Manuscript Title: with Forced Linebreak

Abrari Noor Hasmi* and Hadi Susanto†

(Dated: March 29, 2022)

An article usually includes an abstract, a concise summary of the work covered at length in the main body of the article.

Usage: Secondary publications and information retrieval purposes.

PACS numbers: May be entered using the environment PACS numbers.

Structure: You may use the `Description` environment to structure your abstract.

I. INTRODUCTION

The success of machine learning especially deep learning have prompted a new interest in scientific community to apply the machine learning method to solve challenges in physical problems modeled by partial differential equations involving spatiotemporal variables. Depending on the approaches used, there are several perks including deep learning/data-driven approaches to physical problems which includes : (1) ability to incorporate measurement data in the model predictions [1] (2) Speed up computations by modelling computation expensive task and using machine learning approach as closure model to describe missing dynamics [2].

Reservoir computing is a specialized version of recurrent neural network, contain an internal states called as reservoir with fixed but random parameters and an output layer which will be trained based on training data. Surprisingly despite the simplied learning compared to RNN, RC has shown superiority in learning tasks compared with RNN [3–5] especially in modelling dynamical systems. An extension of RC for partial differential equation was presented in [1] for Kuramoto-Sivashinsky (KS) equation, an inclusion of knowledge based model was presented in [6, 7]. Sensitivity of the spectral radius of adjacency matrix was investigated in [8], and Bayesian optimization was employed to select the best hyperparameters for each ESN realization [9, 10] .

The generalizability performance of Reservoir Computing includes the ability to capture long term statistics ("climate") in chaotic systems .

This research tries to investingate the forecasting/generative capability of reservoir computing approach applied to coherent structure modelled by the non-linear Schrödinger equations. NLS is an ubiquitous equation in physics that models many phenomena likes optics, ocean waves with modulating amplitude, rogue waves etc. In particular, we would like to train the phenomena of the occurence of rogue waves modelled by the NLS equation. To the best of our knowledge, only [8] which attempted to simulate NLS using Reservoir Computing, in which the authors simulated periodic Akhmadiev breather, Kutnezov-Ma solitons and second order periodic breather collision. The authors concluded that the second order breather collision is more unpredictable compared to the first order breather simulation. As the training data is either time or space periodic in all the simulations, the training data fed to the RC consists of several period of the breather/soliton. In this respect, our research aim is different by the fact that the RC has not yet encounter breather's peak, and we would like RC to naturally learn the dynamics without being explicitly thought about the existance of soliton and modulataion instability.

This report is organized into four sections. This section present a short literature review and motivation about the studies. Section present a brief introduction about parallel reservoir computing. An explanation about training data, typical result of simulation and impact of several hyperparameters are discussed in section. Finally section provides a conclusion and an overview of further research direction.

II. METHOD

A. Reservoir Computing

The main principle in reservoir computing is the separation between internal states (reservoir) and the readout learning. As opposed to the backward propagation method which trains both the states and output parameters, the training in reservoir computing only applies to the readout.

The listening process correspond to constructing reservoir and evolve them based on input and previous states. The reservoir computing adopted was the non-leaky type given by:

$$r_i = \tanh\left(\rho W r_{i-1} + W_{in} u_{i-1}\right) \tag{1}$$

$$\hat{\psi}_i = W_{out}\omega(r_i) \tag{2}$$

* 100060615@ku.ac.ae; Mathematics Department, Khalifa University, Abu Dhabi, United Arab Emirates.; Also at Mathematics and Informatics Department, Institut Teknologi Kalimantan, Balikpapan, Indonesia

† hadi.susanto@ku.ac.ae; Mathematics Department, Khalifa University, Abu Dhabi, United Arab Emirates.

The matrix W is an adjecency matrix which usually set to ensure “Echo States Property”[3]. This condition express that the effect of previous state and previous input at future state should vanish gradually at future time. The adjecency matrix is set to be sparsely connected and random to ensure that the reservoir is rich enough for learning phase. The input coupling matrix W_{in} is a sparse matrix whose entries magnitude not more than σ. Furthermore, the approximate number of non-zero element at each row is fixed.

