Lattice reservoirs: symmetry breaking and information flow in physical reservoir computing

Thomas Aven, Johannes H. Jensen and Gunnar Tufte

Department of Computer Science,
Norwegian University of Science and Technology, Trondheim, Norway
johannes.jensen@ntnu.no

Abstract

Reservoir computing has become immensely popular for exploiting physical systems for computation. A multitude of physical reservoirs have been demonstrated, ranging from assembles of nanomagnets to living cultures of neurons. Unlike abstract software reservoirs, physical reservoirs are subject to spatial constraints which restricts the possible reservoir topologies. Here, we investigate lattice reservoirs, where nodes are placed on a regular lattice which defines the reservoir topology and its weights. Despite their simple regular structure, lattice reservoirs perform surprisingly well, in some cases outcompeting classical Echo State Networks. A key finding is the need for directed edges to facilitate information flow within the reservoir, highlighting the importance of symmetry breaking in physical reservoirs. We take advantage of the spatial nature of lattice reservoirs to discover key computational structures within, revealing what these reservoirs are actually doing. Lattice reservoirs bridge the gap between physics and computation, providing invaluable insight for the design and understanding of physical reservoirs.

Introduction

Reservoir computing (RC) has become a popular framework for exploiting physical systems for computation. A myriad of physical reservoirs have been proposed, ranging from electronic, photonic, magnetic, mechanical and even biological systems (Tanaka et al., 2018). Since computation occurs directly in the physical domain, physical reservoir computing has the potential to be extremely efficient.

Unlike abstract software reservoirs, physical reservoirs are subject to the physical constraints of the real world. Physical reservoirs must necessarily be embedded in physical space, which effectively places limits on the connections (edges) between computational elements (nodes). These limits in turn affect the reservoir topology, as well as the possible locations of input and output signals. Spatial constraints are of major concern when connections are realized directly through physical interactions, but also relevant for systems with explicit wiring.

The fundamental interactions in nature depends on the displacement between particles (Braibant et al., 2012). Consequently, a wide range of physical systems are *spatially coupled*, where topology follows directly from the spatial arrangement of the constituent parts. At the material level, coupling arises from interactions defined by the atomic structure, which for many materials is a periodic lattice structure (Kittel, 2005). Spatial coupling is also prevalent at higher levels of abstraction, e.g., dipolar coupling in magnetic systems, optical modes in nonlinear media, and chemical reaction diffusion systems. While some physical systems, e.g., electric circuits, may pose fewer limitations on topology, long-range connections are always associated with higher wiring cost and complexity. Hence, the topology of physical reservoirs is often much more constrained than their abstract software relatives.

The relationship between physical constraints and reservoir performance is thus a pertinent topic. While there is a wealth of research on specific physical reservoirs, an abstract treatment of the subject could shed light on favorable topological properties of physical reservoirs in general. Such insight is invaluable when considering the viability of physical reservoirs, as well as for tuning their intrinsic behavior.

In this work, we explore lattice reservoirs, where nodes are placed on a regular lattice which defines a fixed reservoir topology and its weights. To reduce the wiring complexity at the input layer, we explore the effect of using a global input signal which is broadcast uniformly to all the nodes in the reservoir.

Our results show that lattice reservoirs perform surprisingly well, in some cases outcompeting classical Echo State Networks (ESNs). A key finding is that reservoir edges need to be directed to facilitate the flow of information within the reservoir. However, only minor symmetry breaking may be necessary, given appropriate topology. Furthermore, we take advantage of the spatial embedding of lattice reservoirs to analyse and learn what the reservoirs are actually *doing*.

Background

Classical RC was initially introduced as an efficient way to harness the computational power of recurrent neural networks (RNNs) (Jaeger, 2001; Maass et al., 2002). Utilizing a random RNN as a "reservoir", input sequences are projected into a high-dimensional space, incorporating temporal information in the state of the RNN. A readout layer is then trained using supervised linear regression, to produce some desired output based on the reservoir state. The most popular variant of RC is the Echo State Network, where the reservoir consists of a random network of non-linear neurons with continuous activations.

Interestingly, there is no need for the reservoir to be a neural network at all – any high-dimensional, driven system exhibiting complex dynamic behavior can be used (Schrauwen, 2007). Because of this, RC has become an immensely popular method for exploiting physical systems for computation. A myriad of physical reservoirs have been proposed, ranging from memristive (Moon et al., 2019), optical (Paquot et al., 2012), photonic (Vandoorne et al., 2014), magnetic (Jensen and Tufte, 2020), mechanical (Coulombe et al., 2017) and even living cultures of neurons (Dockendorf et al., 2009). See Tanaka et al. (2018) for a review.

Many physical systems can be described by lattice models from computational physics (Lavis, 2015). There are numerous examples of physical reservoirs confined to a lattice, e.g., coupled nanomagnets (Jensen and Tufte, 2020), magnetic nanorings (Vidamour et al., 2023), spin-torque oscillator arrays (Tsunegi et al., 2019), skyrmions (Pinna et al., 2020), spin waves (Papp et al., 2021) and a general Ginzburg-Landau model (Opala et al., 2019). In these spatially coupled systems, the reservoir topology is directly determined by the lattice structure, e.g., the dipolar coupling between nanomagnets results in a topology with strong local interactions. As such, the reservoir connections come "for free", without the need for explicit wiring. However, as mentioned earlier, the spatial constraints of the lattice places restrictions on the possible reservoir topologies.

