applied sciences

MDPI

Article

Natural Language Processing Using Reservoir Computers: Network-Theoretic Analysis and Information Flow

Hyeongmo Kim [1,2,†], Wooseung Yoon [1,2,†], Junhyuk Woo [1], Chaehee Kim [1], Sangeun Lee [1], Seungyeon Ji [1,3], Joonho Paik [1,2], Jibaek Lim [1,4], Soyeon Caren Han [5], Hyunsuk Chung [5,*], Sangwook Yi [4,6,*] and Kyungreem Han [1,7,*]

1 Brain Science Institute, Korea Institute of Science and Technology, Seoul 02792, Republic of Korea
2 Department of Physics and Astronomy, Seoul National University, Seoul 08826, Republic of Korea
3 Department of Computer Science and Engineering, Korea University, Seoul 02841, Republic of Korea
4 Department of Philosophy, Hanyang University, Seoul 04763, Republic of Korea
5 School of Computing and Information Systems, University of Melbourne, Melbourne 3053, Australia; caren.han@unimelb.edu.au
6 Department of Artificial Intelligence, Hanyang University, Seoul 04763, Republic of Korea
7 University of Science and Technology-KIST School, Seoul 02792, Republic of Korea
* Correspondence: hyunsuk.chung.1@unimelb.edu.au (H.C.); dappled@hanyang.ac.kr (S.Y.); khan@kist.re.kr (K.H.)
† These authors contributed equally to this work.

Abstract

Reservoir computing (RC) is a highly efficient time-series processing neural network model that can be applied to a wide range of real-world applications, including natural language processing. This study offers physical perspectives on how RC processes natural language, focusing on network structure and information flow. The character memorization task on Shakespeare's works (i.e., a 9,044,939-character database) revealed that memory retention largely depends on the network's structural properties. However, structural property measures such as clustering coefficient, shortest path length, and degree fail to capture the prediction mechanism. Predicting the following character mainly relies on optimal effective memory depth, revealed by Layer-wise relevance propagation analysis. This information-centered view, rather than apparent network topologies, may strongly support the notion of a physical model-based understanding of Large Language Models.

Keywords: reservoir computer; natural language processing; network structure; information flow

Academic Editor: Yutaka Ishibashi

Received: 20 September 2026
Revised: 19 October 2026
Accepted: 28 October 2026
Published: 30 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Reservoir computing (RC) is a highly efficient model that utilizes the complex dynamics of a recurrent neural network (RNN) [1,2], where randomly connected nonlinear neurons in the reservoir exhibit high-dimensional, nonlinear dynamics with intrinsic memory effects. The reservoir maps temporal/sequential inputs into a high-dimensional latent vector space of the information-processing nodes, and a readout model (usually a linear classifier) interprets the information. The readout is trained in a supervised manner while the reservoir remains fixed. This learning strategy enables RCs to solve a wide range of real-world tasks at lower training cost than conventional RNN-based models where all

connection weights are trained. In addition, the RC framework helps circumvent convergence issues during training and vanishing/exploding gradient problems in RNNs [1–7].

RC models have been applied to a wide range of real-world problems for temporal/sequential data such as stock market prediction [8,9], stochastic/chaotic time series prediction [7,10–21], biomedical applications [22–26], speech recognition [27,28], and handwriting recognition [29–33]. Furthermore, RC models have just begun to be applied to natural language processing (NLP) [34–40], yet the underlying mechanisms remain elusive.

This study investigates how RC can efficiently process natural language from a physics perspective, focusing on network structure and information flow therein. Two quantitative tests (i.e., memory retention and character prediction) using Shakespeare's works (i.e., a 9,044,939-character database) are devised. Relevant network structural properties for character memory and prediction abilities are investigated. Furthermore, by tracing the information flow during inference, the most critical mechanisms for NLP at the latent vector space level are elucidated. This also highlights an information-centered approach that uses dynamical modeling to interpret Large Language Models (LLMs).

2. Methods

2.1. Echo State Network

We consider an echo state network (ESN) with K input receiving nodes, N information processing nodes, and O nodes for readout. The input nodes are connected internally following internal weight matrix $W^{in} \in \mathbb{R}^{N\times K}$, the reservoir nodes are connected through $W \in \mathbb{R}^{N\times N}$, and the output nodes upon $W^{out} \in \mathbb{R}^{O\times N}$. The reservoir state $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_N(t)) \in \mathbb{R}^N$ driven by the input state $\mathbf{u}(t) = (u_1(t), u_2(t), \dots, u_K(t)) \in \mathbb{R}^K$ evolves following the equation:

$$x_i(t) = \tanh\left(\sum_{j=1}^{N} W_{ij}x_j(t-1) + \sum_{k=1}^{K} W_{ik}^{in}u_k(t)\right). \quad (1)$$

The output $\hat{\mathbf{y}}(t) = \left(\hat{y}_1(t), \hat{y}_2(t), \dots, \hat{y}_O(t)\right) \in \mathbb{R}^O$ is obtained from the reservoir state by the readout as

$$\hat{\mathbf{y}}(t) = g(W^{out}\mathbf{x}(t) + \mathbf{b}), \quad (2)$$

where g is the output activation function and $\mathbf{b} \in \mathbb{R}^O$ is a bias vector. Only W^{out} and $\mathbf{b}$ are trained to minimize the error between the target vector $\mathbf{y}(t)$ and the output vector $\hat{\mathbf{y}}(t)$, whereas W^{in} and W are randomly generated and remain fixed. All ESNs use the same W^{in}, with weights sampled from a Gaussian distribution with zero mean and variance $0.9^{-2}K^{-1}$.

The W are varied to represent four network topologies: random, regular, small-world, and scale-free (see Appendix A). To examine the effect of network density, the number of links M (i.e., non-zero entries in W) is varied from 50,000 to 500,000 by adding 50,000 links at each step.

2.2. Natural Language Processing Using Echo State Network

We apply the ESN to natural language processing, specifically to character-level language modeling (Figure 1a). In addition to the components in Section 2.1, the model includes an embedding matrix of the input characters, $E \in \mathbb{R}^{K\times V}$, where V is the vocabulary size. Each character is represented as a one-hot vector $\mathbf{s} \in \mathbb{R}^V$ and mapped to an embedding vector $\mathbf{u} = E\mathbf{s}$. The elements of E are independently sampled from a standard Gaussian distribution and kept fixed across all experiments.

