

Memory and criticality in Recurrent Neural Networks

Author

Naima Elosegui Borras

Institute of Neuroinformatics, University of Zurich and ETH Zurich

Advisors

Prof. Remi Monasson

École Normale Supérieure, Department of Statistical Physics

Prof. Benjamin Grewe

Institute of Neuroinformatics, University of Zurich and ETH Zurich

Dr. Pau Vimellis Aceituno

Institute of Neuroinformatics, University of Zurich and ETH Zurich

Neural Systems and Computation MSc Thesis

14 September 2022

Abstract

In what parameter regime does a complex network operate optimally? This remains an open question of great scientific interest. A well studied property in biological and artificial networks, particularly in recurrent neural networks (RNNs), is the notion of criticality, namely when a network transitions from ordered to chaotic dynamics. Near criticality, RNNs typically demonstrate enhanced memory and computational capabilities. Determining how close to the critical line and at what side of it a network must operate differs across networks and tasks. We show experimentally that being below the critical line (sub-critical), towards the ordered regime, significantly improves short-term memory (STM) and non-linear computational capabilities in Reservoir Computing (RC) networks, namely a subset of RNNs. These experimental findings match the initial hypothesis that for STM, one wants the output of the network to be more driven by the input (ordered phase) than by network dynamics (chaotic phase). Moreover, we formulate a theoretical framework for how an input affects network dynamics in an RC network and in an associative memory network. In accordance with experimental results, theoretical results also suggest that computational capabilities are enhanced in sub-critical regimes for low-connectivity networks.

CONTENTS

1	Introduction	3
2	Criticality analysis in a Reservoir Computing setting	5
2.1	Methods	5
2.1.1	Network definition	5
2.1.2	Tasks and training	6
2.2	Theoretical Results	6
2.2.1	Finding the critical line	6
2.2.2	Effect of input on the randomly connected network	9
2.3	Experimental results	14
2.3.1	Result consistency across the critical line	14
2.3.2	MI scores for direct memory task	17
2.3.3	MI scores for 3-bit parity task	19
3	Analysis of content-addressable memory network	20
3.1	Network definition	20
3.2	Adding input to the network	21
3.3	Criticality definition	22
4	Discussion	23
5	Conclusions	25
6	Appendix A: Additional results for the direct memory retrieval task	26
7	Appendix B: Additional results for the non-linear computation task	40
8	Appendix C: Additional results for P_{aff} and N_{aff}	54
9	Acknowledgements	56

1 INTRODUCTION

A key question when designing a network is how to initialize the parameters to optimally perform a set of tasks. An artificial neural network (ANN) can be studied as a dynamical system, where ordered and chaotic dynamics can be identified. In the ordered regime, small network perturbations quickly decay, whereas in the chaotic regime these perturbations are maintained and amplified [1]. The critical regime corresponds to the phase transition from ordered to chaotic dynamics, and is widely studied in the context of biological systems [2] and artificial recurrent neural networks (RNNs) [3] [4] [5]. RNNs are a subset of neural networks with connectivity between and within layers [6]. In an ordered state, a network accurately encodes the input it receives at a given time step, yet the trace of previous inputs decays fast. The converse occurs in the chaotic state, where the network is very sensitive to noise in the input and is dominated by its internal dynamics. Criticality (also referred to as “edge-of-chaos”) is thus optimal for network computations [5] [7], information transfer [1] and memory capacity [8]. Theoretical results demonstrate that for certain RNNs, being close to the critical line but beyond the “edge-of-chaos” can be beneficial for temporal integration tasks [9]. Furthermore, the notion of criticality is crucial when initializing deep neural networks (DNNs), where the network is trainable if its parameters are initialized sufficiently close to the “edge-of-chaos” transition [10]. Initializing DNNs close to criticality enforces the variance of the activation of every layer to remain constant, avoiding exploding or vanishing gradients, in particular, for hyperbolic tangent activation functions, this initialization scheme is known as the Xavier initialization [11]. Returning to the initial question, initializing the parameters of a network “near” criticality is a good choice, now the challenge of defining “near” comes into play. It is therefore paramount to research the critical regime to improve our understanding of what classes of biological and artificial neural networks benefit from its dynamical properties across tasks, such as memory storage and retrieval.

One of the main motivations behind studying criticality in RNNs is its potential link to criticality in the brain. The notion of criticality can be extended to biological systems, where evidence suggests that a system composed of similar units, such as networks of neurons in the brain, could be operating at criticality [2]. The main experimental indicator of criticality in the brain is given by the power-law distribution of neuronal avalanches [12], namely bursts in neural activity. Power-law distributions are observed in the parameters of systems that are at criticality [13], yet it has been shown that power-law distributions can also emerge in systems that are away from criticality [14] and, often times, the data to prove this relationship is insufficient [15] [16]. Nevertheless, these observations encourage future investigations into accurately identifying criticality in the brain. To ensure that the power-law behaviours observed experimentally in neural data are actually due to criticality, three conditions must be met [13]: 1) the power-law should be present only when the system is at criticality, thus tuning parameters to be in the super- or sub-critical regimes should make this relationship disappear; 2) the exponent of the power-law of different parameters should be related, as power-laws appear at criticality for different network parameters; and 3) there should exist a data collapse in the experimental data. Data collapse means that rescaling average neuronal avalanche shapes across different avalanche durations, using the appropriate critical exponent, should make the curves match. Having these criteria in mind, experimental data can be analysed more rigorously for trademarks of criticality.

Human recordings using electrocorticographic data have demonstrated that the difference between sleep/anesthesia and wakefulness can be seen in the dynamics of the temporal data shifting from stable to fluctuations around the line of instability, suggesting criticality-like hallmarks in consciousness states [17]. Criticality in the brain has been extensively studied in the cortex, where it has been hypothesized that the cortex operates near criticality, thereby improving its functional properties, namely dynamic range and information processing and storage [18]. Parallels between criticality in the mouse visual cortex and ANNs trained to classify images have been drawn [4], where criticality in this particular case is defined by looking at the eigenspectrum of the covariance matrix of each of the networks. Overall, stronger experimental evidence is needed to determine which regions or states in biological neural networks operate at criticality and what beneficial properties can be drawn from this regime.

To recapitulate, the notion of criticality is equivalent across biological and artificial networks, yet how to define criticality differs. In biological neural networks, it is common to observe neural avalanches following

an inverse power-law distribution when the network is operating at criticality [19], nevertheless power-law distributions are also observed when operating out of the critical regime [14]. In the RC setting, particularly when considering Echo State Networks (ESNs), an important parameter to consider is the spectral radius $\rho(W)$ of the fixed weight matrix W , namely the magnitude of the maximal eigenvalue of the reservoir's weight matrix [4]. ESNs can exhibit the Echo State Property (ESP), where different initial configurations will converge to a given state for different inputs into the reservoir [1], analogous to a network operating in the ordered regime. It is often the case that when $\rho(W) < 1$ the ESP is preserved (ordered state) and when $\rho(W) > 1$ the ESP does not hold (chaotic state), yet it is a common misconception that this condition is sufficient or necessary to obtain the ESP [20]. It is therefore not sufficient to scale the weight matrix of the reservoir to obtain a value of $\rho(W) = 1$ to be at criticality. A theoretical analysis of sufficiency conditions for the ESP was determined by *Yildiz et al.*, [20] and should be considered when designing a network to operate in the critical regime. Finally, another approach taken to identify criticality in RNNs is to analyze how small network perturbations evolve in time, where fading occurs in the ordered regime and amplification in the chaotic one [5].

Thinking about criticality in networks that have a series of attractor states, such as Hopfield networks [21] or variants [22] of it, is more complex, as these networks converge towards different equilibrium states. The dynamics of these associative memory networks, which can be constructed by Hebbian learning, drive the network towards a stored memory or pattern given an input [23]. Recent work has shown that alternating phases of Hebbian learning and short-term synaptic dynamics or homeostatic adaptation can lead to criticality in associative memory networks, without losing memory storage [23]. Furthermore, criticality, in the form of power-law distributions of activity avalanche duration and size, has been shown to arise in self-organizing recurrent networks (SORNs), which combine Hebbian and homeostatic plasticity learning, when the network is driven by random inputs [24]. Theoretically defining what ranges of parameters (such as sparsity, number of neurons or patterns stored) lead to critical dynamics in associative memory networks can be used to explore how criticality affects memory retrieval or transition between attractors in these type of networks.

The question then remains of when it is beneficial to be at criticality, at what distance and on which side of the critical line. Recent work on spiking RNNs demonstrates that complex tasks do profit from criticality, whereas simpler ones do not, although information-theoretic measures demonstrate maximum network capacity at criticality in all cases [25]. Moreover, it has also been shown that *in vivo* spiking populations dynamics demonstrate sub-critical states [16]. The work in this thesis investigates the hypothesis that for short time scales (i.e., small delays between network input and output decoding) it is optimal to operate slightly below the critical line, to exploit sub-critical dynamics which are more input-driven. To explore this hypothesis, this thesis is structured in the following way:

- In Section 2, the focus is on probing the hypothesis in a theoretical and experimental way in the Reservoir Computing (RC) setting [26], which is a sub-category of RNNs. Section 2.2 develops the theoretical framework of how an input affects the network and where analytically the critical line lies. Section 2.3 shows experimental results on two memory tasks at different distances from the critical line, where being slightly below criticality, in the ordered regime, statistically significantly ameliorates short-term memory (STM) retrieval and non-linear computational capabilities.
- In Section 3, a Hopfield-like content-addressable memory [22] model is theoretically expanded to include an input stream. The criticality regime is defined and numerically simulated, based on theoretical predictions, laying the ground for future exploration into the effects of criticality in a variety of tasks, such as integration.
- In Section 4, the potential implications and analysis of this work are discussed. Future avenues of research are also discussed in the context of RNNs and criticality, following the work performed in this thesis.

The code for the analysis and results of this thesis can be found in: <https://github.com/naimaeb/Memory-and-criticality-in-Reservoir-Computing>.

2 CRITICALITY ANALYSIS IN A RESERVOIR COMPUTING SETTING

Memory can be probed in a variety of tasks and architectures. In particular, this work focuses on understanding how STM and computational capabilities are affected when a network is at different distances from the critical line.

To answer this question, both theoretical and experimental approaches were taken. These questions were explored in randomly connected networks with a trainable decoder, as in the RC setting [26] depicted in Figure 1.

RC networks are composed of an RNN reservoir and an encoding vector weighting the input, both of which are fixed at initialization, as well as a trainable output vector that weights the state of the network [8]. These networks exploit the dynamics of the reservoir, require small training datasets and are particularly good for learning dynamical systems data [27]. The choice of the neuronal activation function and network parameters play a key role in task performance. In this work, the case of binary neurons and Gaussian weights and encoding vector are considered. The choice of this regime is two-fold:

1. Comparing to and extending previous work, that was mainly performed on binary neurons in RC networks.
2. Operating with Gaussian random variables render an analytical way to find a closed-form solution of how the input affects the network.

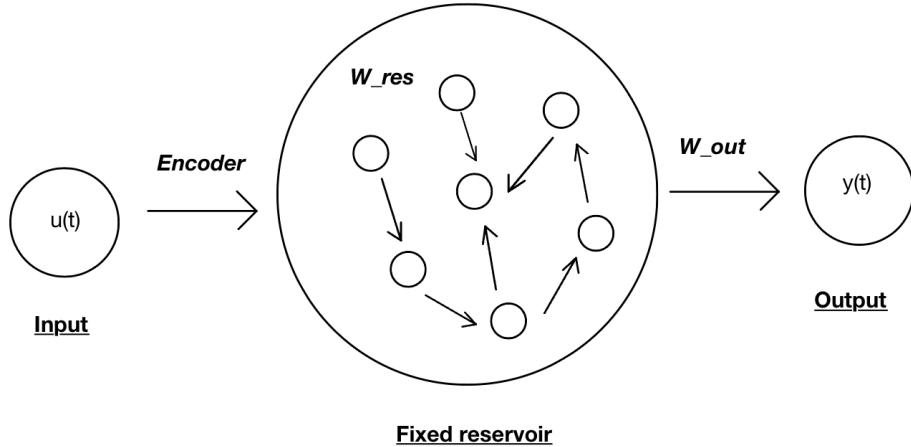


Figure 1: Reservoir Computing network used in this study, with a single input ($u(t)$) and output ($y(t)$) channel. The encoder and reservoir weights (W_{res}) are fixed at initialization, whereas the output weights (W_{out}) are trained to predict the output. Each neuron in the reservoir receives input from K other neurons and note that there are no self-connections and that the connectivity graph is directed (no symmetric weights).