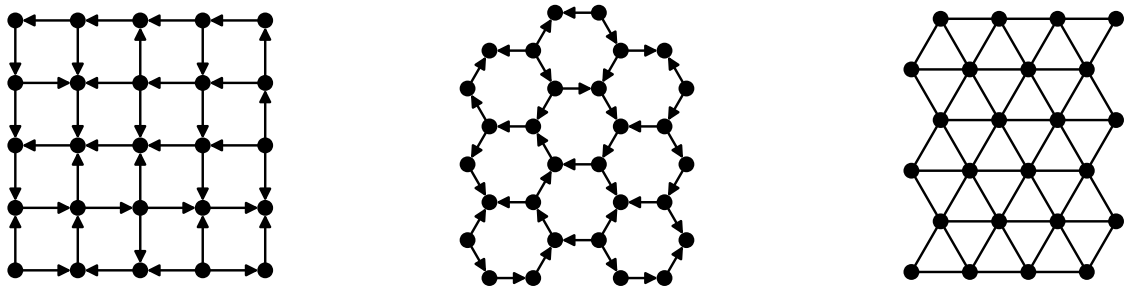
While other physical systems may pose fewer limitations on connectivity, long-range connections are always associated with higher wiring cost and complexity. For example, electric circuits allow arbitrary connections to be realized in principle, but short (local) connections are superior in terms of cost, complexity and signal propagation. For this reason, it is often argued that the connectivity in neural systems is primarily local (Mead, 1990). Understanding the computational properties of spatially coupled systems is thus relevant for a wide variety of physical reservoirs.

Physical constraints are also highly relevant at the interface level (input/output layer) of physical reservoirs. Abstract software reservoirs typically employ random input weights, which diversifies the input throughout the reservoir. In a physical reservoir, this can be costly and/or difficult to realize, as it requires the ability to perturb the physical system locally. An electronic reservoir, for example, would require separate input circuitry and wiring for each reservoir node. Magnetic reservoirs would require even more physical infrastructure in order to generate locally varying magnetic fields or similar.

There is growing evidence that reservoir topology is an important factor on performance. Small-world, scale-free networks have been found to outperform random networks in a range of tasks (Deng and Zhang, 2007; Kawai et al., 2019; Manevitz and Hazan, 2010). Modular topologies were shown to outperform random networks in terms of memory capacity (Rodriguez et al., 2019). Orthogonalization of the reservoir weights can increase memory capacity, even approaching the upper bound (Farkaš et al., 2016). Clearly, reservoir topology can significantly affect the computational properties of reservoirs.

Regular reservoir topologies are of particular interest, as they readily map to physical implementations. Perhaps the simplest example are cyclic reservoirs, i.e., ring topologies, which were shown to perform comparably to the classical ESN (Rodan and Tino, 2011). The topology of cyclic reservoirs is extremely simple, with few adjustable parameters. Cyclic reservoirs extended with regular jumps consistently outperform ESNs (Rodan and Tiňo, 2012). In the context of deep reservoirs, structured schemes such as multiple rings obtained the best performances (Gallicchio and Micheli, 2019). Although topologically simple, cyclic reservoirs do not readily map to spatially coupled physical substrates.

When it comes to other regular topologies, the reservoir computing literature is sparse. Dale et al. (2019, 2020) presents one of the few studies, comparing ring, torus, square lattice and fully-connected networks. While fully-connected reservoirs were found to exhibit the most diverse range of reservoir behaviors, they also have considerably more adjustable parameters than the regular topologies. Torus and lattice reservoirs closely follow the quality of fully-connected reservoirs, with ring topology having the least diverse behavior. Notably, Dale et al. (2019) report that undirected networks generally perform worse than directed ones (regardless of topology). While the topologies in these studies were fixed, the reservoir weights were adjusted through evolutionary search. The lattice reservoirs in our study are much simpler, having both a fixed topology *and* constant weights. Another difference is the lack of self-connections in our lattice reservoirs (nodes are memoryless).



(a) **(b)** **(c)**

Figure 1: Lattice reservoirs with three topologies: (a) square (directed), (b) hexagonal (directed), and (c) triangular (undirected). For the directed topologies, each edge direction is decided by an unbiased coin toss. For undirected topologies, edge directions are symmetrical.

Methods

We consider classical discrete-time ESNs with N internal network nodes, a single input, and a single output node. At time-step t, the reservoir is defined by its input, internal, and output units, denoted by $\mathbf{u}(t)$, $\mathbf{x}(t)$, and $\mathbf{y}(t)$, respectively. The reservoir is defined by three weight matrices: input weights $\mathbf{W}^{in}$, reservoir weights $\mathbf{W}^{res}$, and output weights $\mathbf{W}^{out}$. The initial state of the reservoir is $\mathbf{x}(0) = \mathbf{0}$, and is evolved according to

$$\mathbf{x}(t+1) = \tanh(\mathbf{W}^{res}\mathbf{x}(t) + \mathbf{W}^{in}\mathbf{u}(t)). \quad (1)$$

Note that this is a slightly modified version of the original ESN update equation, where the input stems from the previous time step $u(t)$ (instead of $u(t+1)$). Our modification is arguably more physical, since information transfer is not instantaneous (Stepney, 2021). The output of the reservoir is given by

$$\mathbf{y}(t) = \mathbf{W}^{out}\mathbf{x}(t). \quad (2)$$

Following standard RC practice (Lukoševičius, 2012), input weights $\mathbf{W}^{in}$ are generated as a random matrix with i.i.d. entries in the interval $[-0.5, 0.5]$. $\mathbf{W}^{in}$ is then rescaled with an input scaling $\iota = 0.1$. Reservoir weights $\mathbf{W}^{res}$ are generated similarly, but with a 10% node connectivity. $\mathbf{W}^{res}$ is rescaled to have spectral radius $\rho(\mathbf{W}^{res}) = 0.9$.