https://doi.org/10.3390/app162110791

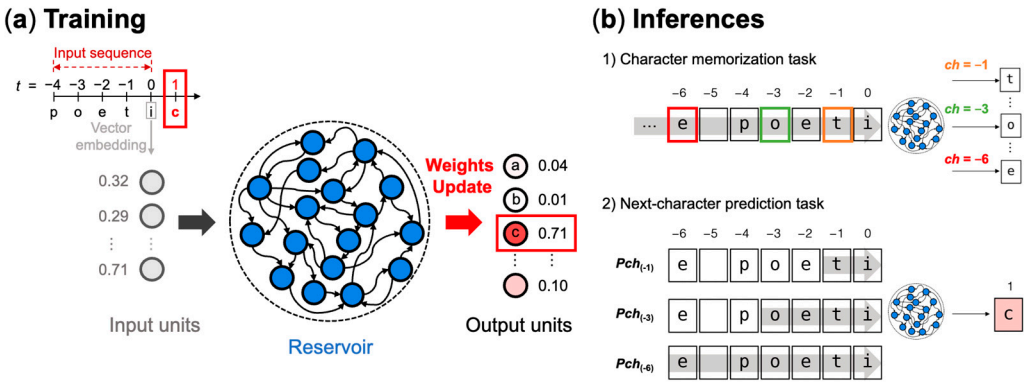


Figure 1. Schematic diagram illustrating the natural language processing using an echo state network. (**a**) Training an ESN for natural language processing. Each character in the input sequence is mapped to 16 input units by an input embedding matrix. The reservoir is driven by the sequence of the input embedding vectors from $t = -T + 1$ to 0. The readout weights are updated to map the final state to a probability distribution over the vocabulary. In this system, an ESN model is trained to infer the target character. (**b**) Two inference tasks on ESN. 1) In the character memorization task, the ESN receives a long character sequence and is evaluated on the reconstruction of the character at position ch relative to the last input character. 2) In the next-character prediction task, the ESN receives the character sequence of length h denoted by $Pch_{(-h)}$ and is evaluated on the prediction of the next-character.

For an input sequence of T characters $\mathbf{s}(1), \mathbf{s}(2), \ldots, \mathbf{s}(T)$, the embedding vectors $\mathbf{u}(t) = E\mathbf{s}(t)$ drive the reservoir according to Equation (1). The reservoir state is initialized to zero, $\mathbf{x}(0) = 0$, and only the final reservoir state $\mathbf{x}(T)$ is used for the readout. The final state is mapped to a probability distribution $\hat{\mathbf{y}} \in \mathbb{R}^O$ over the vocabulary (i.e., $O = V$) by

$$\hat{\mathbf{y}} = \mathrm{softmax}(W^{out}\tilde{\mathbf{x}}(T) + \mathbf{b}), \qquad \tilde{x}_i(T) = \frac{x_i(T) - \langle \mathbf{x}(T) \rangle}{\sigma_x(T)}, \tag{3}$$

with a softmax activation function and the normalized reservoir state $\tilde{\mathbf{x}}(T)$, obtained using the mean $\langle \mathbf{x}(T) \rangle = \sum_{i=1}^{N} x_i(T)/N$ and standard deviation $\sigma_x(T) = \sqrt{\sum_{i=1}^{N} [x_i(T) - \langle \mathbf{x}(T) \rangle]^2/N}$. The readout parameters (i.e., W^{out} and $\mathbf{b}$) are trained by minimizing the cross-entropy loss

$$H = -\frac{1}{N_s} \sum_{n=1}^{N_s} \sum_{j=1}^{V} y_{n,j} \log \hat{y}_{n,j}\,, \tag{4}$$

where N_s is the number of samples, $y_{n,j}$ is the j-th component of the one-hot encoding vector for the target y_n and $\hat{y}_{n,j}$ is the j-th component of $\hat{\mathbf{y}}$ for the n-th sample.

In this study, we use Shakespeare's works from The Folger Shakespeare [41]. The text is lowercased during preprocessing, yielding 9,044,939 characters with a vocabulary size of $V = 59$. The first 90% is used for training, and the remaining 10% for testing. From the text, training and test samples are generated by sliding a window of T characters with a stride of one, where T is the input sequence length. For the optimization, Adam is used with a learning rate of 10^{-4} and a batch size of 1024 for 70 epochs. We report the test loss at the final epoch. All results are averaged over three network realizations.

2.3. Character Memorization Task

The memory of the reservoir during NLP is quantitatively measured through the reconstruction test (Figure 1b-1). The reservoir is driven by a sequence of $T = 32$ characters, and a readout is trained to reconstruct the character at position ch relative to the last input character. For example, $ch = -1$ denotes the character immediately preceding the last input. We vary ch from -1 to -6. The embedding matrix, reservoir, and training settings are the same as in Section 2.2, and performance is evaluated by the test loss, the cross-entropy H (Equation (4)).

2.4. Next-Character Prediction Task

The prediction power of the networks with different topologies is evaluated through the next-character prediction task, where the readout predicts the character immediately following the input sequence (Figure 1b-2). To examine the effect of context length on prediction performance, we vary the length of the input sequence. Specifically, the reservoir receives an input sequence consisting of $a + 1$ consecutive characters, denoted by $Pch_{(-a)}$. We vary a from 1 to 6. All other settings, including the reservoir, the readout, and the training procedure, are identical to those of the character memorization task.

2.5. Layer-Wise Relevance Propagation

The information flow during language processing is quantified using layer-wise relevance propagation (LRP) [42,43]. We evaluate the relevance assigned to the input $\mathbf{u}(t)$ at each time step $(t = -T + 1, \ldots, 0)$ with respect to the model prediction $\hat{y}_c$ with $c = \mathrm{argmax}_s\, \hat{y}_s$. This relevance identifies which time steps in the input sequence contribute most strongly to the inference. The relevance of the final state $x_i(t = 0)$ with respect to $\hat{y}_c$ is given by

$$R_i^x(t = 0) = W_{ci}^{out} x_i(t = 0). \tag{5}$$

The relevance is then propagated backward through the reservoir dynamics and distributed between the preceding reservoir state and the input at each time step. For $t \leq 0$, the relevances assigned to the past state $x_i(t - 1)$ and the input $u_k(t)$ are calculated as

$$R_i^x(t - 1) = \sum_{j=1}^{N} \frac{W_{ji} x_i(t - 1)}{Z_j(t) + \epsilon \mathrm{sgn}\left[Z_j(t)\right]} R_j^x(t), \tag{6}$$

$$R_k^u(t) = \sum_{j=1}^{N} \frac{W_{jk}^{in} u_k(t)}{Z_j(t) + \epsilon \mathrm{sgn}\left[Z_j(t)\right]} R_j^x(t), \tag{7}$$

where $Z_j(t) = \sum_{k=1}^{N} W_{jk} x_k(t - 1) + \sum_{k=1}^{K} W_{jk}^{in} u_k(t)$ denotes a normalization factor and ϵ is a stabilizer for numerical stability. For the following analysis, the normalized temporal relevances are defined as

$$\overline{R^u}(\Delta) = \frac{\sum_{k=1}^{K} |R_k^u(-\Delta)|}{\sum_{\Delta'=0}^{T-1} \sum_{k=1}^{K} |R_k^u(-\Delta')|}, \tag{8}$$

$$\overline{R^x}(\Delta) = \frac{\sum_{i=1}^{N} |R_i^x(-\Delta)|}{\sum_{\Delta'=0}^{T-1} \sum_{i=1}^{N} \left|R_i^x(-\Delta')\right|}, \tag{9}$$

to quantify the extent to which the input and reservoir state at each time step contribute to the model prediction, respectively.