To assess memory in randomly connected reservoir networks, two tasks are devised, the first one being a direct memory retrieval task and the second one a 3-bit parity task. Furthermore, a theoretical framework of when the trace of an input would fade in the system is also formulated, to then compare to experimental results.

2.1 METHODS

2.1.1 NETWORK DEFINITION

The randomly connected recurrent network is defined similar to the network in *Bertschinger et al.*, [5], where the connectivity matrix is a K-regular graph and the input is scaled by a Gaussian encoding vector.

where the second equality holds by symmetry. Equation 9 can be solved analytically by rewriting α and β as standard normal Gaussian variables where $\alpha' = \frac{\alpha}{\sigma_\alpha}$ and $\beta' = \frac{\beta}{\sigma_\beta}$, where σ_α and σ_β are the standard deviation, or square root of the variance, of α and β , respectively. Then the product of the density functions α' and β' renders a bivariate Gaussian density function, where

$$\begin{pmatrix} \alpha' \\ \beta' \end{pmatrix} \sim N \left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right]$$

Equation 9 can then be rewritten in terms of α' and β' , where $d\alpha = \sigma_\alpha d\alpha'$ and $d\beta = \sigma_\beta d\beta'$, thus obtaining

$$\begin{aligned} P(1) &= 2 \int_{-\infty}^0 \int_{\frac{-\alpha' \sigma_\alpha}{\sigma_\beta}}^{\frac{\alpha' \sigma_\alpha}{\sigma_\beta}} \phi(\alpha'; 0, 1) \phi(\beta'; 0, 1) d\beta' d\alpha' \\ &= 2 \int_{-\infty}^0 \int_{\frac{-\alpha' \sigma_\alpha}{\sigma_\beta}}^{\frac{\alpha' \sigma_\alpha}{\sigma_\beta}} \frac{1}{2\pi} e^{-\frac{1}{2}(\alpha'^2 + \beta'^2)} d\beta' d\alpha' \end{aligned} \quad (10)$$

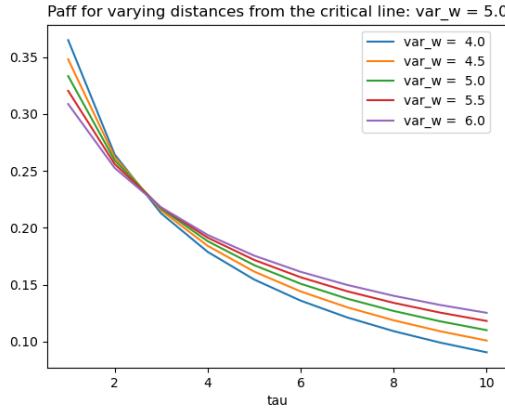
As the bivariate normal distribution composed of two uncorrelated standard normal distributions is symmetric about the x and y axis, the integral in 10 can be solved by finding the area spanned by the bivariate normal distribution in the sector of the plane given by the angle 2θ , where θ is the angle between the β' axis and the line $\beta' = \frac{\alpha' \sigma_\alpha}{\sigma_\beta}$. Thus, $\theta = \arctan\left(\frac{\sigma_\beta}{\sigma_\alpha}\right)$ and

$$\begin{aligned} P(1) &= \frac{2\theta}{\pi} \\ &= \frac{2 \arctan\left(\frac{\sigma_\beta}{\sigma_\alpha}\right)}{\pi} \end{aligned} \quad (11)$$

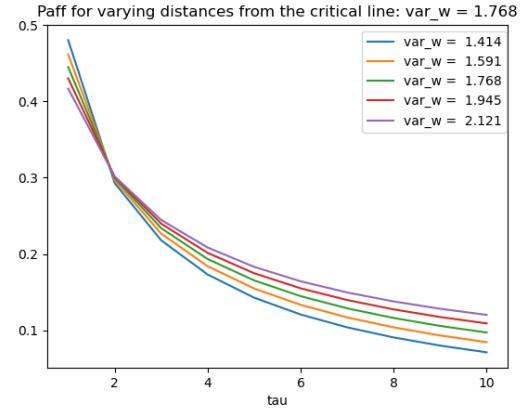
Therefore, the analytical solution for the criticality condition in Equation 11 in terms of σ_w and σ_e is given by

$$\sigma_w = \frac{s\sigma_e}{\sqrt{1 - Ks^2 + s^2}} \quad (12)$$

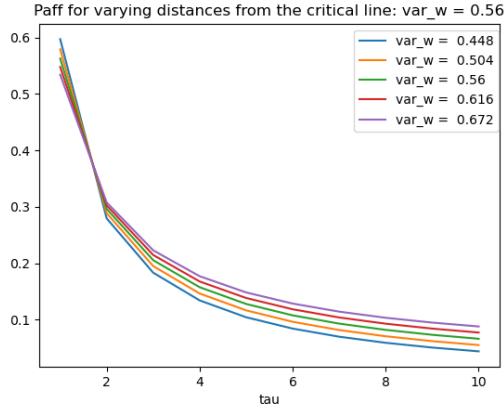
where $s = \tan\left(\frac{\pi}{2K}\right)$.



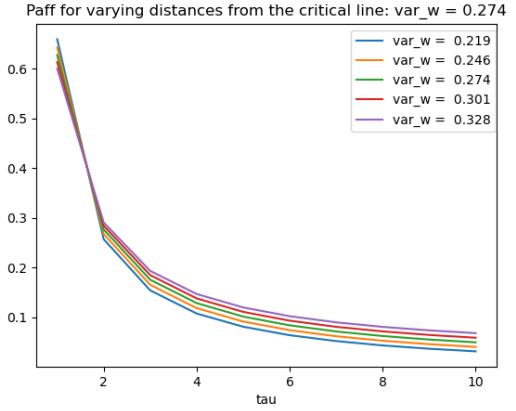
(a) $K = 3$



(b) $K = 4$



(c) $K = 6$



(d) $K = 8$

Figure 2: Probability of a neuron being affected by the input τ time steps ago. Different connectivity degrees are displayed, which individually determine what is the critical value of σ_w^2 . Sub-critical (blue and orange), critical (green) and super-critical (red and purple) are depicted. All cases are considered at $\sigma_e^2 = 5$

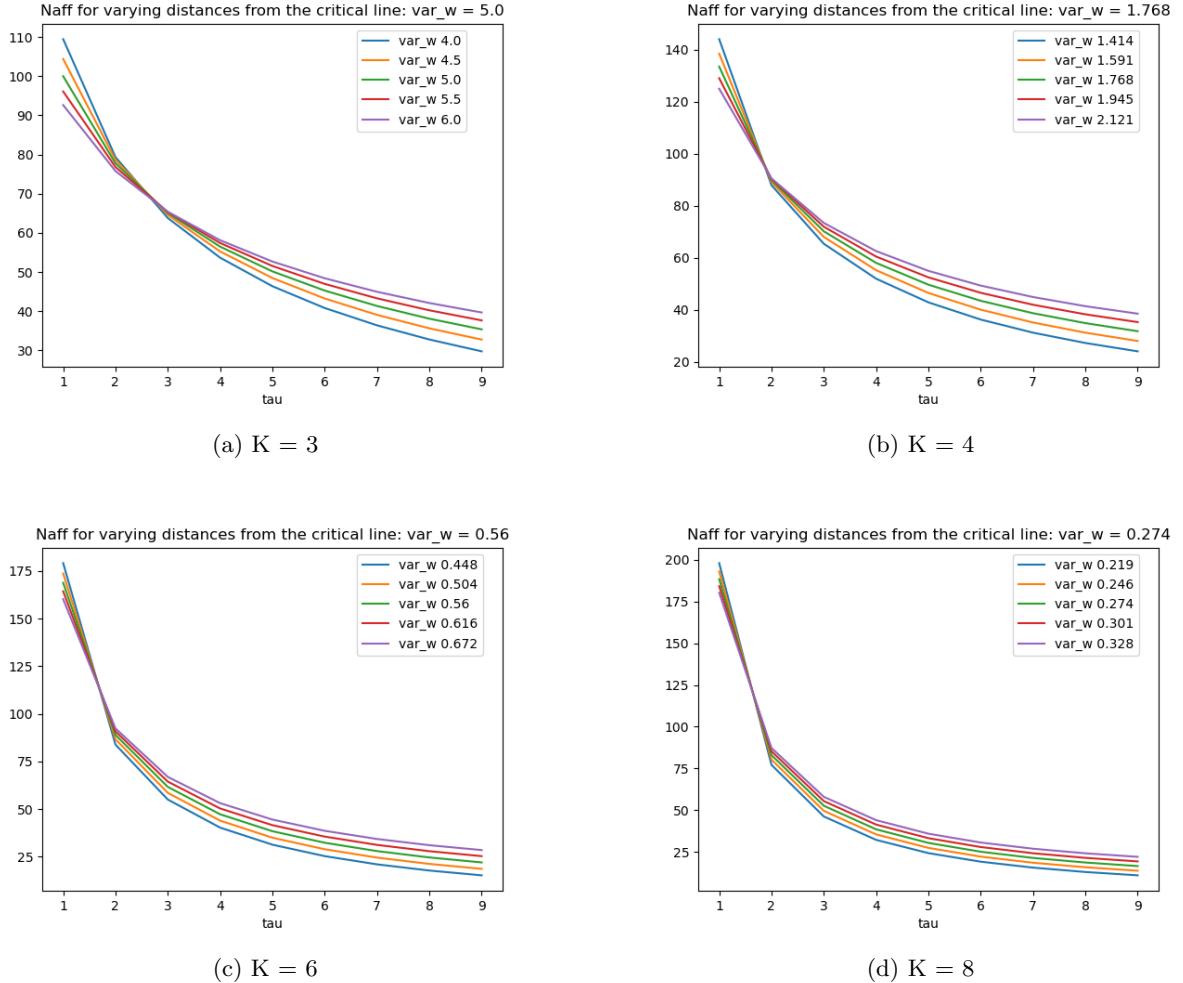


Figure 3: Expected number of neurons affected by the input τ time steps ago. Different connectivity degrees are displayed, which individually determine what is the critical value of σ_w^2 . Sub-critical (blue and orange), critical (green) and super-critical (red and purple) are depicted. All cases are considered at $\sigma_e^2 = 5$ and the number of neurons is 300.

2.3 EXPERIMENTAL RESULTS

To assess what role criticality plays in memory in random networks, the network was trained on two distinct tasks: direct memory retrieval and a 3-bit parity task. The direct memory task aims to show how the dynamics of the network will affect memory encoding of a particular input τ time steps ago, whereas the parity task demonstrates the computational capabilities of the network at different distances from the critical line. For each of the tasks, the MI and mean accuracy scores¹ were used to quantify at what distance from criticality decoding accuracy is significantly higher, at the 95% significance level. The full set of results (over a wide range of parameters) and their statistical significance is depicted in Appendix A for the direct memory task and in Appendix B for the parity task.

2.3.1 RESULT CONSISTENCY ACROSS THE CRITICAL LINE

The behavior across the critical line is expected to be the same for the range of parameters that satisfy the relationship in Equation 12. Furthermore, taking the log of the parameters σ_e^2 and σ_w^2 will render a linear relationship with a slope of 1, thus it is expected that the behavior across this line is invariant. To ensure that this is true and that the criticality criterion is correctly identified, we looked at the results of training a network on the direct memory task for a range of σ_w^2 and σ_e^2 for different delay values τ . MI was used to quantify the prediction performance, as depicted in Figure 4 (for a network connectivity of $K = 4$) and in Figure 5 (for a network connectivity of $K = 8$). Although there is some noise associated to numerical simulations, these results demonstrate that the critical line is correctly defined by Equation 12, as the MI score is mostly invariant across the critical line. Now the interesting question lies in the detailed information that can be extracted across the critical line, so we perform a more in-depth analysis of the performance near the criticality criterion on a direct memory retrieval (Section 2.3.2) and a non-linear computation task (Section 2.3.3).

¹Note that when a value of “nan” is seen in the p-value table, this is caused by the variance of the sampled data being 0 (namely all network runs have the same score, corresponding to 1). In this case, the score for the criticality and near-criticality variables is the same.