Motivated by the wiring cost and complexity associated with local physical input, we also explore a global input scheme, where all nodes see the same input, i.e., $\mathbf{W}^{in} = \iota$. The output weights $\mathbf{W}^{out}$ are adapted using single value decomposition.

Lattice reservoirs

Lattice reservoirs are created by replacing the reservoir weights $\mathbf{W}^{res}$ of the ESN with the adjacency matrix of a given lattice. Hence, nodes in lattice reservoirs are only connected to their nearest neighbors on the lattice, with identical weights and no self-connections. For this study, we consider square, hexagonal and triangular lattices in Euclidean space, as illustrated in Fig. 1. Lattices have equal width and height, but are compared w.r.t. their total number of nodes N.

Note that the adjacency matrix of the lattice is symmetric, hence the resulting reservoir weights are effectively undirected. We obtain *directed* lattice reservoirs by making a fraction of the edges directed, with edge direction decided by an unbiased coin toss.

Finally, the reservoir weights $\mathbf{W}^{res}$ are scaled to have a spectral radius of 0.9, analogous to scaling the coupling, or spacing, between nodes in a physical system. Beyond this, lattice reservoirs remain the same as the classical ESN.

Shrinking and growing reservoirs

The spatial embedding of lattice reservoirs presents some unique opportunities for analysis and understanding. Since every node is placed on a regular lattice, the structure of the reservoirs can be readily visualized. Furthermore, the set of reservoir topologies is finite, as all reservoir nodes must be placed on the lattice. We exploit these properties to analyze the internals of lattice reservoirs.

Towards that end, we shrink and grow lattice reservoirs in a greedy deterministic fashion. Shrinking proceeds by removal of nodes from the lattice in an incremental manner, where each iteration removes the node that results in the lowest increase in benchmark error (see Fig. 5). In other words, the performance impact of all possible node removals is evaluated, after which the node causing the smallest increase in error is removed.

Lattice reservoirs can be grown in a similar manner. There is a finite amount of free positions on the lattice where new nodes may be inserted into the network, and for each such position there is only a handful of ways to direct the incident edges. By evaluating all possibilities (including all edge directions), we attach the node that causes the largest decrease in benchmark error (see Fig. 8).

Throughout the shrinking or growth process, $\mathbf{W}^{res}$ is rescaled to have spectral radius of 0.9. Both shrinking and growing of lattice reservoirs thus follow a simplistic, exhaustive approach. We can apply the same method to shrink classical ESNs. However, since ESNs have no topological constraints, they cannot readily be grown with the same exhaustive approach.

Benchmarks

For this study, the NARMA-10 benchmark is used as the primary evaluation of the reservoir performance (Atiya and Parlos, 2000). Given an input stream $u(t)$ i.i.d. drawn uniformly from the interval $[0, 0.5]$, the goal of the reservoir is to produce an output given by

$$\begin{aligned} y(t) =& \alpha y(t-1) + \beta y(t-1) \sum_{i=1}^{10} y(t-i) \\ &+ \gamma u(t-1) u(t-10) + \delta, \end{aligned} \tag{3}$$

with parameters $\alpha = 0.3$, $\beta = 0.05$, $\gamma = 1.5$ and $\delta = 0.1$. The time series input is split into a training and test set, with $L_{train} = 2000$ and $L_{test} = 3000$. Performance is measured using the normalized root mean square error (NRMSE).

For all experiments, the first 200 time steps are discarded to provide a washout of the initial reservoir state. Reported results are the average of 20 randomizations of each reservoir model. This sample size was found to be appropriate to pinpoint definite trends in the results. Standard deviations for all experiments, as well as the code for reproducing each experiment, are available online[1].

[1] https://github.com/thomaav/reservoir-computing

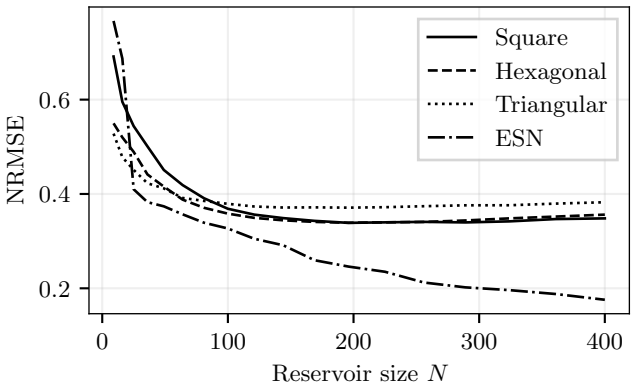


Figure 2: Undirected lattice reservoir performance: Square, hexagonal, and triangular topologies, compared to classical ESNs. The plots show average NARMA-10 NRMSE (lower is better) as a function of reservoir size N.