https://doi.org/10.3390/app162110791

3. Results

As shown in Figure 2, cross-entropy loss H in the character memorization task largely depends on the network structure. For long-term memorization tasks (i.e., $ch = -4$, -5, and -6), the loss decreases as the number of links M increases. A context-independent unigram predictor [44] based solely on the empirical character frequencies of the corpus yields a cross-entropy loss of 3.056, serving as the baseline for the task. For each task, random networks exhibit the steepest decrease in loss H with increasing M, followed by scale-free networks, while regular and small-world networks exhibit the most gradual decreases. Increasing the number of links improves memory retention in all network topologies, although the degree of improvement varies, with random networks showing the greatest efficiency. The effect of increasing connectivity on memory retention is strongly topology dependent, with random networks showing the largest gain per additional link. In contrast, short-term memorization tasks (i.e., $ch = -1$, -2, and -3) show low cross-entropy loss H regardless of the number of links M and the network topology. This indicates that a small number of recurrent links is sufficient to retain information from recent time steps.

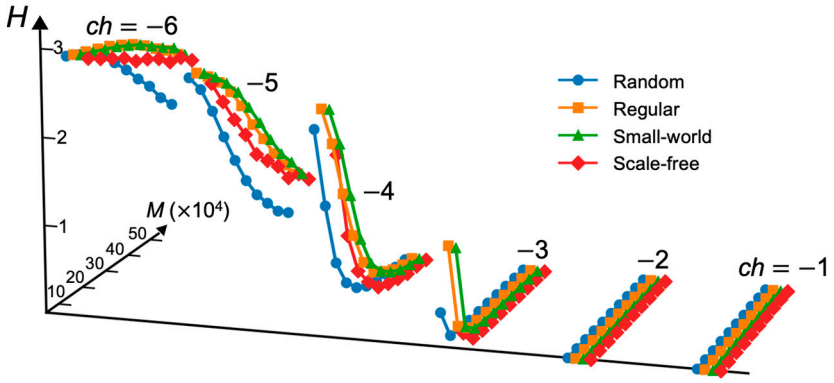


Figure 2. Character memorization tasks using different network topologies: random, regular, small-world, and scale-free. The tasks are specified by $ch = -1, -2, \ldots, -6$, where $ch = -a$ represents inference of the character at $t = -a$. For each topology, the number of links M is varied from 50,000 to 500,000 in increments of 50,000. Cross-entropy loss H between the predicted and target characters is displayed.

Next, we describe the correlation between character memorization performance and a basic structural measure, mean degree $\langle k \rangle$. Mean degree $\langle k \rangle$ is the average number of links connected to a node. As shown in Figure 3a, the cross-entropy loss H in character memorization tasks decreases as mean degree $\langle k \rangle$ increases, regardless of network type. Across all tasks ($ch = -2, -3, \ldots, -6$), the loss curves collapse onto a single curve, indicating that mean degree provides a common structural description of memory retention across regular, random, small-world, and scale-free networks. Even at the same M, networks with larger $\langle k \rangle$ exhibit lower loss, explaining the superior memory retention of random networks (Figure 2). These results indicate that mean degree is the primary structural quantity governing memory retention; other structural properties such as the clustering coefficient and shortest path length may provide a small margin for fine-tuning the memory (see Figures A1 and A2 in Appendix B). However, next-character prediction does not rely on the network's structural properties; not only the clustering coefficient and the shortest path length, but even the mean degree poorly characterizes the underlying prediction mechanism. For comparison, a bigram predictor based only on the single preceding character [44] yields a cross-entropy loss of $H = 2.465$, providing a simple low-order statistical baseline for the prediction task. As shown in Figure 3b, the cross-entropy loss

varies nonlinearly with mean degree—this indicates that the optimal mean degree may reflect nonlinear information processing in the high-dimensional latent vector space due to the network topology.

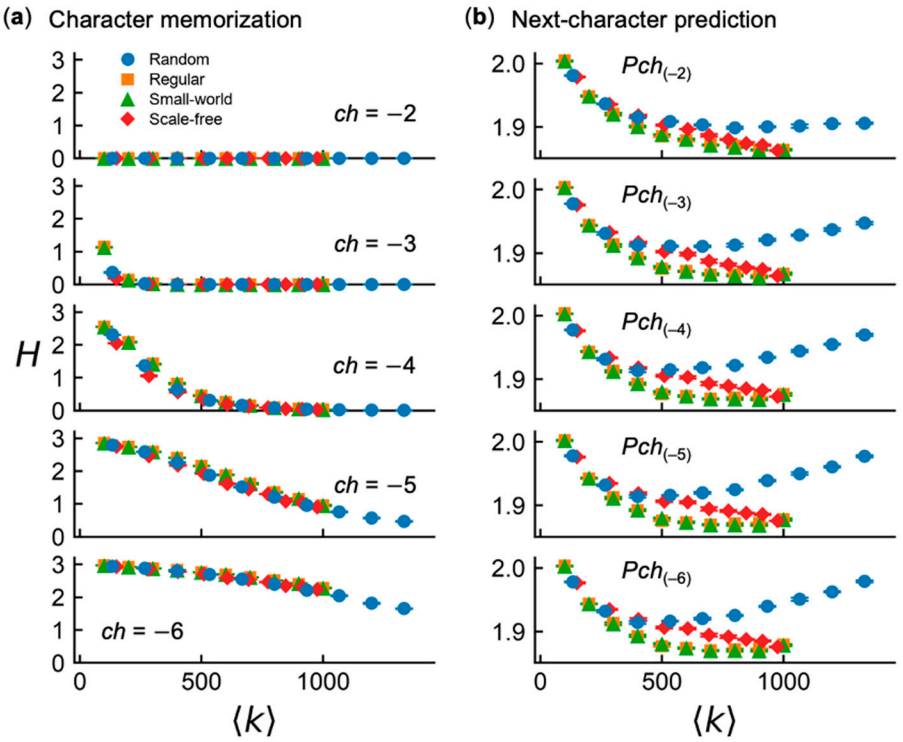


Figure 3. Correlation between natural language processing performance and mean degree of the network. (**a**) Cross-entropy loss H in the character memorization tasks ($ch = -2, -3, \ldots, -6$) is shown as a function of the mean degree $\langle k \rangle$. (**b**) Cross-entropy loss in the next-character prediction tasks ($Pch_{(-2)}, Pch_{(-3)}, \ldots, Pch_{(-6)}$) is shown as a function of $\langle k \rangle$, where $Pch_{(-T+1)}$ denotes the prediction task using an input sequence of length T. Both panels show results for the same network structures as in Figure 2, with error bars denoting the standard error over three network realizations.