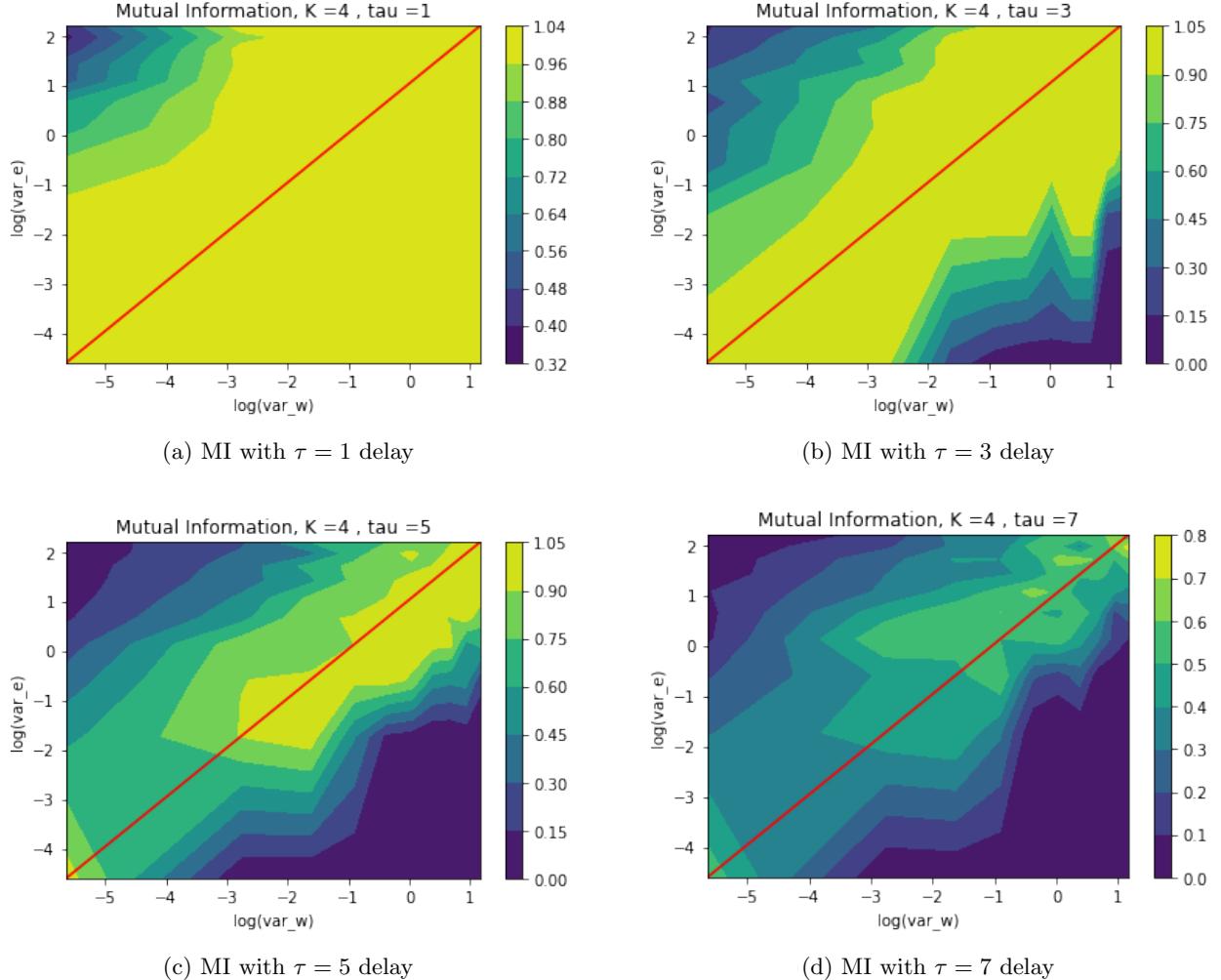


Figure 4: log-log plot of MI score across the critical line. A range of values of σ_e^2 (log value on y-axis) and σ_w^2 (log value on x-axis) were studied. The connectivity of the network was given by $K = 4$. For a range of delays τ the MI on the direct-memory task was quantified on the test set.

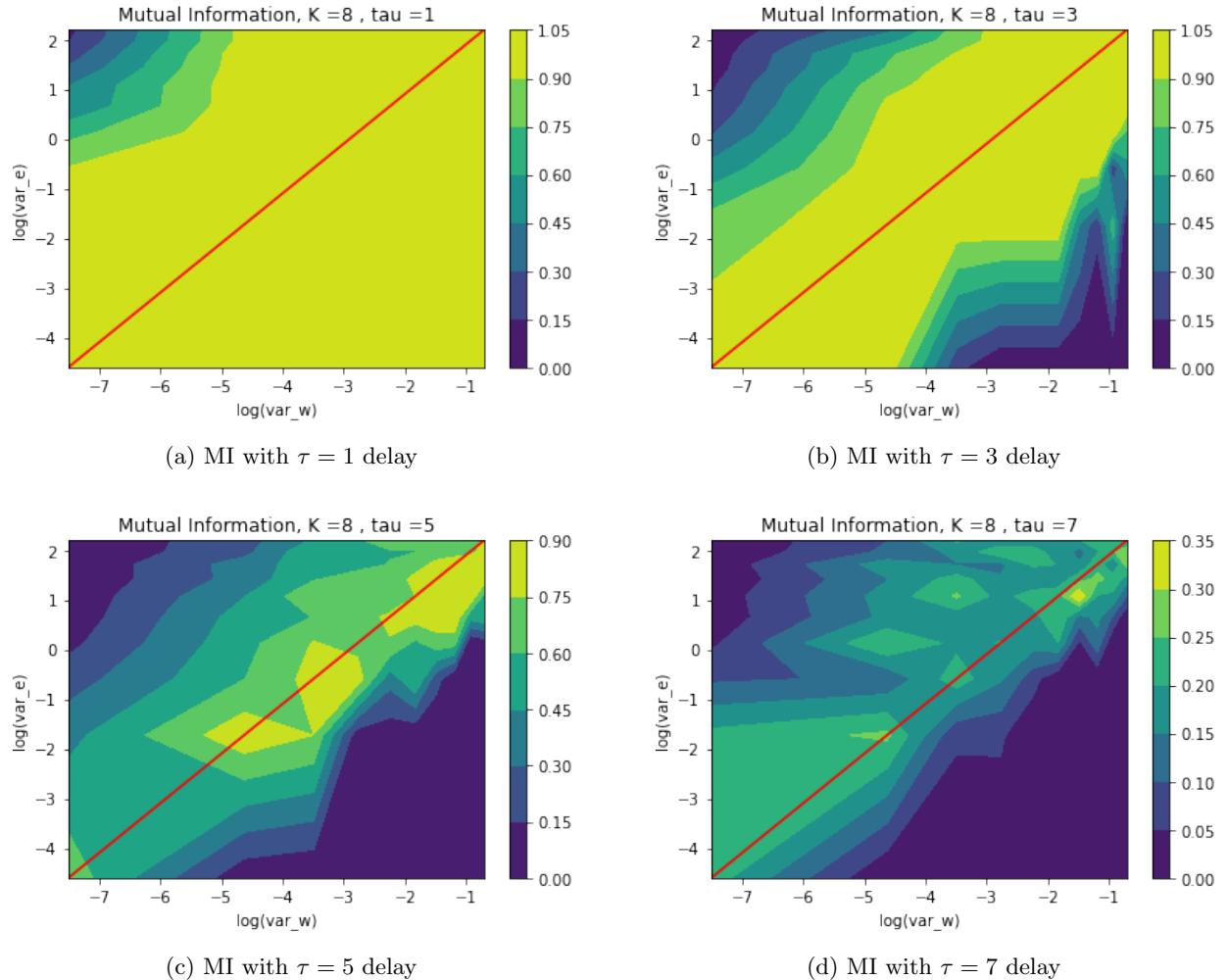


Figure 5: log-log plot of MI score across the critical line with $K = 8$ on the direct memory task.

2.3.2 MI SCORES FOR DIRECT MEMORY TASK

The first task used to study how sub-criticality affects STM is that of memory retrieval, namely how well can the network recall the input it was given τ time steps ago. Figures 6 and 7 depict the decoding performance, measured by the MI score between the input τ time steps ago and the predicted input in sub-critical (blue), critical (orange) and super-critical (green) regimes. A variety of network connectivity (K) cases are tested. In figures 6 and 7 the left panels depict the MI scores for all delays τ that were tested, but for short time scales ($\tau \in [3, 5]$), as viewed in the zoomed-in right panels, sub-criticality shows to be beneficial for STM retrieval. In particular, one can notice how for short delays, sub-criticality is beneficial, whereas for longer delays critical and super-critical regimes allow for enhanced memory retrieval. These results are statistically significant at a 95% confidence level, as further shown in Appendix A, where a wider range of parameters are tested as well.

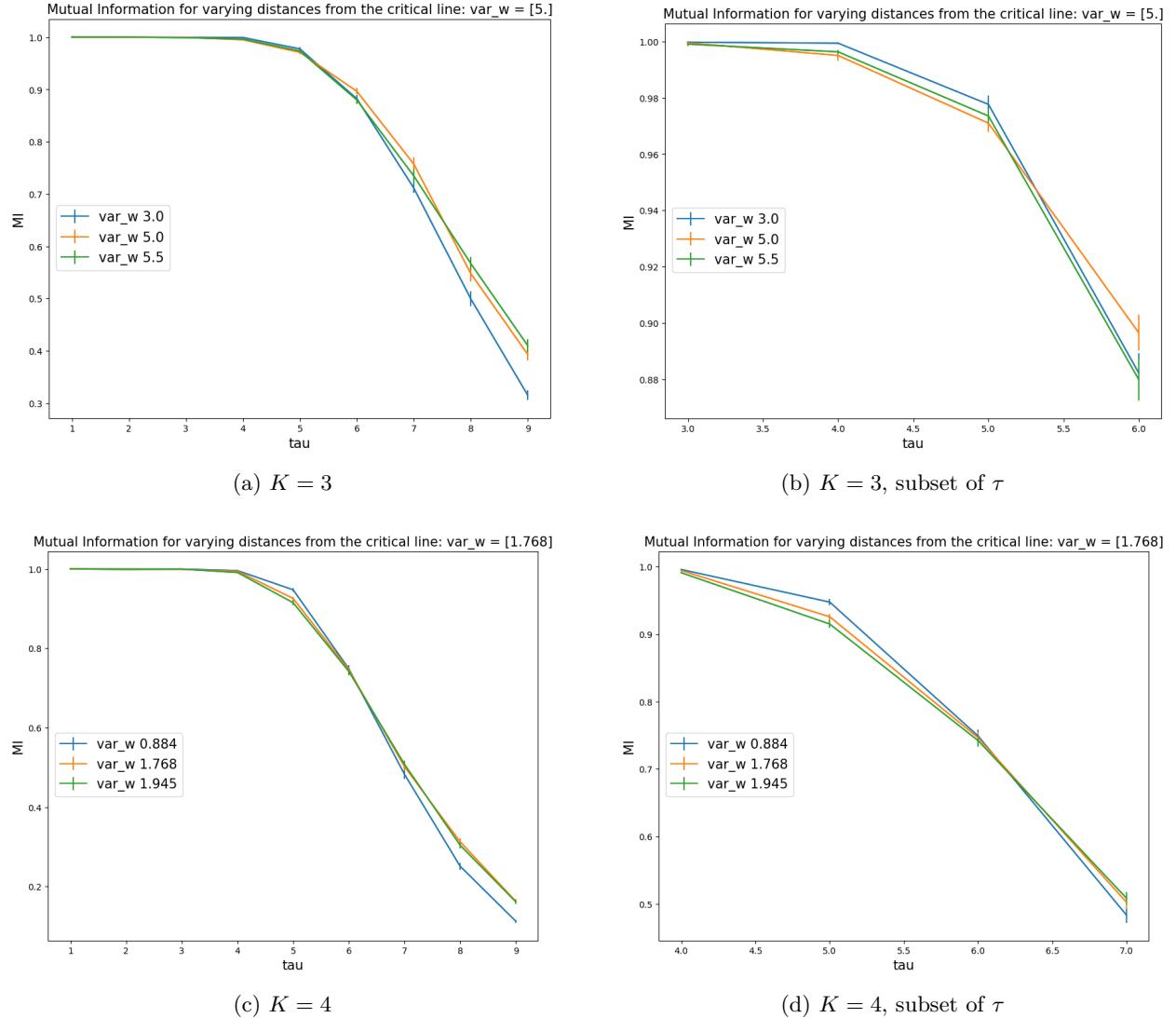


Figure 6: MI results on the direct memory task for a range of time delays τ . $\sigma_e^2 = 5$ for all simulations. The mean MI score and standard error of the mean for each trial are plotted ($N = 300$ neurons; 50 trials) Different σ_w^2 values determine whether a network is sub-critical (blue), critical (orange) or super-critical (green). (a) MI score on the test set for a network connectivity of $K = 3$ and all values of τ tested. (b) Subset of τ values from sub-figure (a). (c) MI score on the test set for a network connectivity of $K = 4$ and all values of τ tested. (d) Subset of τ values from sub-figure (c).