During the prediction phase, the input vector $u_i^{(k)}$ in equation 1 is replaced by the recent available prediction from the readout (equation ??) using the output matrix obtained during the training phase. However, as the vector $v_i^{(k)}$ consist of overlapping domain, then the output should be constructed by combining information from the neighbouring reservoir $y_i^{(k-1)}, y_i^{(k)} y_i^{(k+1)}$. Calling this prediction as $\tilde{v}_i^{(k)}$, then the steps in prediction phase are:

1. Compute $y_i^{(k)} = W_{out}^k \hat{r}_i^{(k)}$
2. Construct $\tilde{v}_i^{(k)}$ based on $y_i^{(k-1)}, y_i^{(k)} y_i^{(k+1)}$
3. Update the reservoir states: $r_{i+1}^{(k)} = \tanh\left(W^{(k)} r_i^{(k)} + W_{in}^{(k)} \tilde{v}_i^{(k)}\right)$

The implementation of the reservoir computing is based on modification of the code [11] which provide implementations of the reservoir computing.

In this research, we follows the parallel RC [1], which split the spatial data into overlapping domains and train multiple reservoirs for each data chunk. The reasoning of splitting the data is that the exchange of information between two very far seperated spatial grid is not immediate, thus creating reservoir state modelling a connection between faraway points is not physical and also a waste of computation effort.

The function $\omega : \mathbb{R}^M \to \mathbb{R}^M$ is a nonlinear feature transformation of the reservoir [1, 5]. An example of nonlinear transformation which proposed in [1] is:

$$\omega_1(r) = \begin{cases} r_j & j \text{ is odd} \\ r_j^2 & j \text{ is even} \end{cases} \tag{3}$$

For breivity we will denote the even part of transformation as $\hat{\omega}_1(r) = r^2$. Pathak *et al.* [1] noted that the non-linear transformation break the odd symmetry present in the reservoir computing. Study [5] examined different non-linear transformations for simulations of Lorenz-96 system, which are inspired by the non-linearity present in the simulated equation. They conclude that the non-linear transformation enhance the predictive capability of the simulations, however they did not give conclusive result about which non-linearity transformation should be used, as all the transformations give comparable result. Another experiment on the non-linearity effect was done in [12] comparing several other non-linear transformation such as 2.2 and 1.8 power, tanh, exp and sin on the even elements. The numerical experiment on Lorenz-63 showed that square-even transformation had the best performance. Barbosa *et al.* [13] showed that an improvement of the RC prediction can be achieved by respecting the symmetry of the equation present in the original system into the reservoir computing. As the NLS equation has an odd symmetry, the non-linear transformation that should be employed should preserve this symmetry. For that respect, we propose several alternatives non-linear transformation, namely $\hat{\omega}_2(r) = |r|\, r$, the odd version of (3), $\hat{\omega}_3(r) = r^3$ and $\hat{\omega}_4(r) = \mathrm{sgn}(r)\sqrt{|x|}$.

B. Training Process

The loss function is:

$$E^{(k)} = \min_{W_{out}^k} \sum_{i=1}^{N_{train}} \left\| W_{out}^{(k)} \omega(r_i^{(k)}) - s_i^{(k)} \right\|_2^2 + \beta \left\| W_{out}^{(k)} \right\|^2 \tag{4}$$

In essential, the training can be done by using any method to solve linear least square problem. By defining the $\mathbf{R}^{(k)}$ and $\mathbf{S}^{(k)}$ as a matrix whose i-th column is vector $\hat{r}_i$ and s_i respectively, then the training process is equivalent to solve the following matrix equation:

$$\left(\mathbf{R}\mathbf{R}^T + \beta \mathbf{I}\right) W_{out}^T = \mathbf{R}\mathbf{S}^T$$

with notation $\mathbf{R}$ denotes the conjugate transpose of matrix $\mathbf{R}$, and $\mathbf{I}$ is the identity matrix with suitable dimension.

Several literature have discussed the importance of hyperparameters tuning to the prediction capability of the reservoir computing [8, 9, 14]. In particular, [9] observed that varying spectral radius ρ and entries magnitude σ, does not neccessarily require reinitialization of W and W_{in}, thus this enable optimization of these parameters at every realization of reservoir as done in [9, 10, 13]. This study varied only the spectral radius due to computational concern using linear grid search between $[0.4, 1.5]$.

As we would like to investigate the generalizability of the reservoir computing with different dynamics compared to training data, a robust cross validation is an essential step. The cross validation was run by using the recycled validation [9] which deemed as robust and efficient as the training only once need to done for every hyperparameter. The recycled parameter trained the reservoir using the whole training set, and then run feedback/prediction mode on K slices of training data based on exact reservoir states at the beginning of each slice of cross validation set.