Layer-wise relevance propagation (LRP) analysis provides an explanation of the prediction mechanism by quantifying information flow in high-dimensional latent vector space. As shown in Figure 4a, information flow from the reservoir state $\mathbf{x}(-\Delta)$ and input $\mathbf{u}(-\Delta)$ to the model prediction is quantified by normalized relevances $\overline{R^x}(\Delta)$ and $\overline{R^u}(\Delta)$, respectively. The relevance decays exponentially with Δ, and we thus define an effective memory depth ξ of the reservoir by fitting the exponential curve $f(\Delta) = Ae^{-\Delta/\xi}$ to the relevance decay curve $\overline{R^x}(\Delta)$. All fits are performed using the least-squares method, with coefficients of determination $R^2 > 0.98$ across all network topologies and input windows. The memory depths also exhibit low fitting uncertainty, as quantified by their relative standard errors. The relative standard error is defined as $\sqrt{\mathrm{Var}(\xi)}/\xi$, where $\mathrm{Var}(\xi)$ is the corresponding diagonal element of the parameter covariance matrix. The mean relative standard errors ranged from 1.2% to 4.4% across network topologies, and remained below 8% for all fitted profiles (see Table A1 in Appendix C). To compare the effective memory depth per model parameter, we calculate the normalized memory depth:

$$\tilde{\xi} = \xi / M. \tag{10}$$

As shown in Figure 4b, the normalized memory depth $\tilde{\xi}$ decreases with M in regular, small-world, and scale-free networks; however, that of the random network exhibits nonmonotonic changes.

https://doi.org/10.3390/app162110791

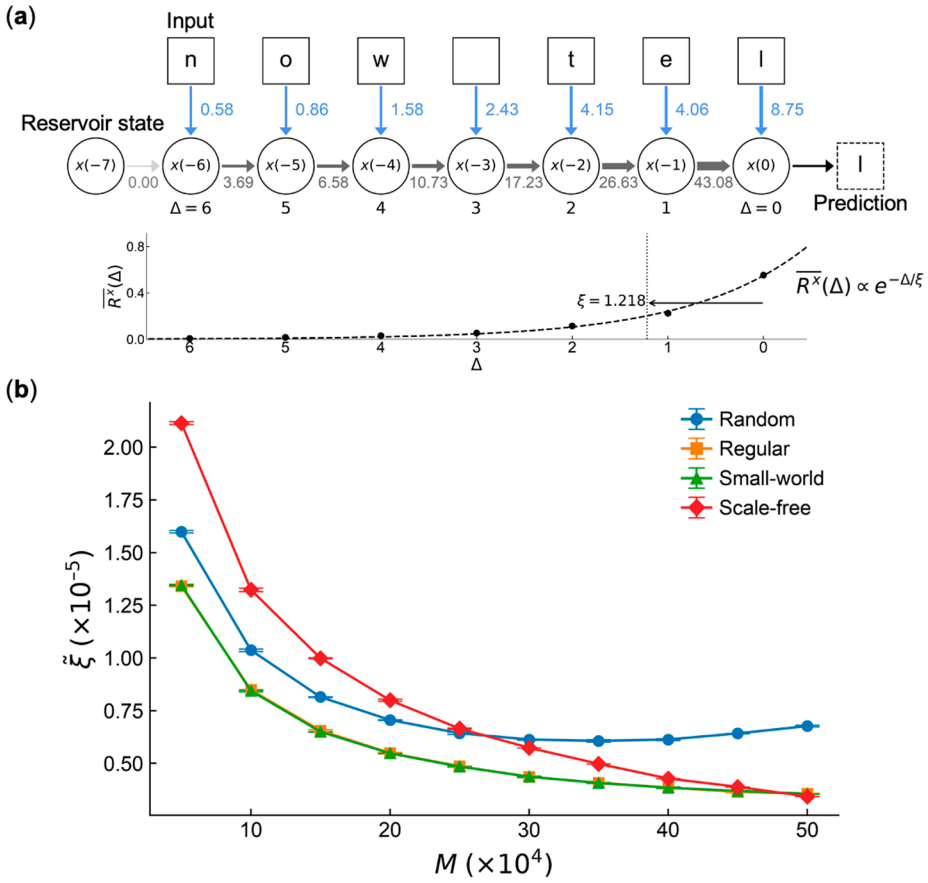


Figure 4. Information flow during next-character prediction. (**a**) Schematic illustration of LRP-based quantification of information flow and extraction of the effective memory depth ξ. The prediction task $Pch_{(-6)}$ is illustrated as an example. The arrow thickness corresponds to the values of $\overline{R^u}$ and $\overline{R^x}$, the magnitude of relevances assigned to the input and the reservoir state at each time step, respectively. The effective memory depth ξ is obtained by fitting the relevance decay curve $\overline{R^x}(\Delta)$ to an exponential function $f(\Delta) \propto e^{-\Delta/\xi}$. (**b**) Effective memory depth normalized by the link number is displayed for different network topologies. Error bars denote the standard error over three network realizations.

The normalized memory depth $\tilde{\xi}$ is large at small M regardless of network topology, leading to poor prediction performance (Figure 3b). As M increases, the value of $\tilde{\xi}$ in regular, small-world, and scale-free networks converges toward an optimal value $\tilde{\xi} \approx 0.3$, whereby the cross-entropy loss is minimized. This convergence is faster in regular and small-world networks than in scale-free networks, consistent with the larger improvement in prediction performance observed in Figure 3b. In contrast, the memory depth in random networks does not reach the optimal value over the entire range of M. It decreases with increasing M and then slightly increases at large M, mirroring the nonmonotonic behavior of the cross-entropy loss in the prediction task.

This LRP-based analysis shows that prediction requires an appropriate, rather than maximal, effective memory depth. By resolving the reservoir's latent vector dynamics at the level of information-flow, the analysis should elucidate the prediction mechanism, overcoming limitations of network structural concepts.

4. Discussion and Conclusions

This study described the mechanisms underlying natural language processing (NLP) in the recurrent neural network model called an ESN (the representative reservoir computer model) from a physical perspective. The primary approaches applied were (1)

network-theoretic analysis to grasp the influence of network structure and (2) layer-wise relevance propagation (LRP) for tracing information flow during NLP embedded in the latent vector space of the reservoir network topologies.

It is revealed that character memorization is correlated with the network's structural properties, such as the degree of a node, clustering coefficient, and shortest path length; this certainly can provide design principles for how network topology can be fine-tuned to have larger memory capacities at the same network size (i.e., the same number of links and nodes) (Figure 3a). However, the character prediction mechanism may not be fully explainable at the topology level (Figure 3b), suggesting that facets of information processing in a high-dimensional latent vector space should be explored.

As such, layer-wise relevance propagation analysis provided an alternative explanation for the prediction mechanism. Effective memory depth was defined based on temporal information flow, and accurate next-character prediction was found to require an optimal effective memory depth (Figure 4). The analysis revealed that network structures retaining excessive past information increase the effective memory depth, which in turn degrades prediction performance. This information-centered view, rather than the apparent network topology concept, may strongly support the notion of a physical model-based understanding of Large Language Models (LLMs).