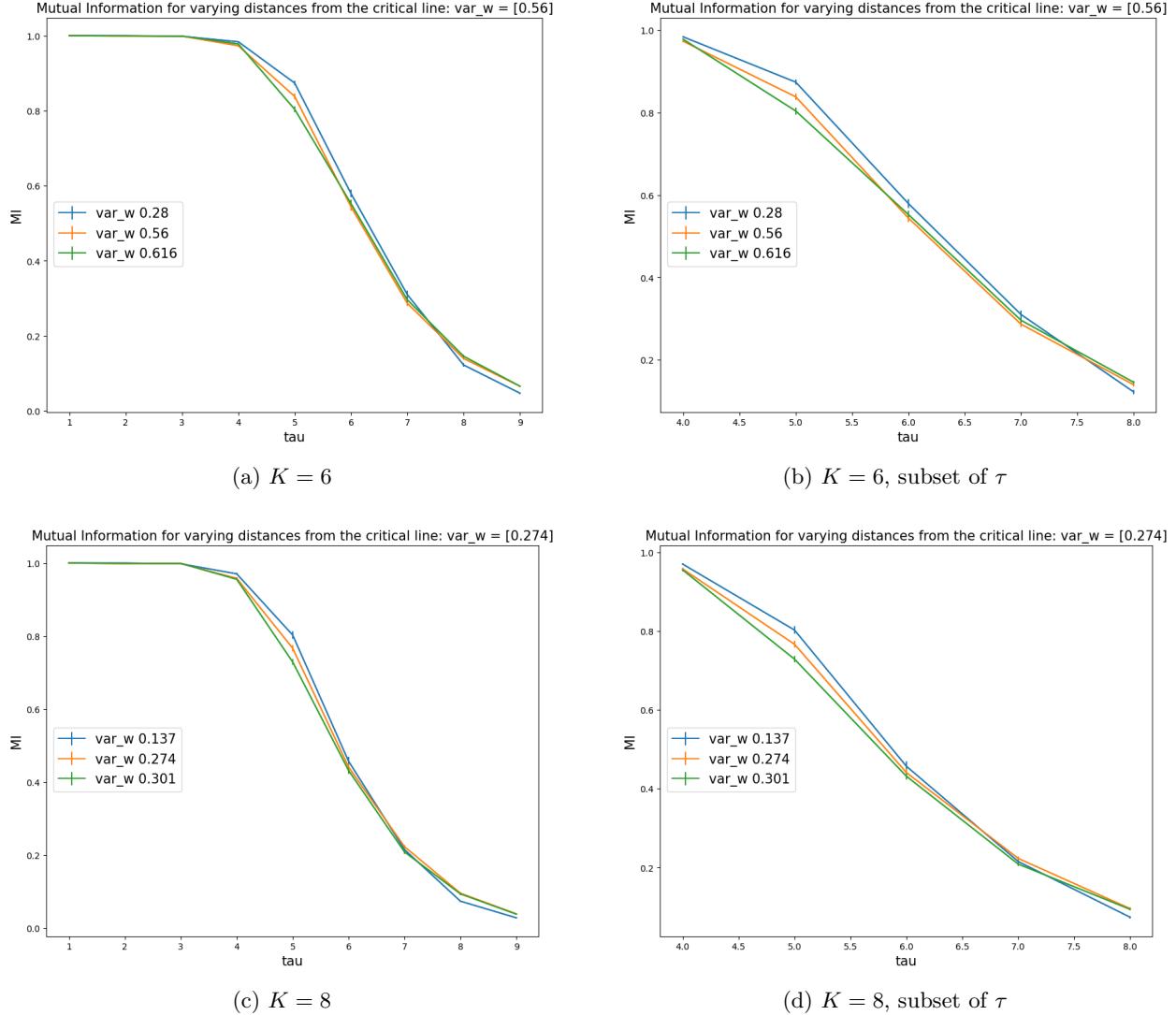


Figure 7: MI results on the direct memory task for a range of time delays τ . $\sigma_e^2 = 5$ for all simulations. The mean MI score and standard error of the mean for each trial are plotted ($N = 300$ neurons; 50 trials). Different σ_w^2 values determine whether a network is sub-critical (blue), critical (orange) or super-critical (green). (a) MI score on the test set for a network connectivity of $K = 6$ and all values of τ tested. (b) Subset of τ values from sub-figure (a). (c) MI score on the test set for a network connectivity of $K = 8$ and all values of τ tested. (d) Subset of τ values from sub-figure (c).

2.3.3 MI SCORES FOR 3-BIT PARITY TASK

To test how sub-criticality impacts non-linear computations in the binary RC network, the network was trained to predict a 3-bit parity operation on the input stream τ time steps ago. Figure 8 demonstrates the MI score between the true 3-bit parity operation and its predicted outcome in sub-critical (blue), critical (orange) and super-critical (green and red) regimes. What can be observed from the results, which probe a variety of network connectivity degrees, is that sub-critical regimes strongly improve non-linear computations that require STM ($\tau < 8$) in the network. These results were also shown to be statistically significant at the 95% confidence level, as displayed in Appendix B, where a wider range of parameters are also tested, showing similar results. Future work aims to observe other non-linear computational tasks, perhaps including the parity of more bits, to analyse not only how sub-criticality affects non-linear computational capabilities, but to also analyse what role criticality plays in task complexity.

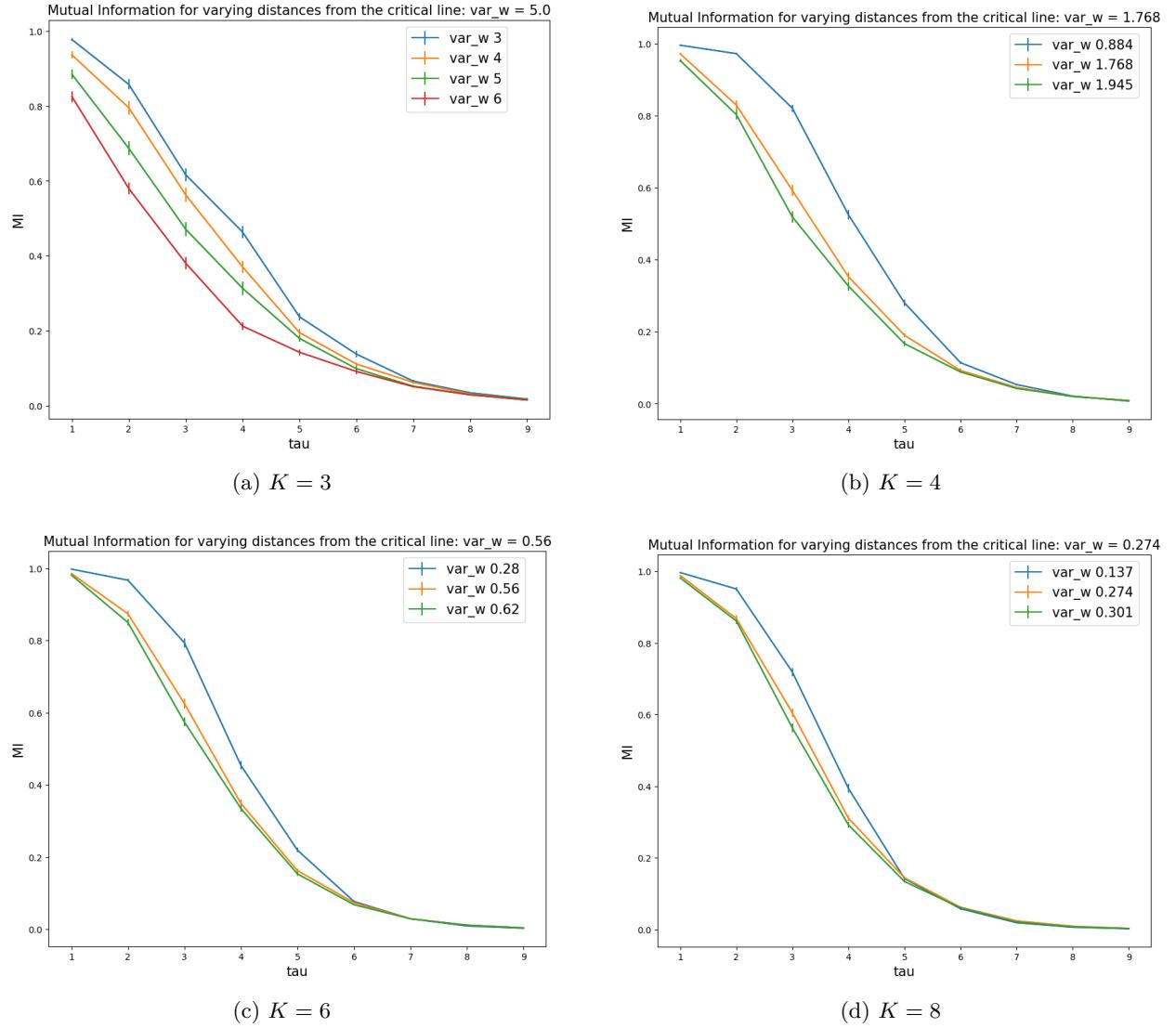


Figure 8: MI results on the 3-bit parity task for a range of time delays τ . $\sigma_e^2 = 5$ for all simulations. The mean MI score and standard error of the mean for each trial are plotted ($N = 300$ neurons; 50 trials). Different σ_w^2 values determine whether a network is sub-critical (blue), critical (orange) or super-critical (green and red). (a) Network connectivity of $K = 3$ and all values of τ tested. (b) $K = 4$. (c) $K = 6$. (d) $K = 8$.

3.3 CRITICALITY DEFINITION

In the case of the initial neuron configuration aligning macroscopically with one stored pattern, one can study the overlap $q_i(t)$ of two different network configurations $\sigma_{i1}(t)$ and $\sigma_{i2}(t)$ where

$$q_i(t) = \langle \sigma_{i1}(t) \sigma_{i2}(t) \rangle \quad (41)$$

and

$$q(t) = \frac{1}{N} \sum_{i=1}^N q_i(t) \quad (42)$$

where the update of $q(t + \Delta t)$ in the limit of $p, C \rightarrow \infty$ is given analytically by

$$q(t + \Delta t) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \exp(-y^2 - z^2) \tanh \left(\frac{(m_1 - y\sqrt{\alpha(1+q(t))} - z\sqrt{\alpha(1-q(t))})}{T} \right) \tanh \left(\frac{(m_2 - y\sqrt{\alpha(1+q(t))} - z\sqrt{\alpha(1-q(t))})}{T} \right) dy dz \quad (43)$$

where m_1 and m_2 are the magnetization along the macroscopic pattern for each of the respective patterns at time t , these two values are updated according to Equation 33 or 40 depending on whether one is working in the regime where there is an encoder or not.

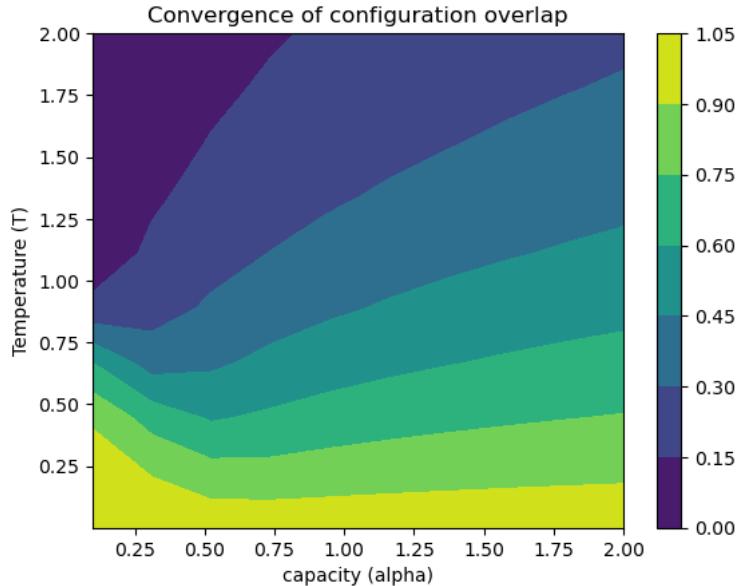


Figure 9: Configuration overlap (1- normalized hamming distance) for varying values of temperature (T) and capacity (α). The initial neuron configuration overlap was given by $q(0) = 0.8$ and the initial magnetization along the macroscopic pattern for both configurations was given by $m_1 = m_2 = 0.6$.