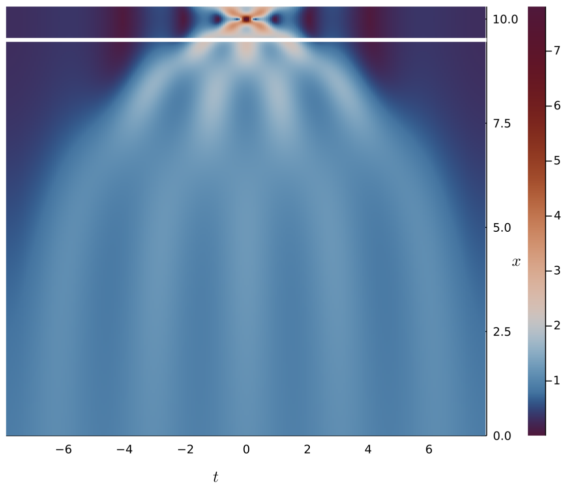

Figure 1. Fifth order breather simulation by NLS

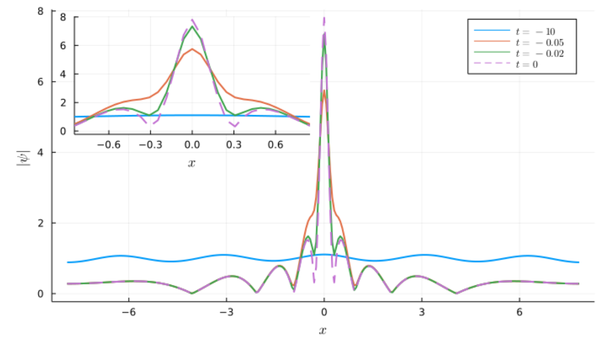

Figure 2. A snapshot of training data at $t = -10$ (beginning of simulation), $t = -0.05, t = -0.02$ and $t = 0$ (expected peak). The inset gives a more detailed view at centre of the coordinate.

Parameter	Explanation	Value
β	Regularization Parameter	10^{-4}
M	Number of reservoirs	16
l	Number of overlap grid	6

Table I. Hyperparameter for RC simulations

III. RESULT

A. Simulation Setup

We trained a set of reservoirs to learn a simulation which governed by the NLS equation:

$$i\frac{\partial \psi}{\partial \tau} + \frac{1}{2}\frac{\partial^2 \psi}{\partial \xi^2} + |\psi|^2 \psi = 0$$

with the amplitude envelope $\psi(\xi, \tau)$ is a function of propagation distance ξ and co-moving time τ. A well known analytic solution of NLs is given by Akhmadiev breather given by

$$\psi(\xi, t) = \left[1 + \frac{2(1-2a)\cosh(\lambda t) + i\lambda \sinh(\lambda t)}{\sqrt{2a}\cos(\Omega x) - \cosh(\lambda t)}\right] e^{it}$$

the behaviour of the solution is governed by a single parameter $a \in (0, 0.5)$, another parameters are related with a, namely the fundamental wave number $\Omega = 2\sqrt{1-2a}$ and growth factor $\lambda = \sqrt{8a(1-2a)}$. Based on this fundamental solution, higher order maximal intensity breather can be constructed by choosing particular $a_2, a_3 \ldots, a_k$ such that under particular phase and amplitude ,the breathers collides at particular time and position resulting at a higher peak with the procedure given by [15]. We simulated the construction of 5^{th} order breather on a uniform background describing a highly non-linear situation. The objective of this simulation is to investigate under what conditions the Reservoir Computing can learn and simulate the maximum peak amplitude of a breather in NLS. For that purpose, we intentionally hide the peak of thr breather from the training data and preserve it as a part of test set.

The training and test data is generated by using 6th order split-step method [16] implemented in Julia library the NonlinearSchrodinger.jl [17]. A brief explanation of the split-step method is given in Appendix. The initial condition was prepared according to [15] which generate the initial condition for maximum intensity soliton via Darboux Transformation. The modulation parameter used in generating the training set is $a_1 = 0.4802$ in which there exists a unique maximum intensity soliton up to fifth order. The length of the spatial domain is one spatial period $[-L/2, L/2]$ with $L = 2\pi/\Omega$ and 256 spatial nodes. The simulation was started from $t = -2$, with $\Delta t = 5 \cdot 10^{-4}$ and the peaks was achieved after , meaning there are 4000 time grids between the start of training data and the peak. Figure 2 illustrate the complexity of the result that we want to achieve. The simulations begin with relatively uniform amplitude , toward $t = 0$, a higher order breather wave representing the rogue waves occurs.

The hyperparameters used in the prediction of NLS is given in Table I which are decided in reference to the simulation of Kuramoto-Sivashinsky equations using RC [18].