Such dynamical descriptions of language processing may provide a way to address theoretical questions in LLMs. Recent work has applied reservoir-based architectures to token-level language modeling by combining ESNs with additional trainable components, suggesting that reservoir dynamics can remain useful within larger language-model architectures [40]. Next-character prediction performance improves with increasing training data and gradually saturates (see Figure A3 in Appendix D). Whether more complex scaling behavior emerges in larger reservoir-based language models remains an open question.

The concept of 'AI as a complex dynamical system' was first proposed by Weinan E. [45], which was concretized through the concept of neural ordinary differential equations (ODEs) [46], which map feature transformations in neural networks to the numerical methods used to solve ODEs. This enabled interpreting neural networks as dynamical systems governed by equations of motion (corresponding to neural ODEs) and has been successfully applied to the design of ResNets with improved stability and robustness [47–50]. In line with this, the transformer architecture can be formalized as a multi-particle dynamical system [51–56]: once inputs are transformed into vector representations in an embedding space through data encoding and feature projection, these representations—system states—evolve according to the system dynamics defined by attention layers, pointwise feedforward operations, and layer normalization.

Previous studies have formalized transformer dynamics primarily focusing on self-attention mechanisms for unimodal processing, without incorporating feature fusion and cross-modal processing. However, our recent work successfully describes both self-attention and cross-attention mechanisms by describing latent vector dynamics under heterogeneous interactions, and explains how cross-modal bias can emerge during multimodal information processing [57].

The concept of memory depth based on our LRP analysis may extend this dynamical model view by incorporating an information-centric view, providing a useful framework for quantifying information flow in latent vector dynamics. The dynamical system modeling, combined with an information-theoretic viewpoint, may serve as a universal model to understand information processing of deep learning models and LLMs for complex tasks, including NLP, ultimately to explain and interpret the black-box-like self-attention, feedforward, and cross-attention stacks and the resulting undesired and incomprehensible biases in embedding, latent, and representation spaces.

https://doi.org/10.3390/app162110791

Author Contributions: Conceptualization, K.H., S.C.H., H.C., S.Y.; methodology, K.H., W.Y., H.K.; software, W.Y., H.K., J.W., J.L., J.P., K.H.; validation, W.Y., H.K., K.H.; formal analysis, K.H., H.K., W.Y.; investigation, K.H., H.K., W.Y., J.L., J.W.; resources, W.Y., H.K., J.L., J.W., J.P.; data curation, W.Y., H.K., J.W., J.L., J.P.; writing—original draft preparation, K.H., H.K., W.Y., J.L., C.K., S.L., S.J.; writing—review and editing, K.H., S.C.H., S.Y., H.C.; visualization, H.K., W.Y., J.L., J.W., C.K., S.L., S.J.; supervision, K.H., S.Y., S.C.H., H.C.; project administration, K.H.; funding acquisition, K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2025-02217259, Development of self-evolving AI bias detection-correction-explain platform based on international multidisciplinary governance) and (No.RS-2026-25519206, Development of human-like intelligent generative agent based on interactive multimodal reverse prompting) and the Korea Institute of Science and Technology (KIST) Institutional Program (Project No. 2E33721, 26E0131 and 26V0032).

Data Availability Statement: The data presented in this study were derived from the following publicly available resource: https://www.folger.edu/explore/shakespeares-works/all-works accessed on 10 September 2026. The code is available from the corresponding authors upon request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RC	Reservoir Computing
ESNs	Echo State Networks
NLP	Natural Language Processing
RNN	Recurrent Neural Network
LRP	Layer-Wise Relevance Propagation
ODEs	Ordinary Differential Equations

Appendix A. Generation of Network Topologies

$W \in \mathbb{R}^{N\times N}$ in Section 2.1 defines a recurrent network of N reservoir units, in which a link from unit j to unit i exists if $W_{ij} \neq 0$. The number of links M thus equals the number of non-zero entries in W, and self-connections are excluded ($W_{ii} = 0$). We consider four network topologies for W: random, regular, small-world, and scale-free (Appendices A.1–A.4). The random topology consists of directed links, whereas the other three topologies consist of pairs of reciprocal links. For all topologies, the non-zero entries are independently sampled from a Gaussian distribution with zero mean and variance $0.9^{-2}N^{-1}$. M is varied from 50,000 to 500,000 in increments of 50,000.

Appendix A.1. Random

A total of M distinct links are selected uniformly at random from $N(N-1)$ possible links. With $N = 750$, M is varied from 50,000 to 500,000 by adding 50,000 links at each step.

Appendix A.2. Regular

Each of N units on a ring is connected to its z nearest neighbors ($z/2$ on each side) by reciprocal links following the Watts–Strogatz model [58], yielding $M = Nz$. With $N = 1000$, z is varied from 50 to 500 in steps of 50, so that M ranges from 50,000 to 500,000.

Appendix A.3. Small-World

Starting from the ring lattice in Appendix A.2, each pair of reciprocal links is rewired with probability $p = 0.1$ following the Watts–Strogatz model [58], with both links of a pair moved together, so that the network remains reciprocal. Since rewiring preserves the number of links, $M = Nz$ as in Appendix A.2. Unlike in Appendix A.1, links are not added cumulatively; instead, a new network is generated for each z.

Appendix A.4. Scale-Free

Following the Barabási–Albert model [59], units are added one at a time, and each new unit is connected by pairs of reciprocal links to m existing units, which are chosen with probability proportional to their degree. This yields $M = 2m(N - m)$ links. To obtain exact values of M, N is set to $m + 625$, and m is varied from 40 to 400 in steps of 40, giving M from 50,000 to 500,000 in increments of 50,000.

Appendix B. Character Memorization and Prediction Task with Network-Theoretic Measures