Figure 9 is obtained by iterating Equation 43 until the value of $q(t)$ converges. From this plot, the critical line for this network for a range of values of T and α can be determined, where a value of 1 corresponds to both configurations having the same values, hence to an ordered state, whereas when the configuration overlap is smaller than 1, this corresponds to the chaotic regime.

4 DISCUSSION

This thesis explores criticality in an RC network and demonstrates that being slightly below the critical line, that is towards the ordered regime, results in a statistically significant STM improvement on a direct memory task. When probing the computational capabilities of the network (with the 3-bit parity memory task) near criticality, the results also showed sub-criticality regimes enhanced non-linear computational capabilities. Theoretical results suggest that being sub-critical for very short time scales (delay of $\tau = 1, 2$) leads to the input being maintained longer in the system, yet for delays of $\tau > 2$, corresponding to longer memory spans, theoretical results suggest critical and super-critical regimes improve memory.

Near the critical line, an RC network maximises its information storage and transmission capabilities, as it combines the ability to transmit network perturbations into future time steps (chaos) while maintaining an accurate network representation of the input (order) [17]. If one wants to maintain input in a system for short time spans, such as in STM, one would hypothesize that being towards the ordered side of criticality would improve this property. By theoretically deriving where criticality lies in terms of network parameters in an RC network (described in Section 2), this work shows how being in a sub-critical regime (in the ordered phase) statistically improves STM retrieval and non-linear computations requiring STM.

Criticality in this RC network depends on three parameters, namely the connectivity of the network K , the variance of the encoding vector σ_e^2 and the variance of the distribution over the weights σ_w^2 , in particular, the parameters of σ_e^2 and σ_w^2 can be condensed into one single parameter, since what is important is their ratio. Strengthening σ_e^2 makes the network more sensitive to the input (ordered phase), increasing K and σ_w^2 leads to a network driven primarily by its internal dynamics (chaotic phase). By fixing K and σ_e^2 and varying σ_w^2 (for a range of K and σ_e^2 values), the results in section 2.3.2 and Appendix A consistently demonstrate that for time delays of τ between 4 and 7, being at values of 50%–90% of the critical value of σ_w^2 , statistically significantly improve direct memory retrieval. Furthermore, as both the MI and accuracy metrics agree, and statistical significance is shown in both analysis, we can assert that for short timescales, being sub-critical is optimal for STM decoding. These results agree with the notion of criticality not being optimal for simple tasks [25].

When analysing computational capabilities in the RC network, using a 3-bit parity task, the evidence demonstrating that being towards the ordered side of the critical line improves performance even more notably than in the direct memory task. For time delays of $\tau < 8$, being in the sub-critical regime statistically significantly improves prediction performance at the 95% confidence level, as depicted in Appendix B and in Section 2.3.3. Endowed with the theoretical tools to study criticality and with proof that sub-critical dynamics ameliorate STM storage and non-linear computational capabilities, it would be interesting to extend this work in the following manner:

1. Add multiple channels: Test the parity of bits from k different channels, as opposed to the one from a single channel.
2. Task difficulty: The nature of the network (binary neurons with a Heaviside activation function) makes the 3-bit parity task difficult to solve, as it is not linearly separable, nevertheless the performance is still high for short time delays. Probing more complex tasks, such as computing the parity of more bits, would be interesting to understand the role that criticality plays when analysing task complexity. Is it more beneficial to be closer or further from the critical line when task complexity is increased in these type of networks?
3. Nature of the network: looking at networks with continuous activation functions and continuous input could also be an interesting extension to this work, making it comparable to the main body of literature on RC networks. Furthermore, including trainable reservoir weights would make this study generalizable to more classes of RNNs, beyond RC networks.

The theoretical calculations and numerical simulations of the neurons affected by the input at a delay of τ , as demonstrated in the results in Section 2.2.2, show that for a maximal delay of $\tau = 2$, being sub-critical

is beneficial, conversely for $\tau > 2$ the super-critical regime allows the input to affect the network for longer. Moreover, the results demonstrate that lower network connectivity leads to a slower fading of the input in the network, particularly in the first time step. Intuitively, if the connectivity is lower, namely if the network is more sparsely connected, this could lead to less memory interference by network interactions. This is in accordance with work on reservoir and input sparsity in RC networks [31], where, in particular, input sparsity ameliorates STM capacity. An interesting extension to consider would be to control the connectivity of the input to the reservoir, re-derive the criticality setting for this conditions and then analyse STM in the network. The question remains as to whether the probability of an input affecting the network is a good proxy for analysing network computational capabilities requiring STM. To answer this question, further investigations would include:

- Accounting for the encoder strength in future time steps: although a strong encoder will lead to future inputs corrupting the input τ time steps ago, it still might be beneficial to account for the initial encoder strength when computing the probability at subsequent time steps. Perhaps one could include a decaying encoder strength term proportional to the probability of past input affecting the network.
- Modifying the metric to look at STM: One could start by explicitly tracking how an input affects the network, to check for accordance with theoretical predictions. Then one could think about why the probability of an input (τ time steps ago) affecting the network (in subsequent time steps) is incomplete to asses computations requiring STM. A quantity that could be worth investigating would be how future input overwrites past input, which will greatly depend on input nature and network parameters and its activation function.

Overall, the analysis developed to quantify the number of affected neurons by past input serves as a promising basis to theoretically explore STM in the RC setting.

Extending the work on criticality to associative memory networks could potentially benefit the performance of tasks that require attractor dynamics within a network, such as replay in networks modelling spatial navigation [32] or behavioural action sequences [33]. The theoretical and numerical results derived in Section 3 serve as a starting point to characterize criticality, given by growth or decrease in network perturbations, in these networks as a function of connectivity, patterns stored, encoder nature and temperature. Integration tasks would be particularly interesting to look at in this network, where following the initial hypothesis that STM benefits from sub-critical dynamics, short integration times spans could potentially benefit from sub-critical regimes.

5 CONCLUSIONS

In conclusion, this work shows that working in the sub-critical regime is beneficial for encoding short-term memory (STM) and performing non-linear computations in Reservoir Computing (RC) networks with binary neurons and parameters drawn from Gaussian distributions. The computational capabilities of these networks are analysed by using a parity task, where sub-critical regimes strongly outperform super-critical and critical regimes. Future experiments and theoretical formulation could, together with probing a wider range of tasks, help quantify what exact role sub-critical regimes play in the computational capabilities of networks performing tasks that require STM.

How an input affects the system over time and where the critical line lies in RC networks is analysed, providing a theoretical framework that could potentially be extended to other network classes. In particular, extending the theoretical work to RC networks with continuous activation functions and multiple input channels could help formulate experimental paradigms that further strengthen the hypothesis that STM benefits from sub-critical dynamics. The probability of a neuron being affected by the input is higher in sub-critical regimes for very short delays and lower for longer delays than in super-critical regimes, suggesting that this quantity alone is perhaps not sufficient to explain how STM evolves in RC networks. This finding motivates theoretically analysing other parameters that could affect STM, such as probability of input interference with past memories, to check for stronger agreement with experimental results. Moreover, endowed with a theoretical framework for associative memory networks, future experiments could also help show how sub-critical regimes affect integration over short time periods.

In summary, this work shows that sub-criticality is beneficial for memory retrieval and non-linear computations that require STM in RC networks. These results are supported by theoretical formulations that show that, in low-connectivity networks, operating in sub-critical regimes ameliorates STM. Overall, the theoretical and experimental framework in this study serves as a promising foundation to continue investigating what role sub-criticality plays in STM, where developing parallels with biological neural networks could be a fascinating long-term goal to pursue.

6 APPENDIX A: ADDITIONAL RESULTS FOR THE DIRECT MEMORY RETRIEVAL TASK

In this section, a wide range of parameter values are shown for the direct memory task to corroborate the results presented in Section 2.3.2. Figures 10 - 17 show the results on the test-set of training the network to perform the direct memory task for fixed values of $K \in [3, 4, 6, 8]$ and $\sigma_e^2 \in [1, 5]$ and a range of values for σ_w^2 , which correspond to values below and above the critical line. The plots show the mean over network trials and the standard error on the mean for each data point. In each Figure, panels (a) and (c) depict the results of accuracy and MI, respectively, for all values of σ_w^2 and delay τ that were tested. Panels (b) and (d) depict values of τ where one can clearly observe a difference in performance on a subset of 3 curves: performance below the critical line, at the critical line, and above the critical line. Tables 1 - 16 statistically quantify whether MI and accuracy were statistically significantly higher (at the 95% confidence level) in different σ_w^2 values and delay times τ when compared to values at criticality.

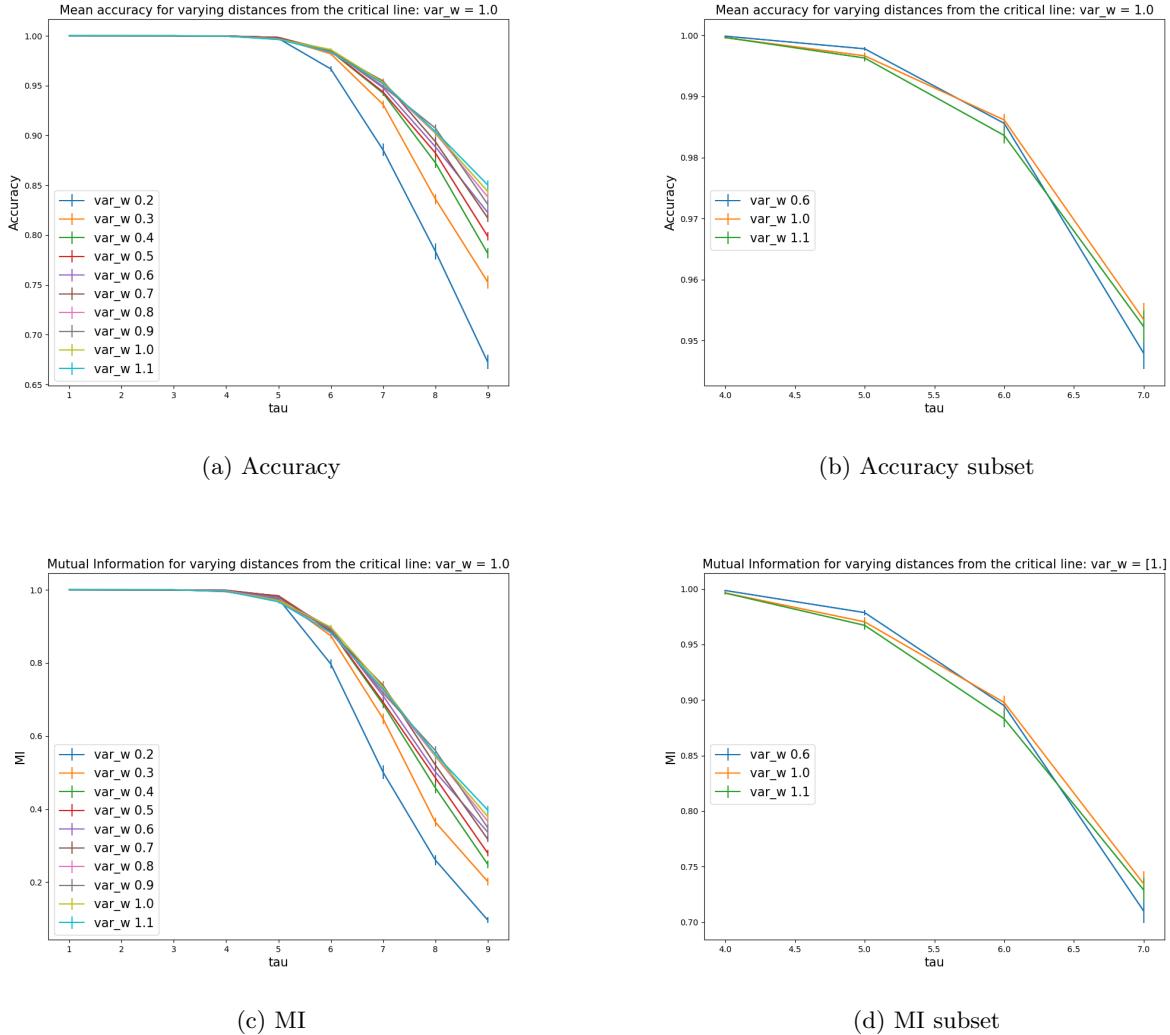


Figure 10: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 3$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 1$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