Results & discussion

Undirected lattice reservoirs

First, we consider lattice reservoirs with undirected edges. Fig. 2 shows how reservoir performance scales with reservoir size N. From the plot, it is clear that restricting reservoir topologies to regular lattice structures results in a significant performance penalty, compared to classical ESNs. Although small lattice reservoirs display somewhat competitive results, performance stagnates as reservoir size is increased beyond $N = 100$. In contrast, classical ESNs scale well with size, where NRMSE is consistently reduced by adding more reservoir nodes. For example, square lattice reservoirs of size $N = 400$ obtain a NRMSE of around 0.35, while ESNs of the same size achieve a NRMSE of 0.18.

Reservoir performance appears to be very consistent across the three types of lattice. It seems that it is the overall lattice structure that is important, not the particular type of lattice. Although the different lattices effectively decide the number of incident edges per node, the effect on performance is negligible.

It is somewhat surprising that undirected lattice reservoirs perform as well as they do. One might expect that very little reservoir diversity can result from such a uniform network topology (all reservoir weights are identical and all edges are undirected). However, there are two main sources of diversity. First, the random input weights enables nodes to see different variations of the input signal. Secondly, the edges of the lattice play an important role, as the edge nodes have fewer neighbors and hence different activations compared to the rest of the lattice. Introducing global input (uniform input weights) results in significant performance penal-

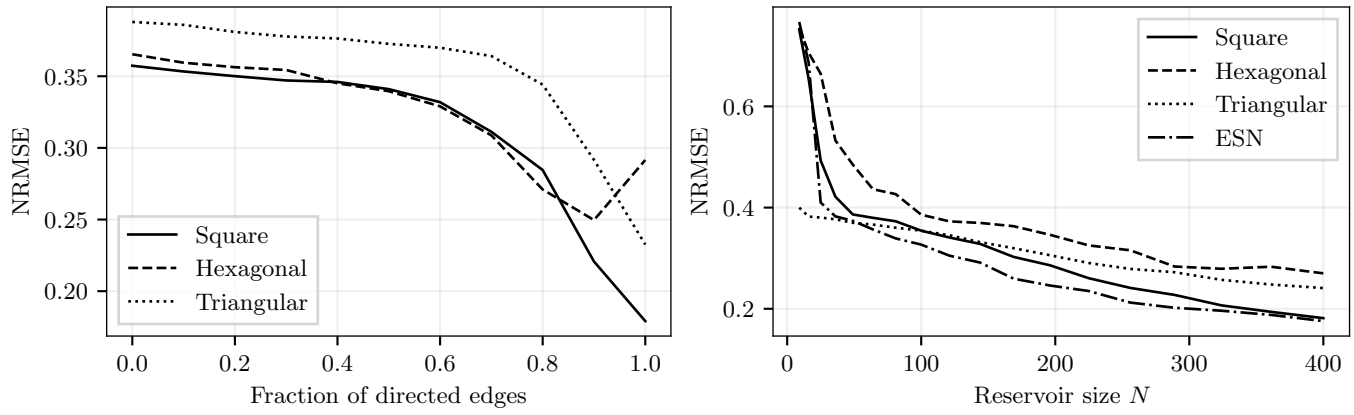


Figure 3: Directed lattice reservoir performance. The plots show average NARMA-10 NRMSE (a) as a function of the fraction of directed edges ($N = 400$), (b) as a function of reservoir size N, compared to classical ESNs.

ties. Interestingly, performance is largely unaffected by periodic boundaries alone (with local input). However, introducing both global input and periodic boundaries is completely detrimental. Although undirected lattice reservoirs perform surprisingly well, they do not measure up to classical ESNs.

Directed lattice reservoirs

Next, we investigate the effect of introducing directed edges to the lattice reservoirs. Fig. 3a shows the effect of making a fraction of the reservoir edges directed, for reservoirs of fixed size $N = 400$. As can be seen, reservoirs generally show reduced error with increasing fractions of directed edges. An apparent exception is hexagonal reservoirs, where fully directed reservoirs results in a performance degradation. Square and triangular reservoirs do not exhibit this behavior, likely because each node has more incident edges, which in turn reduces the likelihood of poor topologies.

Fig. 3b plots performance of fully directed lattice reservoirs as function of reservoir size N. As can be seen, fully directed reservoirs exhibit clear reductions in error with increased reservoir size. This is in stark contrast to their undirected counterparts, which only perform marginally better as reservoir size is increased (compare Figs. 2 and 3b). In fact, the performance of directed lattice reservoirs approaches that of the classical ESN, spearheaded by the square topology. Standard deviations are consistently low across results[1], which indicates that our method of generating directed edges reliably creates good reservoirs – one does not need to "get lucky" with directions.

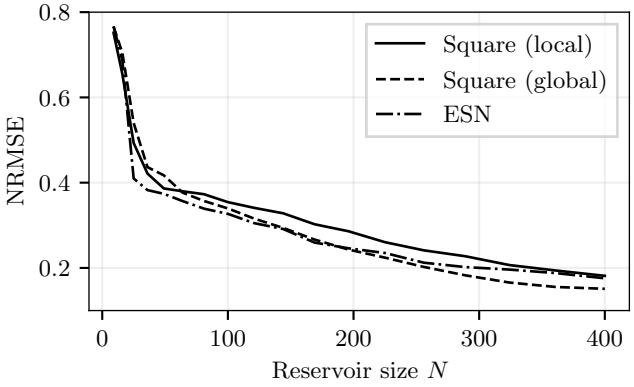


Figure 4: Global input performance. The plots show average NARMA-10 NRMSE for square lattice reservoirs with local and global input schemes, compared to classical ESNs (with local input).

