(t)$ is the k-dimensional input vector at time t, $W_{in} \in \mathbb{R}^{N\times k}$ is the input weight matrix, $W_{res} \in \mathbb{R}^{N\times N}$ is the recurrently connected reservoir weight matrix, $W_{out} \in \mathbb{R}^{N\times m}$ is the trained output weight matrix where m is the dimensionality of the target and $y(t)$ is the RC output at time t.

Typically W_{in}, W_{res} and W_{out} are all randomly initialized and only W_{out} is ever trained with the classical approach being closed form linear regression to fit a matrix of reservoir state vectors to the target time series using W_{out}. For more details on RC training and best practices see [8].

In effect this approach relies on the temporal embedding of the input time series in the reservoir to be sufficiently high dimensional and non-linear such that the task at hand can be solved using only a single, typically linear readout layer in the form of W_{out}. Consequently the topology of W_{res} has a pronounced effect on the performance of the RC as a whole. Work such as [11], [4] and [7] have all dealt with the impacts of reservoir topology on RC performance, but these have not taken into consideration the constraints of real-world applications. Additionally, impacts of the distribution of input nodes into the reservoir given a specific topology have likewise been ignored. *We aim to advance the literature by comparing performance across RCs equipped with the generic randomly distributed input nodes and a selective input condition in which input nodes are concentrated into a particular region of the reservoir. Our results reveal that a selective input condition combined with a ring topology significantly boost linear memory performance of the RC relative to an identical RC equipped with the generic input condition.*

2 METHODS

Random, ring, lattice and small world topologies were tested. All reservoirs were of size N where $N \in \{25,50,100,200\}$ except in the case of the small world reservoirs where N was restricted to $\{100,200\}$. Reservoir weights were assigned uniformly at random from [-1, 1] and scaled such that spectral radius of W_{res} is ρ which is common practice in order to ensure reservoir stability [8]. In all cases $\rho = 0.9$

With random reservoirs each node is connected with each other reservoir node with probability $p \in \{0.05,0.1,0.2,1.0\}$. In the cases of $N = 25$ and $N = 50$ p was limited to $\{0.2,1.0\}$ and $\{0.1,0.2,1.0\}$ respectively in order to maintain a minimum degree of network connectivity. Ring reservoirs were constructed such that given the nodes of the reservoir in a continuous 1-D array, each node was connected to itself as well as its r nearest neighbors to the left and r nearest neighbors to the right where $r \in \{1,2,4,8\}$. Lattice reservoirs were built similarly to ring reservoirs except the nodes were treated as being arranged in a 2-D rather than 1-D array and the value r was varied such that $r \in \{1,2,4\}$. Small world reservoirs were constructed

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

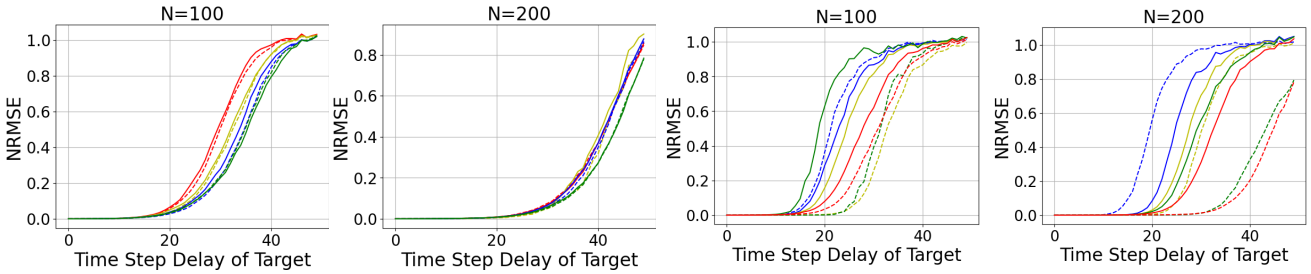

Figure 1: Random Reservoir and Ring Reservoir Performance on Generic Digit Recall Task - Solid lines correspond to RCs using generic input, dashed lines to selective input. In the case of random reservoirs on the left red corresponds to $p = 0.05$, yellow to $p = 0.1$, blue to $p = 0.2$ and green to $p = 1.0$. In the case of ring reservoirs on the right blue corresponds to $r = 1$, yellow to $r = 2$, green to $r = 4$ and red to $r = 8$.

by connected unconnected nodes in ring reservoirs with probability $p \in \{0.5, 0.1, 0.2\}$. Small world reservoirs were constructed for each of the ringer reservoir configurations. W_{in} was generated using two distinct methods across all reservoir configurations. In the first, generic condition each entry in W_{in} was assigned a non-zero weight w with probability p where $w \in [-0.1, 0.1]$ and $p = 0.1$. In the second, selective condition the first m entries in W_{in} were each assigned a non-zero weight w where $w \in [-0.1, 0.1]$ and $m = 0.1N$.