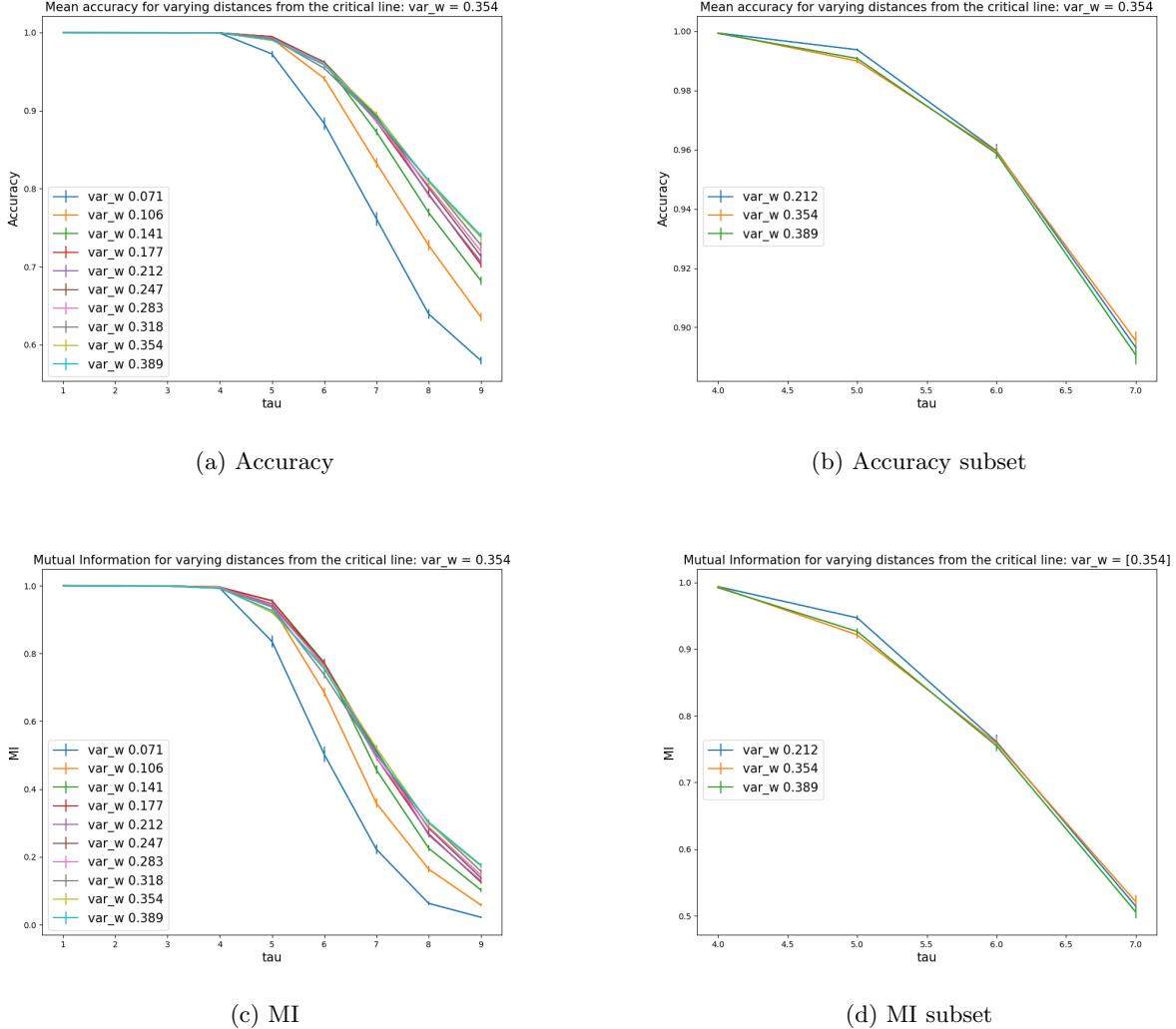


Figure 11: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 4$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 0.354$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

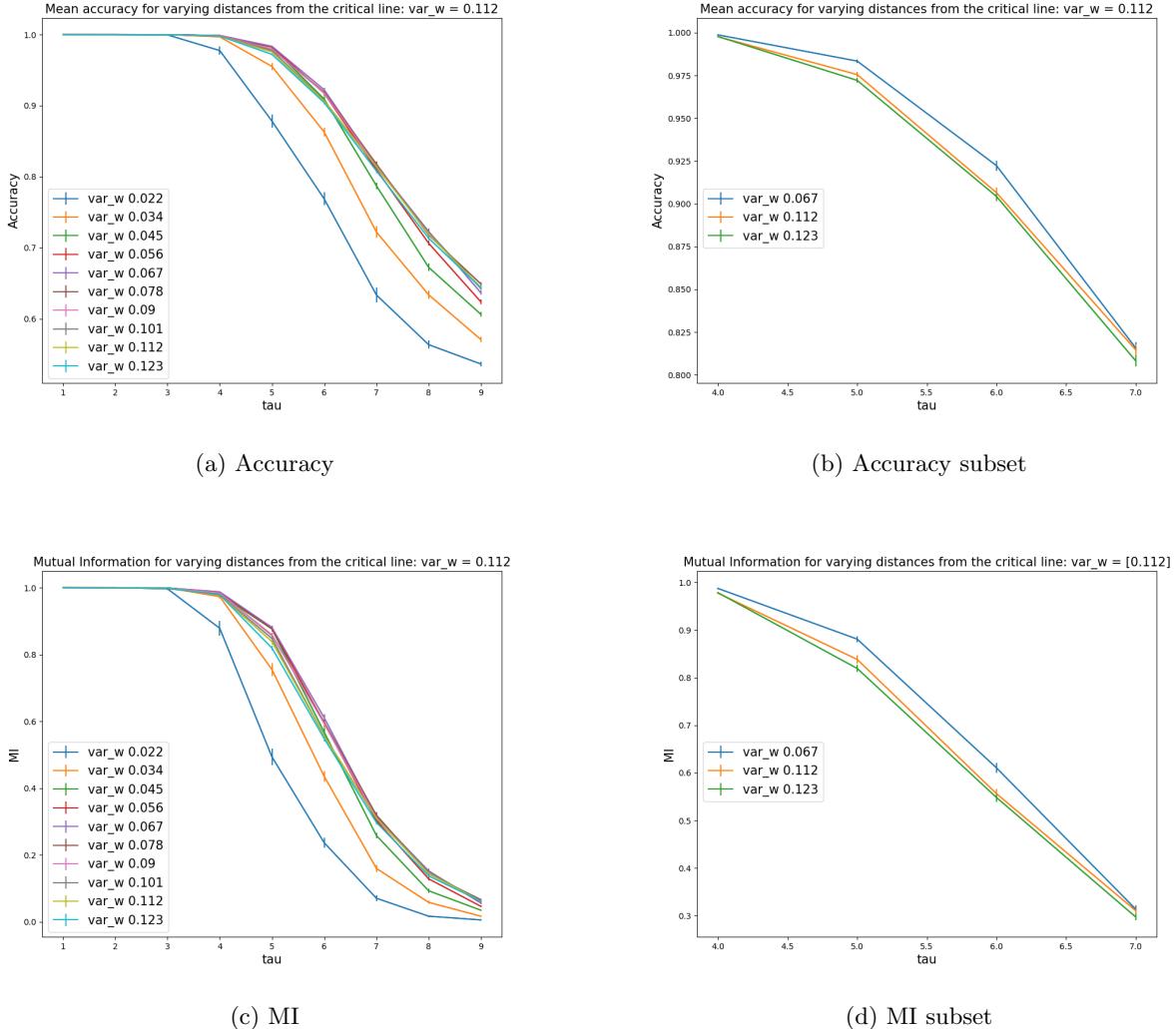


Figure 12: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 6$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 0.112$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

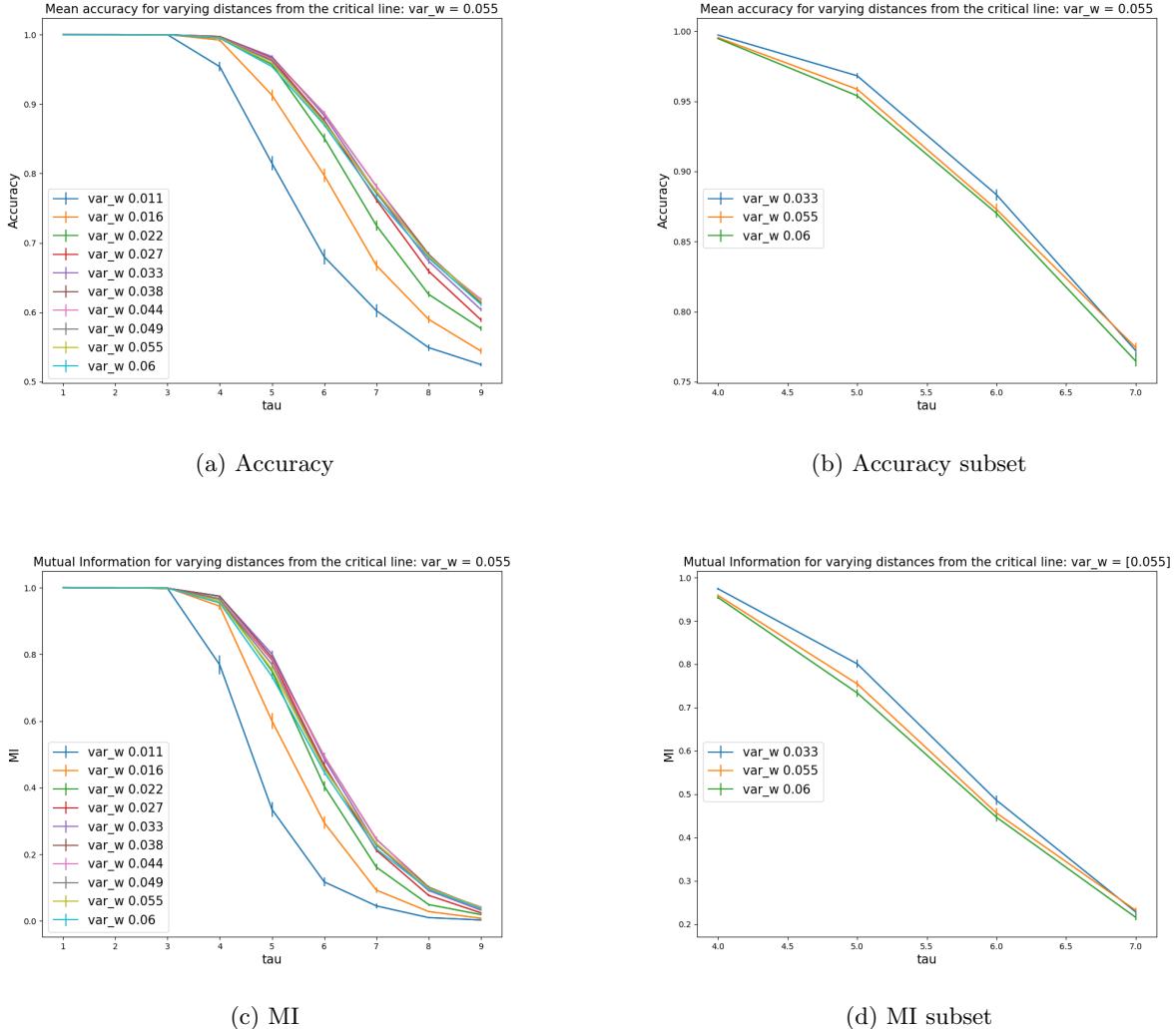


Figure 13: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 8$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 0.055$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