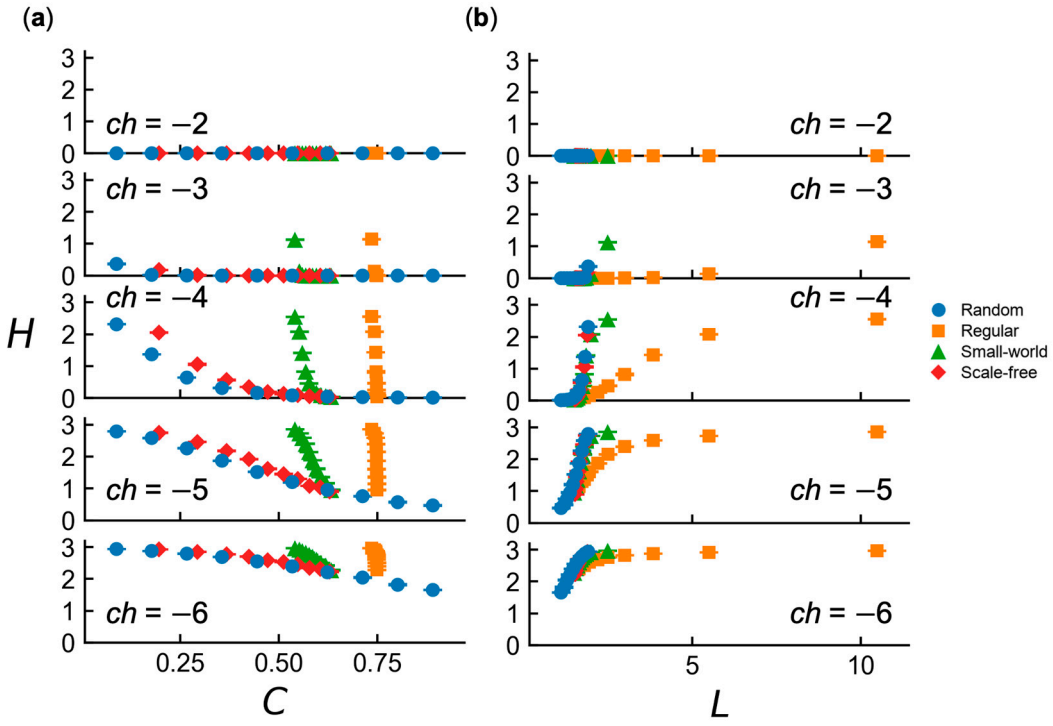


Figure A1. Character memorization task: network structural measures C and L. (**a**) Cross-entropy loss H in the character memorization tasks ($ch = -2, -3, \ldots, -6$) as a function of the average clustering coefficient C and (**b**) the average path length L. Error bars denote the standard error over three network realizations.

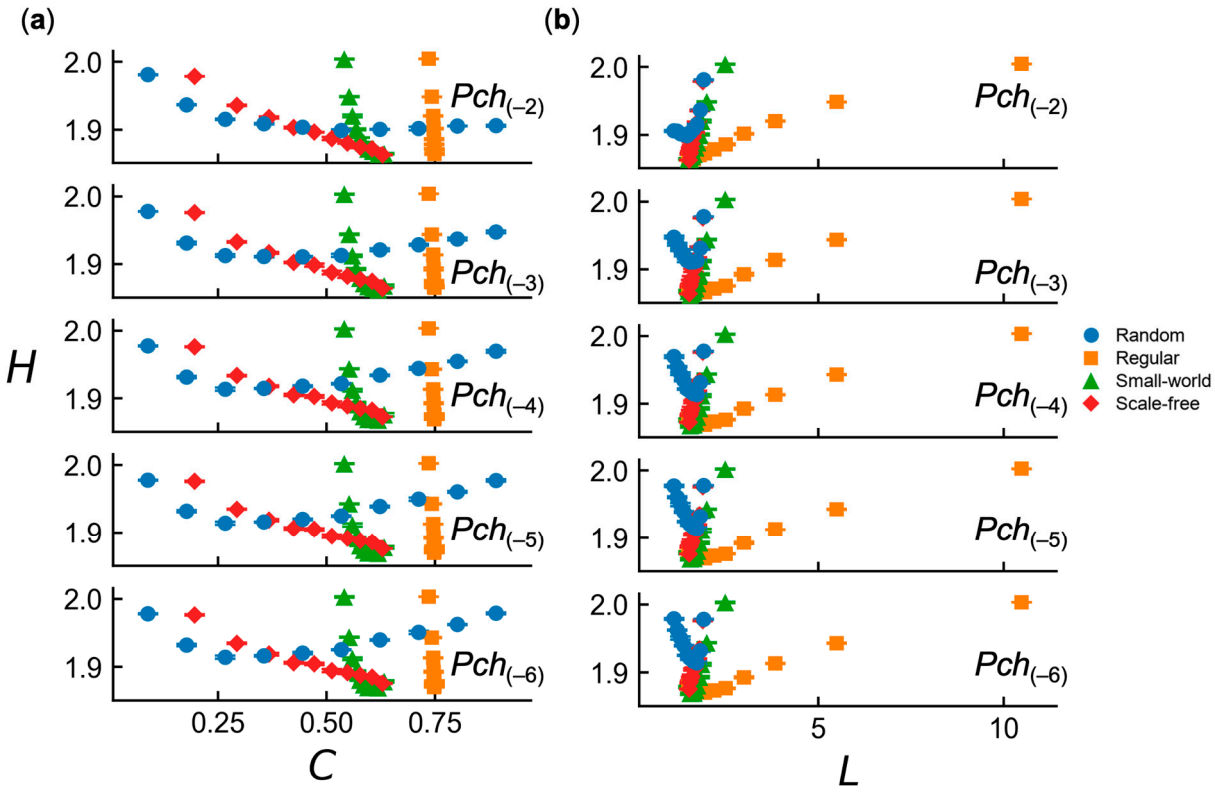


Figure A2. Next-character prediction task: network structural measures C and L. (**a**) Cross-entropy loss H in the next-character prediction tasks ($Pch_{(-2)}, Pch_{(-3)}, \ldots, Pch_{(-6)}$) as a function of the average clustering coefficient C and (**b**) the average path length L. Error bars denote the standard error over three network realizations.

Appendix C. Uncertainty Analysis of Effective Memory Depth Fitting

Table A1. Relative standard error statistics for the fitted memory depth ξ for each network topology. The mean, median, and maximum are evaluated over all input windows for each topology. The presented values are averages over three realizations for each topology, with standard errors across realizations.

Network Topology	Mean ($\sqrt{\mathrm{Var}(\xi)}/\xi$) (%)	Median ($\sqrt{\mathrm{Var}(\xi)}/\xi$) (%)	Max ($\sqrt{\mathrm{Var}(\xi)}/\xi$) (%)
Random	3.537 ± 0.048	3.434 ± 0.078	7.596 ± 0.085
Regular	1.210 ± 0.042	0.892 ± 0.022	2.920 ± 0.119
Small-world	1.210 ± 0.034	0.818 ± 0.059	3.029 ± 0.174
Scale-free	4.384 ± 0.529	4.607 ± 0.531	6.243 ± 0.678

https://doi.org/10.3390/app162110791

Appendix D. Next-Character Prediction Performance Across Different Training Dataset Sizes