Reservoirs were tested using three benchmark tasks: the generic digit recall task (GDRT), the NARMA-10 task and the Santa Fe Laser prediction task. The GDRT is a common means of testing an RC's linear memory capacity [13]. It consists of testing an RC's ability to reproduce a random signal following an arbitrary time lag k. The NARMA-10 task, first introduced in [1], requires the RC to produce a target signal $y(t)$ derived using a standard polynomial $P(x)$ and the prior 10 time steps of $x(t)$ where $x(t)$ is pulled uniformly at random from [0,0.5]. The task assesses the RCs ability to recall prior inputs and perform non-linear operations on them. Lastly, the Santa Fe Laser prediction task, first introduced in [5], utilizes a chaotic time series consisting of the precise recording of the firing times of an infrared laser given in milliseconds. The task itself requires the RC to predict the value of the time series at $t+n$ given the time series up through t, for increasing values of n. Given the relatively small size of the reservoirs tested a value of $n = 1$ was only used. In the case of the generic digit recall and NARMA-10 tasks a training set of 2000 and a test set of 1000 data points were used. In the case of the Santa Fe Laser prediction task a training set of 6000 and a test set of 3000 data points were used.

3 RESULTS

For each task 15 different RCs of each of the possible reservoir configurations were generated, and their error was recorded as Normalized Root Mean Square Error (NRMSE). For the sake of concision only the results of the random and ring reservoir RCs are reported in depth.

With the GDRT random reservoirs noticeably improve in terms of memory depth as the size of the reservoir increases, but demonstrate no significant change between the two input conditions as shown in 1. Maximal performance by random reservoir RCs is achieved by the fully connected reservoir at $N = 200$ with memory resolution noticeably diverging near $k = 25$. As seen in 1 ring reservoir RCs show a remarkable robustness in performance. In particular with larger reservoirs of size $N \in \{100, 200\}$ performance responds primarily to input condition with the selective input condition far outperforming the generic input condition. Ring reservoir RCs see their best performance of divergence at $k = 35$ with the selective input condition at $N = 200$ and $r = 8$. Lattice and small world reservoirs see only slight difference in performance between input conditions and do not achieve the depth of the random or selective input ring reservoirs.

With the NARMA-10 task random reservoirs similarly do not respond to input condition and instead increase in performance with reservoir size. Random reservoirs see optimal performance with of an NRMSE of 0.271 at $N = 200$ with selective input and $p = 0.2$. Ring reservoirs again show the greatest response to input condition, but with the NARMA-10 task the generic input condition far outperforms the selective input condition. Optimal performance for ring reservoirs is achieved with the generic input condition at $N = 200$ and $r = 8$ with an NRMSE of 0.269. The selective input condition on the other hand only achieves an optimal NRMSE of 0.371 also with $N = 200$ and $r = 8$. Again lattice and small world reservoirs see only a slight difference in performance between input condition and overall see comparable performance between the random and ring reservoirs.

Lastly, with the Santa Fe Laser Task we see counterintuitive results in which RC performance seems to degrade as reservoir size increases. With random reservoir RCs, the generic input reservoirs achieve their best performance of an NRMSE of 0.577 at $N = 25$ with $p = 0.2$ and selective input reservoirs achieve their best performance of an NRMSE of 0.548 $N = 25$ and $p = 1.0$. With ring reservoirs selective input RCs see their best performance of an NRMSE of 0.505 at $N = 50$ and $r = 4$ and generic input RCs see their best performance of an NRMSE of 0.520 at $N = 25$ and $r = 4$. The lattice and small world reservoirs perform in a near identical manner.

4 DISCUSSION

The results from above show that utilizing a concentrated input distribution with an ring topology significantly increases the RCs linear memory performance to achieve best in class recall depth. At the same time, this increased linear recall seems to come at the cost of a marked decrease in the RCs ability to perform non-linear information processing. These impacts are not seen with RCs using a random reservoir topology. We attribute this to the far shorter path length found in random reservoirs as well as their less ordered signal propagation.

REFERENCES

[1] A. Atiya and A. Parlos. 2000. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks* (2000).

[2] Paolo Baldini. [n.d.]. Reservoir Computing in robotics: a review. ([n. d.]). Preprint at 10.48550/arXiv.2206.11222.

[3] Priyanka Bhovad and Suyi Li. 2021. Physical Reservoir Computing with Origami and its Application to Robotic Crawling. *Nature Scientific Reports* (2021).

[4] Matthew Dale et al. 2020. Reservoir computing quality: connectivity and topology. *Natural Computing* (2020).

[5] Neil Gershenfeld and Andreas Weigend. 1994. The Future of Time Series. In *Time Series Prediction: Forecasting the Future and Understanding the Past.*

[6] Herbert Jaeger. 2001. The" echo state" approach to analysing and training recurrent neural networks-with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148 (01 2001).

[7] Yuji Kawai et al. 2019. A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks* (2019).

[8] Mantas Lukosevicius. 2012. A Practical Guide to Applying Echo State Networks. In *Neural Networks: Tricks of the Trade*. Springer.

[9] Thomas Lymburn et al. 2021. Reservoir computing with swarms. *Chaos* (2021).

[10] Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. 2015. Information Processing via Physical Soft Body. *Nature Scientific Reports* (2015).

[11] Ali Rodan and Peter Tino. 2011. Minimum Complexity Echo State Network. *IEEE Transactions on Neural Networks* (2011).

[12] D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. 2007. An experimental unification of reservoir computing methods. *Neural Networks* (2007).

[13] Chester Wringe, Martin Trefzer, and Susan Stepney. [n.d.]. Reservoir Computing Benchmarks: a review, a taxonomy, some best practices. ([n. d.]). Preprint available at https://arxiv.org/abs/2405.06561.