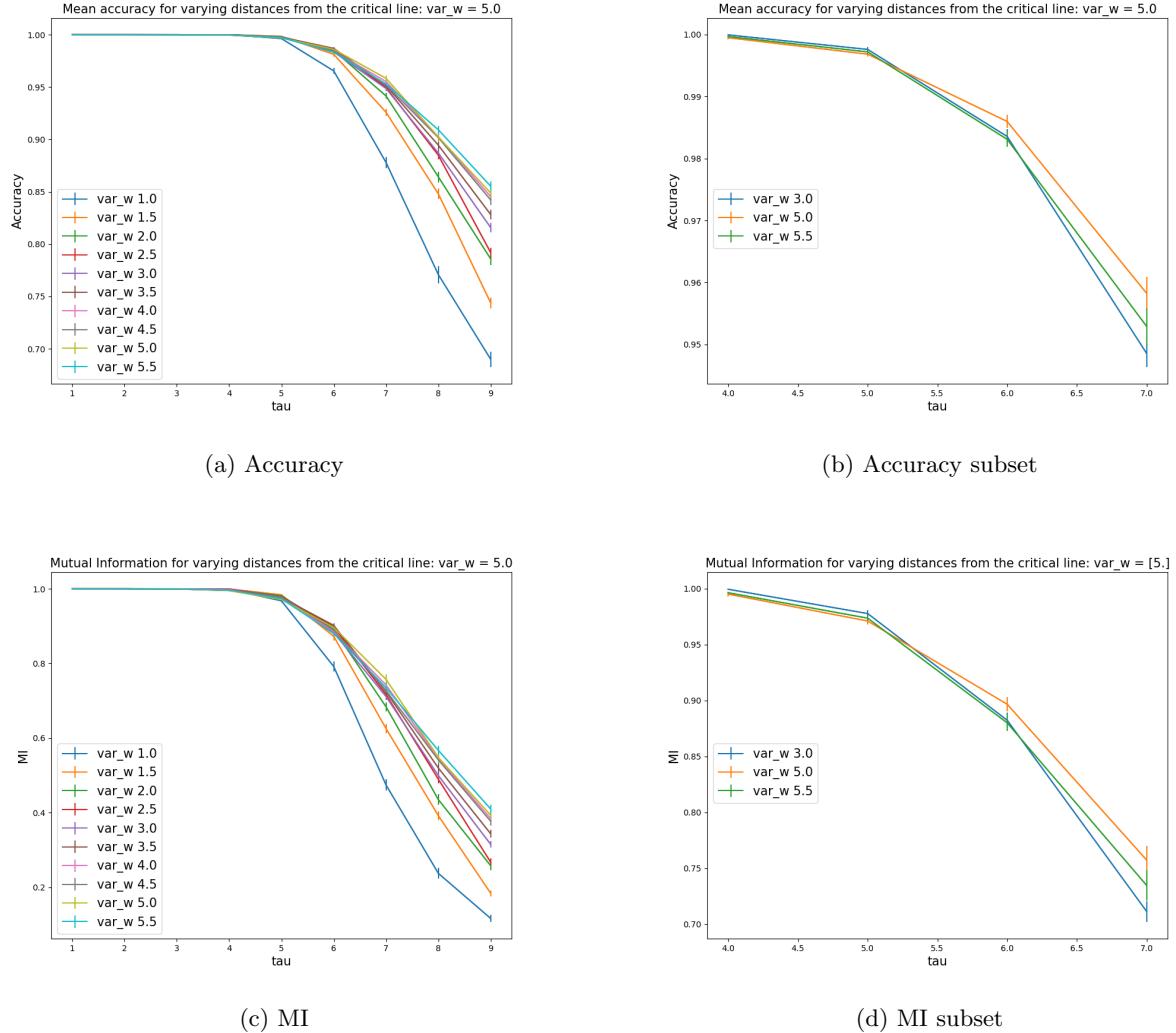


Figure 14: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 3$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 5$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

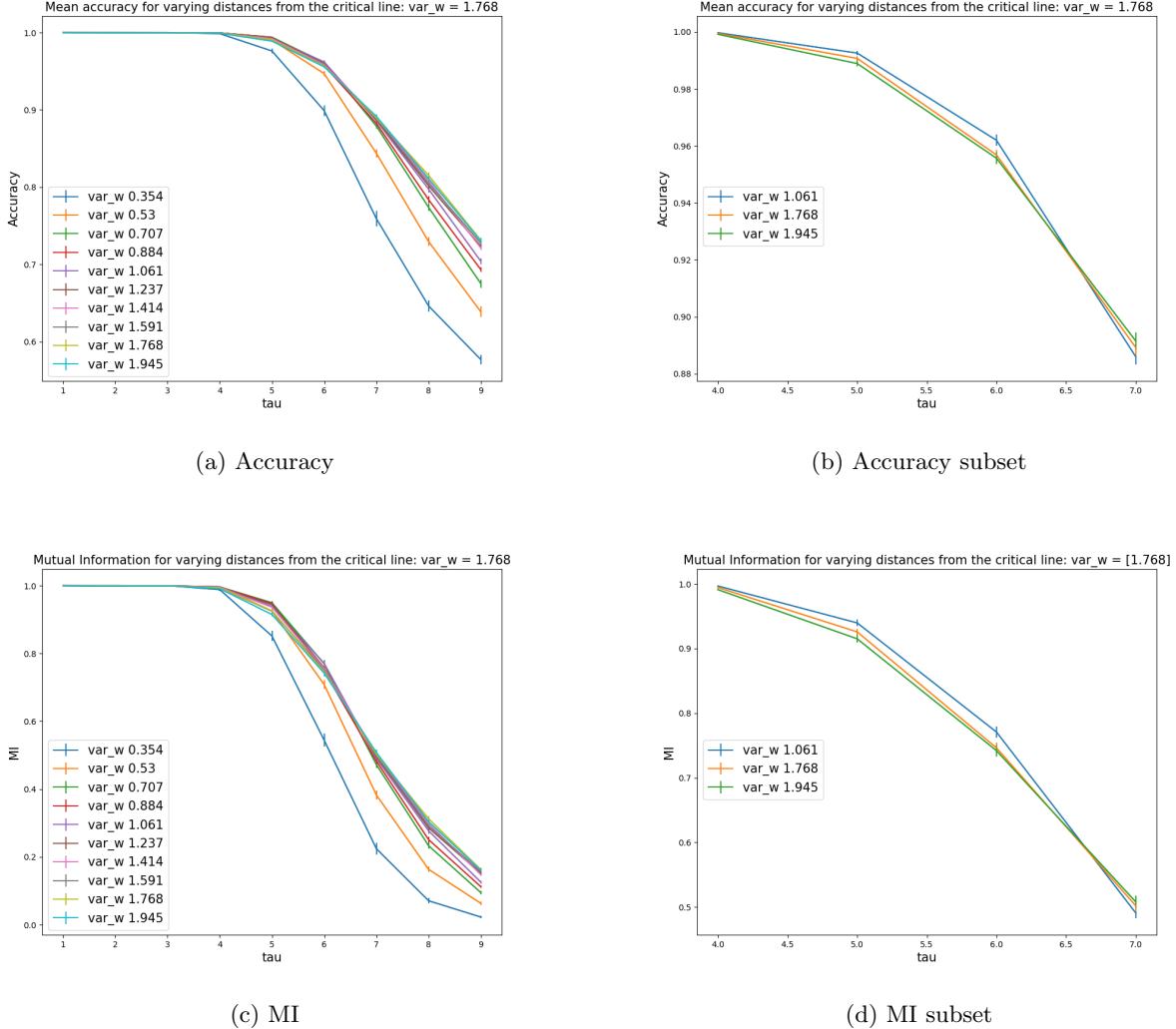


Figure 15: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 4$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 1.768$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

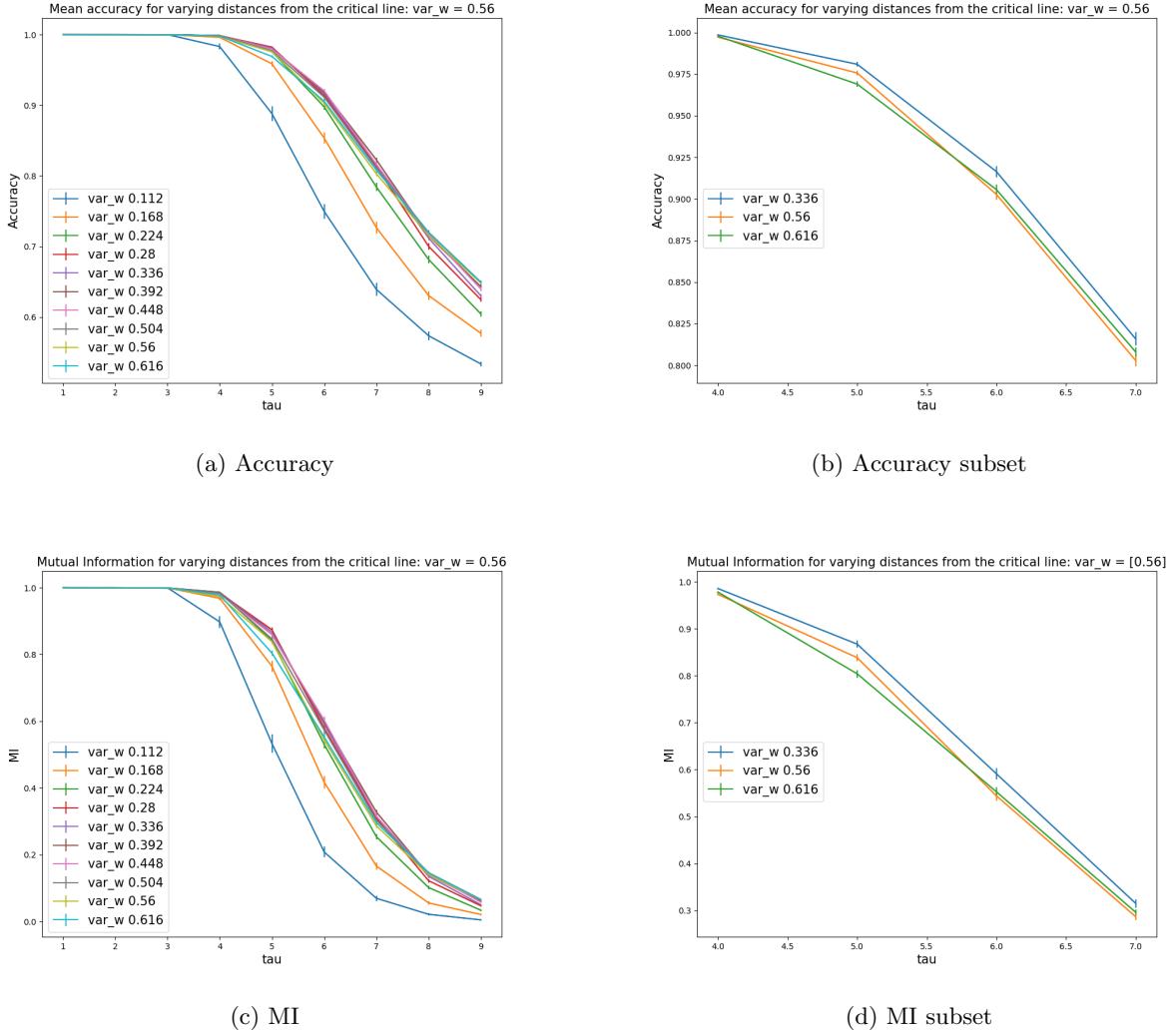


Figure 16: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 6$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 0.56$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