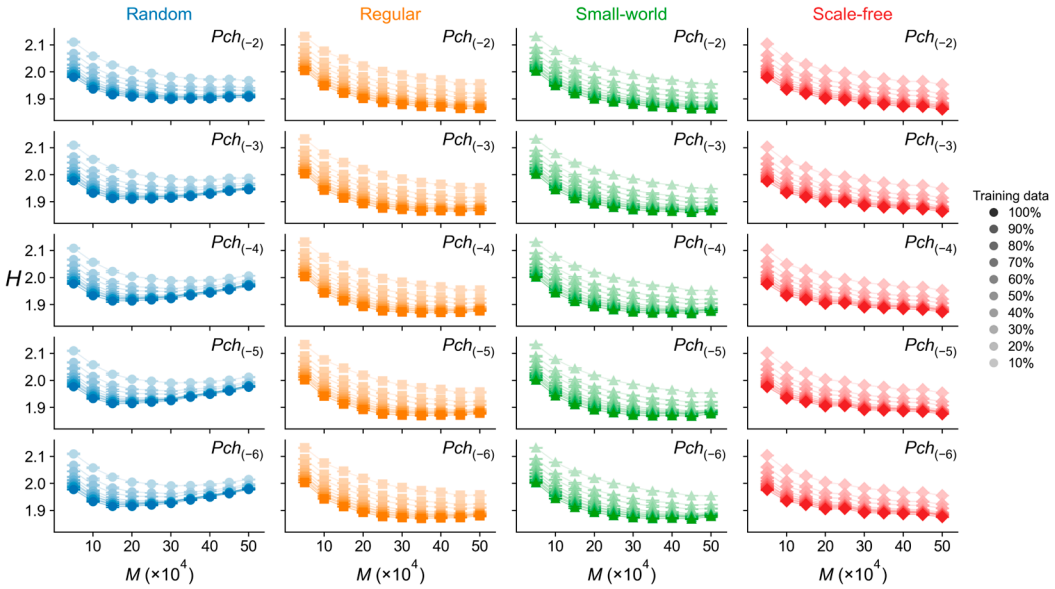


Figure A3. Next-character prediction performance for different training dataset sizes. The training datasets consist of the first 10%, 20%, ..., and 100% of the original corpus.

References

1. Schrauwen, B.; Verstraeten, D.; Van Campenhout, J. An overview of reservoir computing: Theory, applications and implementations. In Proceedings of the ESANN 2007 Proceedings—15th European Symposium on Artificial Neural Networks, Bruges, Belgium, 25–27 April 2007; pp. 471–482.
2. Lukoševičius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127–149. https://doi.org/10.1016/j.cosrev.2009.03.005.
3. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
4. Maass, W.; Natschläger, T.; Markram, H. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Comput.* **2002**, *14*, 2531–2560. https://doi.org/10.1162/089976602760407955.
5. Jaeger, H. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. GMD Technical Report 148; German National Research Center for Information Technology: Bonn, Germany, 2001.
6. Steil, J.J. Backpropagation-decorrelation: Online recurrent learning with O(N) complexity. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541), Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 843–848.
7. Verstraeten, D.; Schrauwen, B.; D'Haene, M.; Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Netw.* **2007**, *20*, 391–403. https://doi.org/10.1016/j.neunet.2007.04.003.
8. Lin, X.; Yang, Z.; Song, Y. Short-term stock price prediction based on echo state networks. *Expert Syst. Appl.* **2009**, *36*, 7313–7317. https://doi.org/10.1016/j.eswa.2008.09.049.
9. Ilies, I.; Jaeger, H.; Kosuchinas, O.; Rincon, M.; Šakėnas, V.; Vaskevicius, N. Stepping Forward Through Echoes of the Past: Forecasting with Echo State Networks. 2007. Available online: https://www.academia.edu/download/81405852/27-NN3_Herbert_Jaeger_report.pdf (accessed on 10 September 2026).
10. Jaeger, H. Adaptive nonlinear system identification with echo state networks. *Adv. Neural Inf. Process. Syst.* **2002**, *15, 609-616.*
11. Lazar, A.; Pipa, G.; Triesch, J. Fading memory and time series prediction in recurrent networks with different forms of plasticity. *Neural Netw.* **2007**, *20*, 312–322. https://doi.org/10.1016/j.neunet.2007.04.020.
12. Rosselló, J.L.; Alomar, M.L.; Morro, A.; Oliver, A.; Canals, V. High-Density Liquid-State Machine Circuitry for Time-Series Forecasting. *Int. J. Neural Syst.* **2016**, *26*, 1550036. https://doi.org/10.1142/s0129065715500367.

13. Pathak, J.; Hunt, B.; Girvan, M.; Lu, Z.; Ott, E. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Phys. Rev. Lett.* **2018**, *120*, 024102. https://doi.org/10.1103/PhysRevLett.120.024102.
14. Weng, T.; Yang, H.; Gu, C.; Zhang, J.; Small, M. Synchronization of chaotic systems and their machine-learning models. *Phys. Rev. E* **2019**, *99*, 042203.
15. Jiang, J.; Lai, Y.-C. Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius. *Phys. Rev. Res.* **2019**, *1*, 033056.
16. Borra, F.; Vulpiani, A.; Cencini, M. Effective models and predictability of chaotic multiscale systems via machine learning. *Phys. Rev. E* **2020**, *102*, 052203.
17. Zhang, C.; Jiang, J.; Qu, S.-X.; Lai, Y.-C. Predicting phase and sensing phase coherence in chaotic systems with machine learning. *Chaos Interdiscip. J. Nonlinear Sci.* **2020**, *30*, *083114*.
18. Pyragas, V.; Pyragas, K. Using reservoir computer to predict and prevent extreme events. *Phys. Lett. A* **2020**, *384*, 126591.
19. Chowdhury, S.N.; Ray, A.; Mishra, A.; Ghosh, D. Extreme events in globally coupled chaotic maps. *J. Phys. Complex.* **2021**, *2*, 035021.
20. Doan, N.A.K.; Polifke, W.; Magri, L. Short-and long-term predictions of chaotic flows and extreme events: A physics-constrained reservoir computing approach. *Proc. R. Soc. A* **2021**, *477*, 20210135.
21. Roy, M.; Senapati, A.; Poria, S.; Mishra, A.; Hens, C. Role of assortativity in predicting burst synchronization using echo state network. *Phys. Rev. E* **2022**, *105*, 064205.
22. Escalona-Morán, M.A.; Soriano, M.C.; Fischer, I.; Mirasso, C.R. Electrocardiogram Classification Using Reservoir Computing With Logistic Regression. *IEEE J. Biomed. Health Inform.* **2015**, *19*, 892–898. https://doi.org/10.1109/JBHI.2014.2332001.
23. Al Zoubi, O.; Awad, M.; Kasabov, N.K. Anytime multipurpose emotion recognition from EEG data using a Liquid State Machine based framework. *Artif. Intell. Med.* **2018**, *86*, 1–8. https://doi.org/10.1016/j.artmed.2018.01.001.
24. Das, A.; Pradhapan, P.; Groenendaal, W.; Adiraju, P.; Rajan, R.T.; Catthoor, F.; Schaafsma, S.; Krichmar, J.L.; Dutt, N.; Van Hoof, C. Unsupervised heart-rate estimation in wearables with Liquid states and a probabilistic readout. *Neural Netw.* **2018**, *99*, 134–147. https://doi.org/10.1016/j.neunet.2017.12.015.
25. Buteneers, P.; Verstraeten, D.; van Mierlo, P.; Wyckhuys, T.; Stroobandt, D.; Raedt, R.; Hallez, H.; Schrauwen, B. Automatic detection of epileptic seizures on the intra-cranial electroencephalogram of rats using reservoir computing. *Artif. Intell. Med.* **2011**, *53*, 215–223. https://doi.org/10.1016/j.artmed.2011.08.006.
26. Ghosh, S.; Senapati, A.; Mishra, A.; Chattopadhyay, J.; Dana, S.K.; Hens, C.; Ghosh, D. Reservoir computing on epidemic spreading: A case study on COVID-19 cases. *Phys. Rev. E* **2021**, *104*, 014308.
27. Verstraeten, D.; Schrauwen, B.; Stroobandt, D.; Van Campenhout, J. Isolated word recognition with the Liquid State Machine: A case study. *Inf. Process. Lett.* **2005**, *95*, 521–528. https://doi.org/10.1016/j.ipl.2005.05.019.
28. Skowronski, M.D.; Harris, J.G. Minimum mean squared error time series classification using an echo state network prediction model. In Proceedings of the 2006 IEEE International Symposium on Circuits and Systems, Kos, Greece, 21–24 May 2006; pp. 3153–3156.
29. Krishnagopal, S.; Aloimonos, Y.; Girvan, M. Similarity learning and generalization with limited data: A reservoir computing approach. *Complexity* **2018**, *2018*, 6953836.
30. Schaetti, N.; Salomon, M.; Couturier, R. Echo State Networks-Based Reservoir Computing for MNIST Handwritten Digits Recognition. In Proceedings of the 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), Paris, France, 24–26 August 2016; pp. 484–491.
31. Wijesinghe, P.; Srinivasan, G.; Panda, P.; Roy, K. Analysis of Liquid Ensembles for Enhancing the Performance and Accuracy of Liquid State Machines. *Front. Neurosci.* **2019**, *13*, 504. https://doi.org/10.3389/fnins.2019.00504.
32. Woo, J.; Kim, S.H.; Kim, H.; Han, K. Characterization of the neuronal and network dynamics of liquid state machines. *Phys. A Stat. Mech. Its Appl.* **2024**, *633*, 129334. https://doi.org/10.1016/j.physa.2023.129334.
33. Woo, J.; Kim, H.; Kim, S.H.; Han, K. Echo State Property upon Noisy Driving Input. *Complexity* **2024**, *2024*, 5593925. https://doi.org/10.1155/2024/5593925.
34. Frank, S.L. *Strong Systematicity in Sentence Processing by an Echo State Network*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 505–514.
35. Tong, M.H.; Bickett, A.D.; Christiansen, E.M.; Cottrell, G.W. Learning grammatical structure with Echo State Networks. *Neural Netw.* **2007**, *20*, 424–432. https://doi.org/10.1016/j.neunet.2007.04.013.