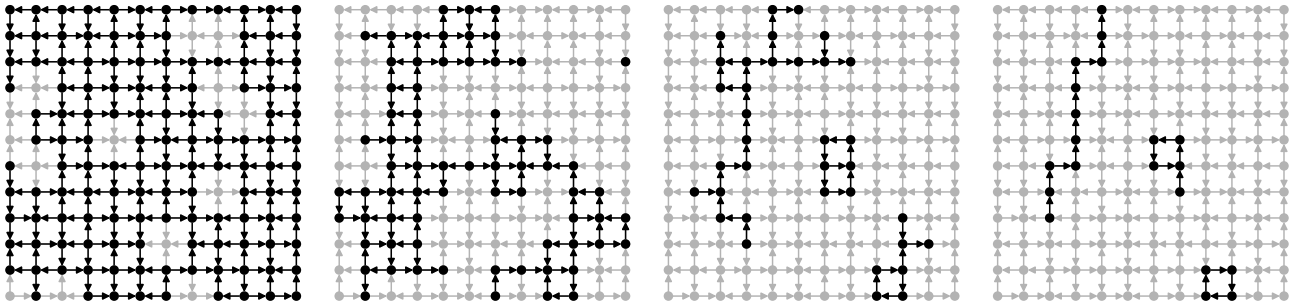
Global input

Fig. 4 compares the performance of directed square lattice reservoirs with local and global input. To our surprise, introducing a global input scheme actually *improves* performance, even surpassing the ESN at $N = 400$. Whereas global input was detrimental for undirected lattice reservoirs, the presence of directed edges results in sufficient reservoir diversity even with uniform input weights. As a side note, we find that classical ESNs show no performance difference with local and global input schemes. This result is particularly promising for physical reservoirs, as it demonstrates the viability of global input schemes, which are significantly simpler to realize.

Table 1 illustrates the simplicity of the square reservoirs

Reservoir type	Reservoir size	Unique input weights	Unique reservoir weights	NARMA-10 NRMSE
Lattice (square)	100	1	1	0.340 (0.019)
Lattice (square)	225	1	1	0.224 (0.016)
Lattice (square)	400	1	1	0.151 (0.009)
ESN	100	100	997 (26)	0.327 (0.020)
ESN	225	225	5098 (74)	0.235 (0.022)
ESN	400	400	16070 (113)	0.176 (0.017)

Table 1: Simplicity of lattice reservoirs compared to classical ESNs. With global input, the square lattice reservoirs contain a single unique magnitude for input and reservoir weights. Displayed values are averages across 20 experiment runs (std. dev.).



(a) $N = 130$, NRMSE = 0.21. **(b)** $N = 70$, NRMSE = 0.27. **(c)** $N = 35$, NRMSE = 0.29. **(d)** $N = 20$, NRMSE = 0.40.

Figure 5: Shrinking of a square lattice reservoir: Four snapshots of the reservoir during incremental removal of nodes. The original 12×12 reservoir is shown in the background in gray.

used in Fig. 4. The input to the entire reservoir is decided by a single scalar weight (the input scaling ι). Furthermore, there is a single unique magnitude used for reservoir weights in $\mathbf{W}^{res}$. For comparison, classical ESNs contain a large amount of unique input and reservoir weights, yet achieve a worse NRMSE on the NARMA-10 benchmark.

We note that the node activations of the square lattice reservoirs with global input are strictly positive, since the NARMA input sequence is strictly positive. This may warrant a change of activation function, as half of $tanh$ remains unused, but this has not been investigated further.

Shrinking lattice reservoirs

We obtain further insight by exploring the effect of shrinking lattice reservoirs. Starting from a 12×12 square lattice reservoir ($N = 144$ nodes), we evaluate the performance impact of removing single nodes. Fig. 6 details the development of benchmark error for each iteration. We also conduct the same experiment with classical ESNs, comparing both models to randomly generated ESNs of the same size.

The results show that both square lattice reservoirs and ESN reservoirs can benefit from removal of a few noisy nodes. Furthermore, almost half the nodes may be removed before performance degrades beyond the initial topology.

Clearly, a large number of the reservoir nodes serve little purpose, and are effectively redundant. Removing redundant nodes only result in a small increase in benchmark error, or even a decrease in some cases. Pruning reservoir nodes in this way decrease the need for computational resources, hence reducing computational cost.

Shrinking reservoirs can also be a valuable for theoretical analysis. Especially so for lattice reservoirs, since they are so readily visualized. Fig. 5 depicts a snapshot of the reservoir topology at four different points during the shrinking process. Performance degrades to that of a delay line (NRMSE = 0.40) when there is around $N = 20$ nodes left in the reservoir (Fig. 5d). This is also exactly what is remaining: a delay line of length 11, and two cycles of length 4. Moreover, an NRMSE of 0.29 is achieved with just $N = 35$ nodes (Fig. 5c). We see a "core" providing the required short-term memory for the NARMA-10 task, with nodes along this stem doing additional processing. This core is also present in the original network (Fig. 5a), but is now easier to spot.