To quantify the error between predictions, a metric should be defined. For that purpose we define normalized root mean square error (NRMSE) as:

$$e(t) = \frac{\left\|\hat{\psi}(t) - \psi(t)\right\|}{\|\psi(t)\|} \tag{5}$$

Together with this norm, we also consider the prediction horizon defined as: $PH(\varepsilon) = \min_t e(t) \geq \varepsilon$, The prediction horizon is the first time the NRMSE exceed tolerance ε, in this study we consider case $\varepsilon = 0.4$.

The result of

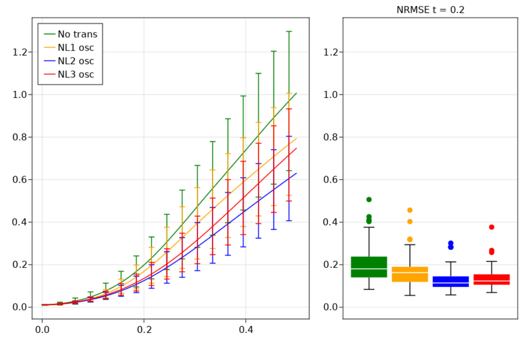

Figure 3. Left: NRMSE of RC simulation on Rogue Wave simulation. The errorbar denotes standard deviation of the result. Right: Boxplot of the 100 simulations of NRMSE during the rogue waves peak.

B. Nonlinear Transformation

We perform 100 realizations of RC simulation for each non-linear transformation described in II A. Each realizations were optimized for the spectral radius by using Recycled Validation method. Our result agree with the result in Barbosa *et al.* [13] which emphasize the symmetry between RC and the modelled system. The odd quadratic transformation ω_2 performed the best compared to the rest of transformation. The cubic transformation ω_3 as expected performed better than ω_1 which attributed to the odd symmetry of ω_3. Despite the odd symmetry, the no transformation case perform poorly compared to ω_1. However, we observe that by using more nodes, the case without non-linear transformation slightly better compared with ω_1.

An observation of the realized

IV. DISCUSSION

ACKNOWLEDGMENTS

The authors PhD is funded by Khalifa University under project number ... and we wish to acknowledge the contribution of Khalifa University's high-performance computing and research computing facilities to the results of this research.

[1] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Physical Review Letters **120**, 24102 (2018).
[2] R. Vinuesa and S. L. Brunton, , 1 (2021), arXiv:2110.02085.
[3] H. Jaeger and H. Haas, Science **304**, 78 (2004).
[4] M. Lukoševičius and H. Jaeger, Computer Science Review **3**, 127 (2009).
[5] A. Chattopadhyay, P. Hassanzadeh, and D. Subramanian, Nonlinear Processes in Geophysics **27**, 373 (2020).
[6] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott, Chaos **28** (2018), 10.1063/1.5028373, arXiv:1803.04779.
[7] A. Wikner, J. Pathak, B. Hunt, M. Girvan, T. Arcomano, I. Szunyogh, A. Pomerance, and E. Ott, Chaos **30** (2020), 10.1063/5.0005541, arXiv:2002.05514.
[8] J. Jiang and Y. C. Lai, Physical Review Research **1**, 33056 (2019), arXiv:1910.04426.
[9] A. Racca and L. Magri, Neural Networks **142**, 252 (2021), arXiv:2103.03174.
[10] F. Huhn and L. Magri, **014402** (2021), 10.1103/physrevfluids.7.014402, arXiv:2106.09780.
[11] F. Martinuzzi, "ReservoirComputing.jl," (2020).
[12] R. Pyle, N. Jovanovic, D. Subramanian, K. V. Palem, and A. B. Patel, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **379**, 20200246 (2021).
[13] W. A. Barbosa, A. Griffith, G. E. Rowlands, L. C. Govia, G. J. Ribeill, M. H. Nguyen, T. A. Ohki, and D. J. Gauthier, Physical Review E **104**, 1 (2021), arXiv:2102.00310.
[14] M. Lukoševičius, in *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012) pp. 659–686.
[15] S. A. Chin, O. A. Ashour, S. N. Nikolić, and M. R. Belić, Physics Letters, Section A: General, Atomic and Solid State Physics **380**, 3625 (2016), arXiv:1607.04504.
[16] H. Yoshida, Physics Letters A **150**, 262 (1990).
[17] O. A. Ashour, (2021), arXiv:2103.14469.
[18] J. Pathak and E. Ott, in *Reservoir Computing* (Springer Nature Singapore, 2021) pp. 117–138.