36. Hinaut, X.; Dominey, P.F. Real-Time Parallel Processing of Grammatical Structure in the Fronto-Striatal System: A Recurrent Network Simulation Study Using Reservoir Computing. *PLoS ONE* **2013**, *8*, e52946. https://doi.org/10.1371/journal.pone.0052946.
37. Schaetti, N. Behaviors of Reservoir Computing Models for Textual Documents Classification. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–7.
38. Cabessa, J.; Hernault, H.; Kim, H.; Lamonato, Y.; Levy, Y.Z. Efficient Text Classification with Echo State Networks. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
39. Ueda, R.; Kuribayashi, T.; Kando, S.; Inui, K. Syntactic Learnability of Echo State Neural Language Models at Scale. *arXiv* **2025**, arXiv:2503.01724.
40. Köster, F.; Uchida, A. Reservoir computing as a language model. *Phys. Rev. Appl.* **2026**, *26*, 024051. https://doi.org/10.1103/sd11-x3ny.
41. Shakespeare, W. *Shakespeare's Plays, Sonnets and Poems from the Folger Shakespeare*; Mowat, B., Werstine, P., Poston, M., Niles, R., Eds.; Folger Shakespeare Library: Washington, DC, USA. Available online: https://www.folger.edu/explore/shakespeares-works/all-works/ (accessed on 13 September 2026).
42. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **2015**, *10*, e0130140.
43. Landt-Hayen, M.; Kröger, P.; Claus, M.; Rath, W. Layer-wise Relevance Propagation for Echo State Networks applied to Earth System Variability. In Proceedings of the 3rd International Conference on Machine Learning Techniques (MLTEC 2022), Zurich, Switzerland, 19–20 November 2022; pp. 115–130.
44. Jurafsky, D.; Martin, J.H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Available online: https://aclanthology.org/anthology-files/pdf/J/J00/J00-4006.pdf (accessed on 10 September 2026).
45. E. W. A proposal on machine learning via dynamical systems. *Commun. Math. Stat.* **2017**, *5*, 1–11.
46. Chen, R.T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D.K. Neural ordinary differential equations. *Adv. Neural Inf. Process. Syst.* **2018**, *31, 6571-6583.*
47. Haber, E.; Ruthotto, L. Stable architectures for deep neural networks. *Inverse Probl.* **2017**, *34*, 014004.
48. Chang, B.; Meng, L.; Haber, E.; Ruthotto, L.; Begert, D.; Holtham, E. Reversible architectures for arbitrarily deep residual neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
49. Ruthotto, L.; Haber, E. Deep Neural Networks Motivated by Partial Differential Equations. *J. Math. Imaging Vis.* **2020**, *62*, 352–364. https://doi.org/10.1007/s10851-019-00903-1.
50. Zhu, M.; Chang, B.; Fu, C. Convolutional neural networks combined with Runge–Kutta methods. *Neural Comput. Appl.* **2023**, *35*, 1629–1643. https://doi.org/10.1007/s00521-022-07785-2.
51. Lu, Y.; Li, Z.; He, D.; Sun, Z.; Dong, B.; Qin, T.; Wang, L.; Liu, T.-Y. Understanding and Improving Transformer from a Multi-Particle Dynamic System Point of View. In Proceedings of the ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations, Online, 26 April 2020.
52. Dutta, S.; Gautam, T.; Chakrabarti, S.; Chakraborty, T. Redesigning the transformer architecture with insights from multi-particle dynamical systems. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 5531–5544.
53. Geshkovski, B.; Letrouit, C.; Polyanskiy, Y.; Rigollet, P. The emergence of clusters in self-attention dynamics. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 57026–57037.
54. Geshkovski, B.; Letrouit, C.; Polyanskiy, Y.; Rigollet, P. A mathematical perspective on transformers. *Bull. Am. Math. Soc.* **2025**, *62*, 427–479.
55. Bruno, G.; Pasqualotto, F.; Agazzi, A. Emergence of meta-stable clustering in mean-field transformer models. In Proceedings of the Thirteenth International Conference on Learning Representations, Singapore, 24–28 April 2025.
56. Rigollet, P. The Mean-Field Dynamics of Transformers. *arXiv* **2025**, arXiv:2512.01868.
57. Kim, H.; Kang, S.; Choi, Y.; Ji, S.; Woo, J.; Chung, H.; Han, S.C.; Han, K. Physics-based phenomenological characterization of cross-modal bias in multimodal models. *arXiv* **2026**, arXiv:2602.20624.

58. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442. https://doi.org/10.1038/30918.
59. Barabási, A.-L.; Albert, R. Emergence of Scaling in Random Networks. *Science* **1999**, *286*, 509–512. https://doi.org/10.1126/science.286.5439.509.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.