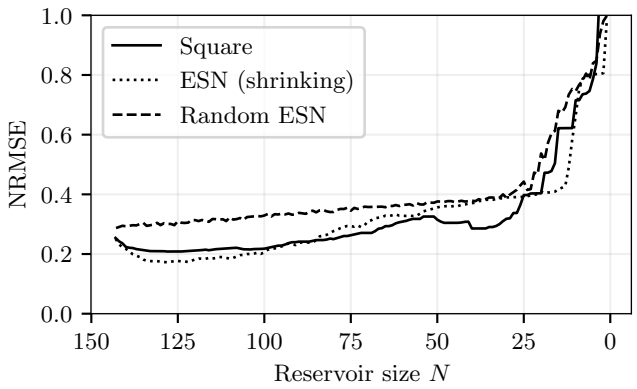


Figure 6: Shrinking reservoir performance. The plots show NARMA-10 NRMSE as a function reservoir size N as nodes are gradually removed. Both square and classical ESN reservoirs are shown, compared to randomly generated ESNs of the same size.

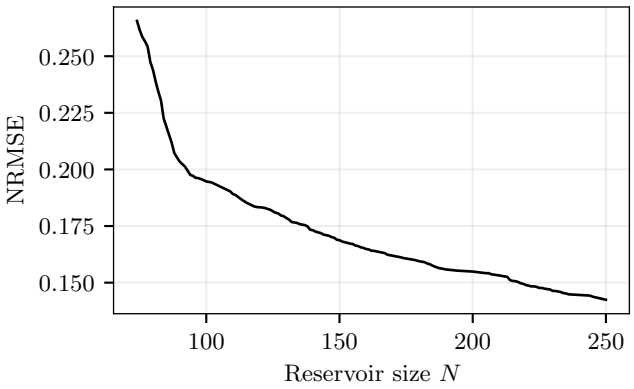


Figure 7: Growth of a square lattice reservoir. The plot shows NARMA-10 NRMSE as a function of reservoir size N during growth. Growth starts from the reservoir depicted in Fig. 8a.

These results illustrate the value of lattice reservoirs as an analytical tool. This rather simple experiment has deduced that the structure doing the heavy lifting is a core stem of memory augmented by surrounding nodes. A comparable analysis is harder to make for the corresponding shrunk ESN reservoir, as the structure of random ESNs is difficult to visualize. Lattice reservoirs, in contrast, readily reveal the spatial structures that emerge to solve computational tasks.

Growing lattice reservoirs

Fig. 7 shows the NRMSE development of a square lattice reservoir during growth. Beginning from a new reservoir with $N = 74$ reservoir nodes (found using the shrinking process), we see a steady decrease in benchmark error for each node addition, and the curve shows no sign of flattening out even at 250 nodes. These results clearly show that building reservoirs specialized for specific tasks will outperform randomly generated ones.

Growing reservoirs also enables further insights into the inner workings of reservoirs. Fig. 8 depicts reservoirs at three different points during growth. Notice how holes appear in the lattice at $N = 250$ (Fig. 8c), which demonstrates that there are parts of the lattice where adding nodes, irrelevant of the directions of its incident edges, will always increase the error.

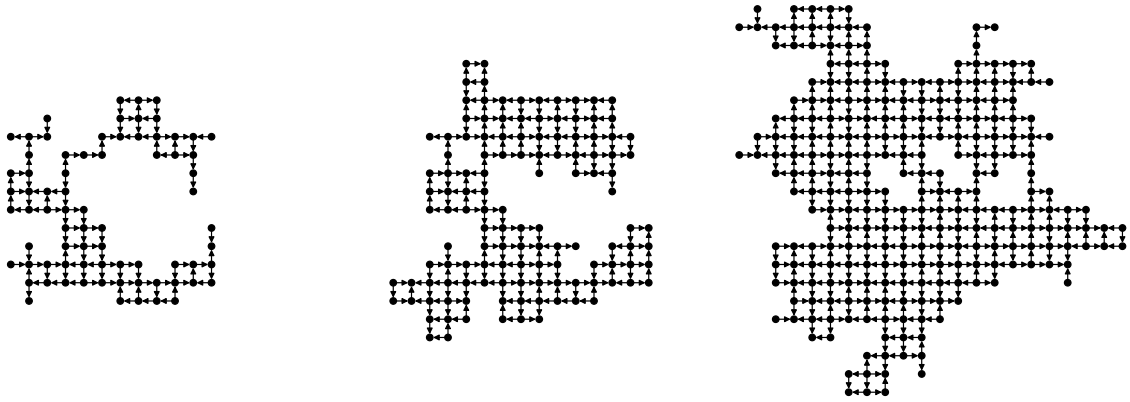
Fig. 8c also illustrates the apparent difficulty of analyzing networks without good heuristics. Consider for example the difference between Fig. 8c, and the much smaller Fig. 5c. The latter provides a much clearer picture of what the reservoir is *doing*.

Restoring undirected edges

What is clear from our results is the importance of directed edges to create a flow of information. But how much information flow is needed? We answer this by analyzing the performance impact of gradually restoring undirected edges. Starting with a fully directional 12×12 square lattice reservoir, we incrementally make each of its 264 edges undirected. At each iteration, the edge causing the least increase in error is made undirected.

The results are shown in Fig. 9. First, there is a small dip in error for the initial 20 or so edges. As an increasing fraction of edges are made undirected, the benchmark error rises gradually. Performance degrades drastically after about 90% of edges have been made undirected.

Again we see that a non-symmetric weight matrix enables richer dynamics, here as a direct consequence of the restored symmetry. However, these results also show that good performance can be obtained with only a modest number of directed edges, as long as the topology is designed appropriately. For physical reservoirs, this indicates that only minor symmetry breaking may be necessary. The requirement for directed edges may not be as strict as previously indicated, as long as the physical topology is chosen carefully. In other words, physical systems with inherent symmetric interactions may still be viable reservoirs, given that an appropriate topology can be found and *some* symmetry breaking is possible.



(a) $N = 74$, NRMSE = 0.27 **(b)** $N = 124$, NRMSE = 0.18 **(c)** $N = 250$, NRMSE = 0.14

Figure 8: Growth of a square lattice reservoir: Three snapshots of the square lattice reservoir during growth. Growth starts from the reservoir in (a), resulting in a gradual increase in performance as shown in Fig. 7.

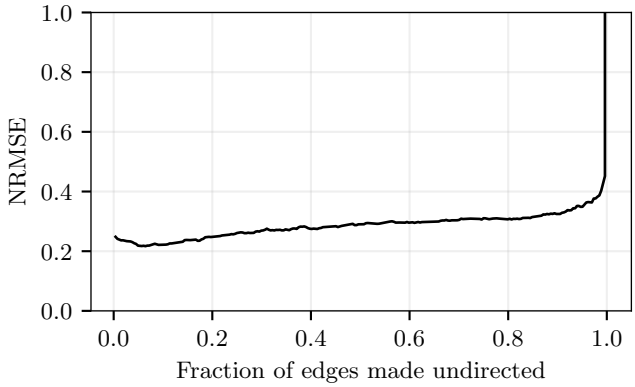


Figure 9: Restoring undirected edges. The plot shows NAMRA-10 NRMSE as the edges of a fully-directed square lattice reservoir are gradually made undirected.

Conclusion

Physical reservoirs are subject to spatial constraints of the real world, which can significantly affect possible reservoir topologies. In this work, we investigate lattice reservoirs, where nodes are placed on a regular lattice which defines both topology and edge weights of the reservoir. Compared to classical ESNs, lattice reservoirs are extremely simple, with all reservoir weights decided by a single scalar value. We simplify the input layer by employing a global input signal broadcast uniformly to all the nodes in the reservoir.

Despite their simplicity, lattice reservoirs perform surprisingly well, in some cases outperforming classical ESNs. A key result is that reservoir edges must be directed to facilitate flow of information. Results suggest that, in order to perform well as a reservoir, a physical system needs to support a directed flow of information. Somewhat surprisingly, a global input signal improves performance in the directed lattice reservoirs. A promising feature for physical reservoirs, as global input schemes are significantly simpler to realize.

The spatial nature of lattice reservoirs makes them easy to visualize, and their finite topology enables several reservoir modification strategies. Shrinking and growing lattice reservoirs can reveal salient features of internal reservoir structure, allowing us to peek into the black box and learn what reservoir is actually doing. For the NARMA-10 benchmark, we find that lattice reservoirs employ a "core" stem of nodes which make up the short-term memory, which is augmented by computational nodes around. Growing reservoirs not only demonstrates the potential for task-optimized reservoirs, but also highlights important computational structures in the lattice. Finally, gradually restoring undirected edges reveals that good performance can be obtained in lattice reservoirs with only minor symmetry breaking.

Our study of lattice reservoirs is only the tip of the iceberg. There seems to be tremendous potential in reservoirs with structured, deterministic topology. Exploration of other lattice topologies, using larger neighborhoods, adding self-connections, and clustering multiple networks, are just a few avenues which may further enhance our understanding of physical reservoirs. Considering other benchmarks and reservoir quality metrics is sure to prove fruitful. More analysis of reservoir structure, for example by classifying input, output and internal nodes, should yield additional insight. Lattice reservoirs bridge the gap between physics and computation, providing invaluable insight for the design of physical reservoirs, and a deeper understanding of important computational structures within.

Acknowledgements

This work was funded in part by the EU FET-Open RIA project SpinENGINE (Grant No. 861618). Thanks to Anders Strømberg for insightful comments and suggestions.

References

Atiya, A. and Parlos, A. (2000). New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11(3):697–709.

Braibant, S., Giacomelli, G., and Spurio, M. (2012). *Particles and Fundamental Interactions*. Undergraduate Lecture Notes in Physics. Springer Netherlands, Dordrecht.

Coulombe, J. C., York, M. C. A., and Sylvestre, J. (2017). Computing with networks of nonlinear mechanical oscillators. *PLOS ONE*, 12(6):e0178663.

Dale, M., Dewhirst, J., O'Keefe, S., Sebald, A., Stepney, S., and Trefzer, M. A. (2019). The Role of Structure and Complexity on Reservoir Computing Quality. In McQuillan, I. and Seki, S., editors, *Unconventional Computation and Natural Computation*, volume 11493, pages 52–64. Springer International Publishing, Cham.

Dale, M., O'Keefe, S., Sebald, A., Stepney, S., and Trefzer, M. A. (2020). Reservoir computing quality: Connectivity and topology. *Natural Computing*.

Deng, Z. and Zhang, Y. (2007). Collective Behavior of a Small-World Recurrent Neural System With Scale-Free Distribution. *IEEE Transactions on Neural Networks*, 18(5):1364–1375.

Dockendorf, K. P., Park, I., He, P., Príncipe, J. C., and DeMarse, T. B. (2009). Liquid state machines and cultured cortical networks: The separation property. *Biosystems*, 95(2):90–97.

Farkaš, I., Bosák, R., and Gergeľ, P. (2016). Computational analysis of memory capacity in echo state networks. *Neural Networks*, 83:109–120.

Gallicchio, C. and Micheli, A. (2019). Reservoir Topology in Deep Echo State Networks. *arXiv:1909.11022 [cs, stat]*, 11731:62–75. arXiv: 1909.11022.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks. *GMD-Report 148, German National Research Institute for Computer Science*.

Jensen, J. H. and Tufte, G. (2020). Reservoir Computing in Artificial Spin Ice. In *ALIFE 2020: The 2020 Conference on Artificial Life*, pages 376–383, Online. MIT Press.

Kawai, Y., Park, J., and Asada, M. (2019). A small-world topology enhances the echo state property and signal propagation in reservoir computing. *Neural Networks*, 112:15–23.

Kittel, C. (2005). *Introduction to Solid State Physics*. Wiley, Hoboken, NJ, 8th ed edition.

Lavis, D. A. (2015). *Equilibrium Statistical Mechanics of Lattice Models*. Theoretical and Mathematical Physics. Springer Netherlands, Dordrecht.

Lukoševičius, M. (2012). A Practical Guide to Applying Echo State Networks. In Montavon, G., Orr, G. B., and Müller, K.-R., editors, *Neural Networks: Tricks of the Trade*, volume 7700, pages 659–686. Springer Berlin Heidelberg, Berlin, Heidelberg.

Maass, W., Natschläger, T., and Markram, H. (2002). Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560.

Manevitz, L. and Hazan, H. (2010). Stability and Topology in Reservoir Computing. In Sidorov, G., Hernández Aguirre, A., and Reyes García, C. A., editors, *Advances in Soft Computing*, volume 6438, pages 245–256. Springer Berlin Heidelberg, Berlin, Heidelberg.

Mead, C. (1990). Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636.

Moon, J., Ma, W., Shin, J. H., Cai, F., Du, C., Lee, S. H., and Lu, W. D. (2019). Temporal data classification and forecasting using a memristor-based reservoir computing system. *Nature Electronics*, 2(10):480–487.

Opala, A., Ghosh, S., Liew, T. C., and Matuszewski, M. (2019). Neuromorphic Computing in Ginzburg-Landau Polariton-Lattice Systems. *Physical Review Applied*, 11(6):064029.

Papp, A., Csaba, G., and Porod, W. (2021). Characterization of nonlinear spin-wave interference by reservoir-computing metrics. *Applied Physics Letters*, 119(11):112403.

Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., and Massar, S. (2012). Optoelectronic Reservoir Computing. *Scientific Reports*, 2:1–6.

Pinna, D., Bourianoff, G., and Everschor-Sitte, K. (2020). Reservoir Computing with Random Skyrmion Textures. *Physical Review Applied*, 14(5):054020.

Rodan, A. and Tino, P. (2011). Minimum Complexity Echo State Network. *IEEE Transactions on Neural Networks*, 22(1):131–144.

Rodan, A. and Tiňo, P. (2012). Simple Deterministically Constructed Cycle Reservoirs with Regular Jumps. *Neural Computation*, 24(7):1822–1852.

Rodriguez, N., Izquierdo, E., and Ahn, Y.-Y. (2019). Optimal modularity and memory capacity of neural reservoirs. *Network Neuroscience*, 3(2):551–566.

Schrauwen, B. (2007). An overview of reservoir computing: theory, applications and implementations. *Proceedings of the 15th European Sympsosium on Artificial Neural Networks*, pages 471–482.

Stepney, S. (2021). Non-instantaneous Information Transfer in Physical Reservoir Computing. In Kostitsyna, I. and Orponen, P., editors, *Unconventional Computation and Natural Computation*, Lecture Notes in Computer Science, pages 164–176, Cham. Springer International Publishing.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. (2018). Recent Advances in Physical Reservoir Computing: A Review. *arXiv:1808.04962 [cs]*. arXiv: 1808.04962.

Tsunegi, S., Taniguchi, T., Nakajima, K., Miwa, S., Yakushiji, K., Fukushima, A., Yuasa, S., and Kubota, H. (2019). Physical reservoir computing based on spin torque oscillator with forced synchronization. *Applied Physics Letters*, 114(16):164101.

Vandoorne, K., Mechet, P., Van Vaerenbergh, T., Fiers, M., Morthier, G., Verstraeten, D., Schrauwen, B., Dambre, J., and Bienstman, P. (2014). Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature communications*, 5:3541.

Vidamour, I. T., Swindells, C., Venkat, G., Manneschi, L., Fry, P. W., Welbourne, A., Rowan-Robinson, R. M., Backes, D., Maccherozzi, F., Dhesi, S. S., Vasilaki, E., Allwood, D. A., and Hayward, T. J. (2023). Reconfigurable reservoir computing in a magnetic metamaterial. *Communications Physics*, 6(1):1–11.