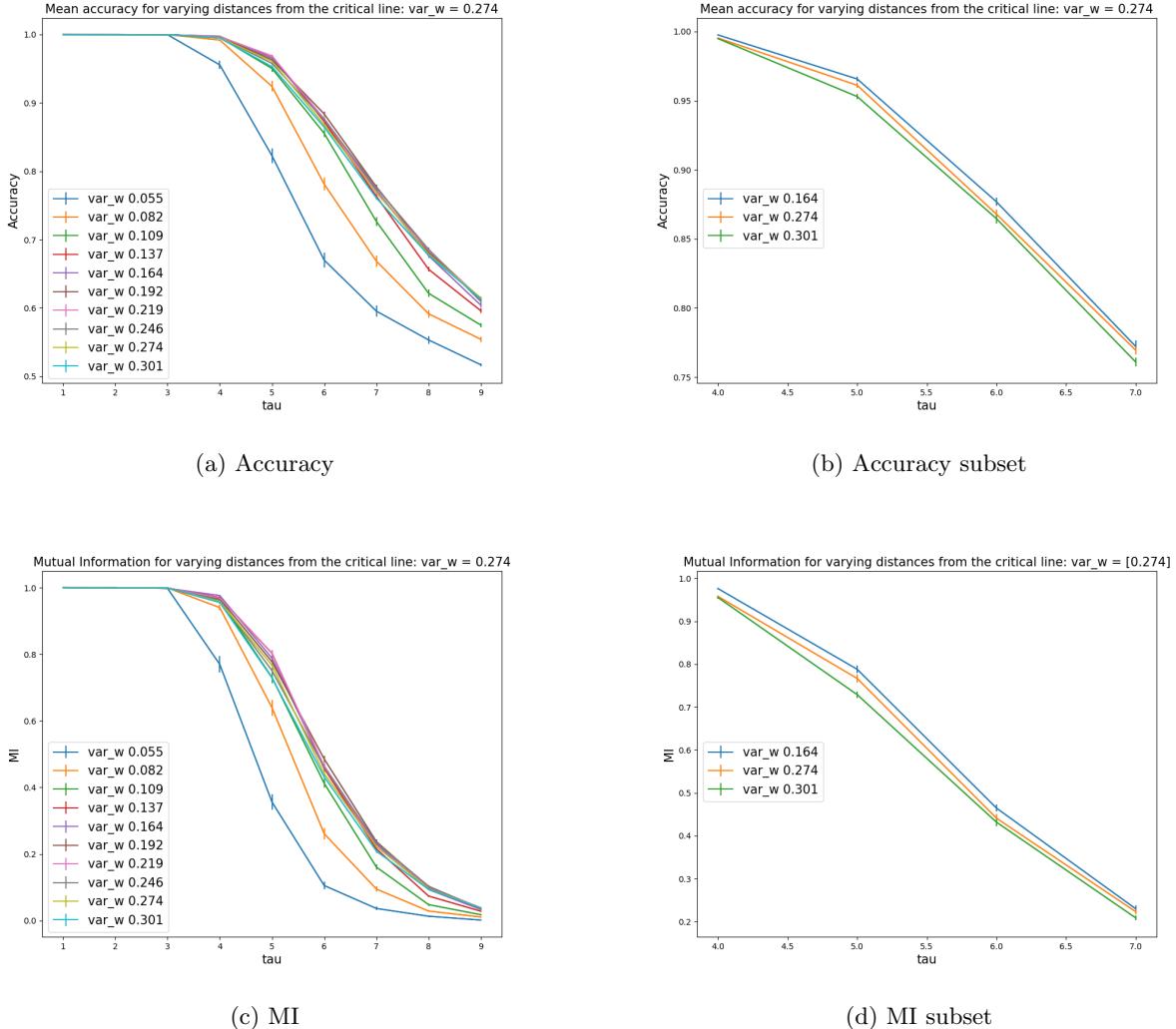


Figure 17: Random network Mutual Information and mean accuracy scores for the direct memory task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 8$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 0.274$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

Table 16: p-value for mean accuracy difference on the direct memory task with $\sigma_e^2 = 5$, $K = 8$ and 300 neurons. Marked in yellow are the respective values of σ_w^2 and τ that are significantly greater than the mean accuracy at criticality at the $\alpha = 0.05$ confidence level. The critical value for σ_w^2 is highlighted in red.

tau	1	2	3	4	5	6	7	8	9
var _w									
0.055	0.161	0.015	0.782	1.000	1.000	1.000	1.000	1.000	1.000
0.082	0.161	0.015	0.256	0.990	1.000	1.000	1.000	1.000	1.000
0.109	0.161	0.015	0.309	0.273	0.978	0.969	1.000	1.000	1.000
0.137	0.161	0.015	0.739	0.002	0.006	0.204	0.861	1.000	1.000
0.164	0.161	0.024	0.781	0.000	0.051	0.031	0.309	0.736	0.994
0.192	0.161	0.089	0.060	0.015	0.270	0.001	0.051	0.286	0.725
0.219	0.161	0.307	0.500	0.003	0.003	0.051	0.600	0.089	0.654
0.246	0.161	0.244	0.529	0.173	0.883	0.059	0.065	0.040	0.945
0.274	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
0.301	0.161	0.272	0.093	0.755	0.998	0.738	0.974	0.668	0.667

7 APPENDIX B: ADDITIONAL RESULTS FOR THE NON-LINEAR COMPUTATION TASK

In Figures 18 - 25 and tables 17 - 32 the same structure as in Appendix A follows for the results of the 3-bit parity task.

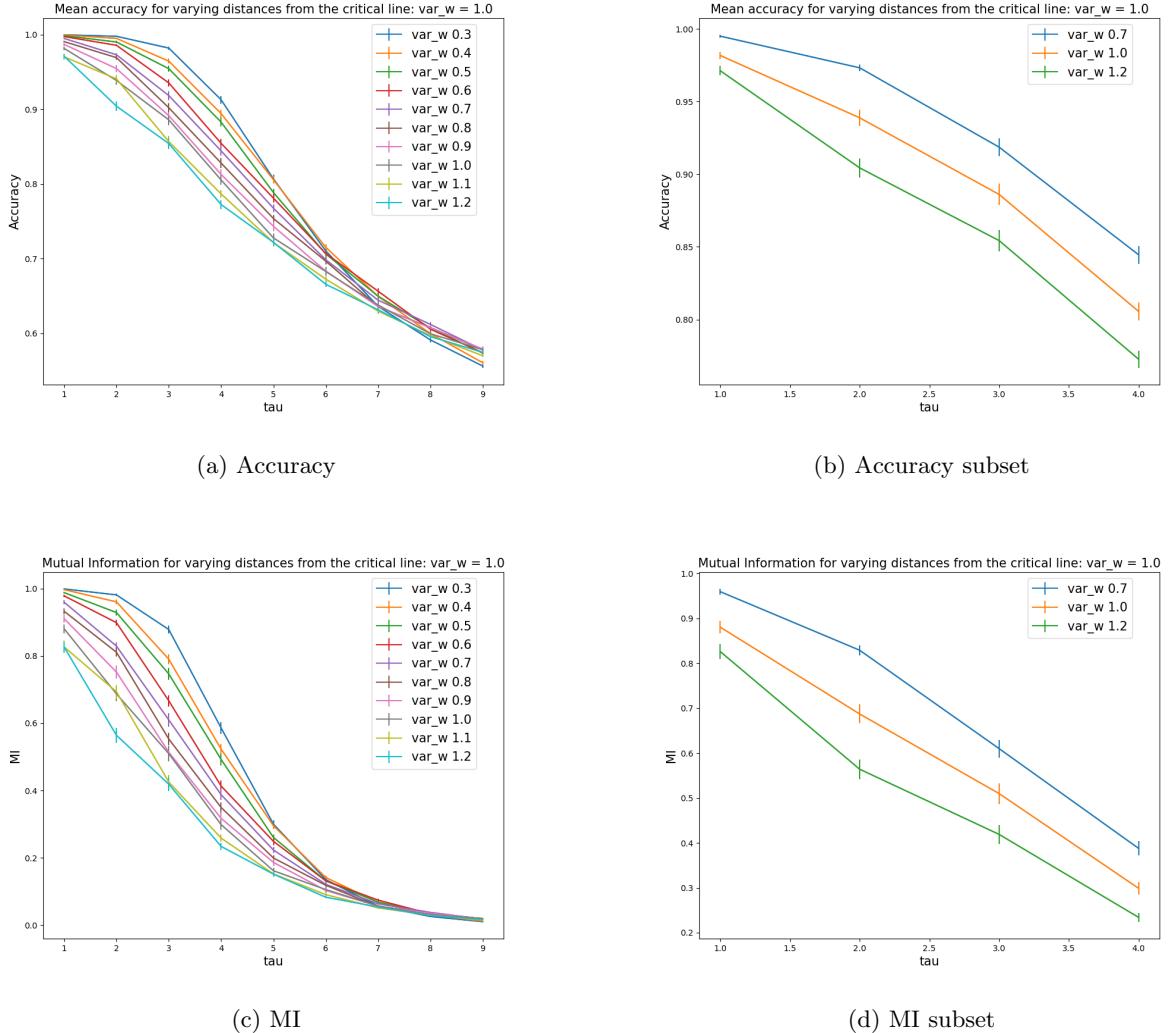


Figure 18: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 3$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 1$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

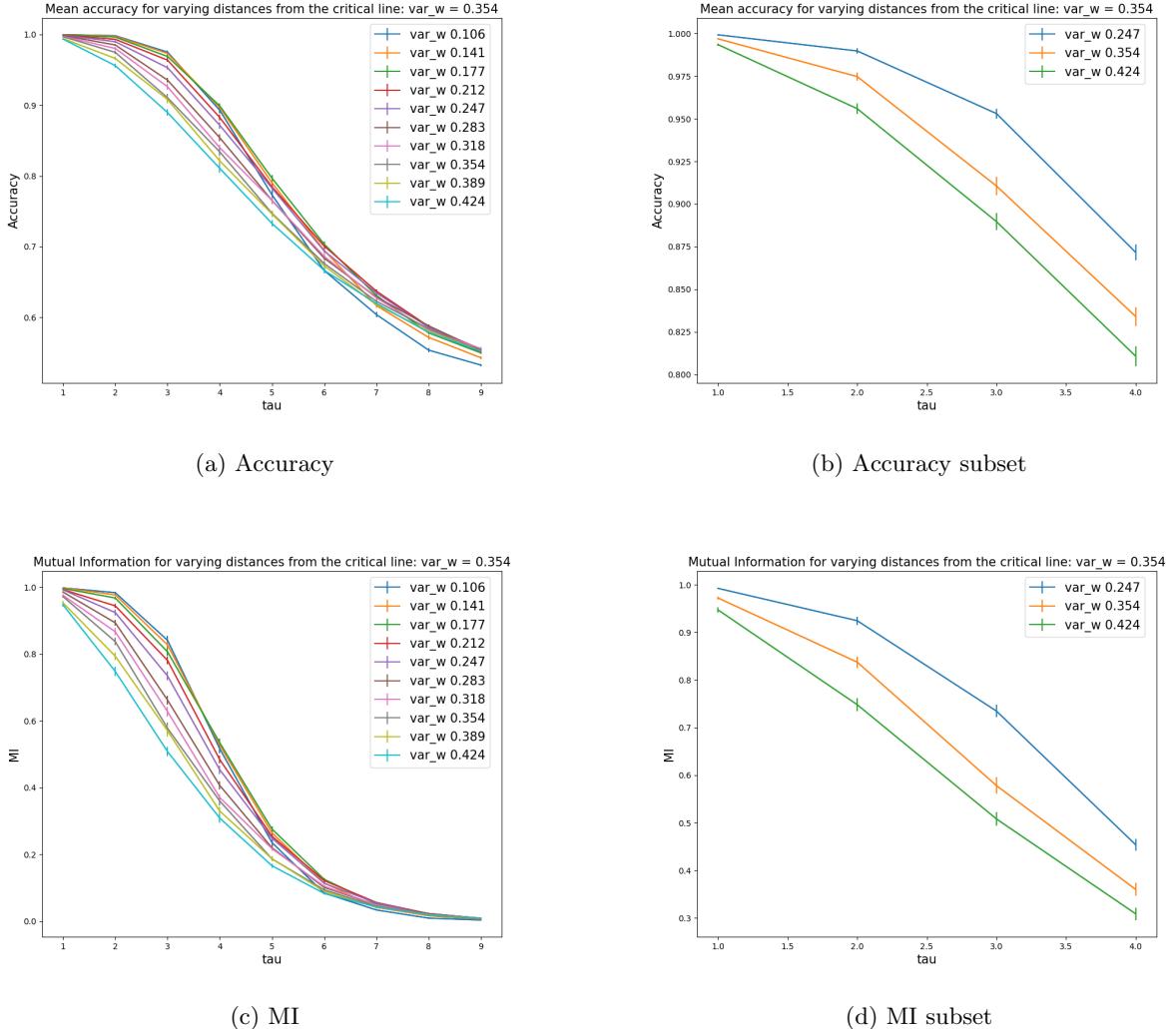


Figure 19: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 4$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 0.354$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

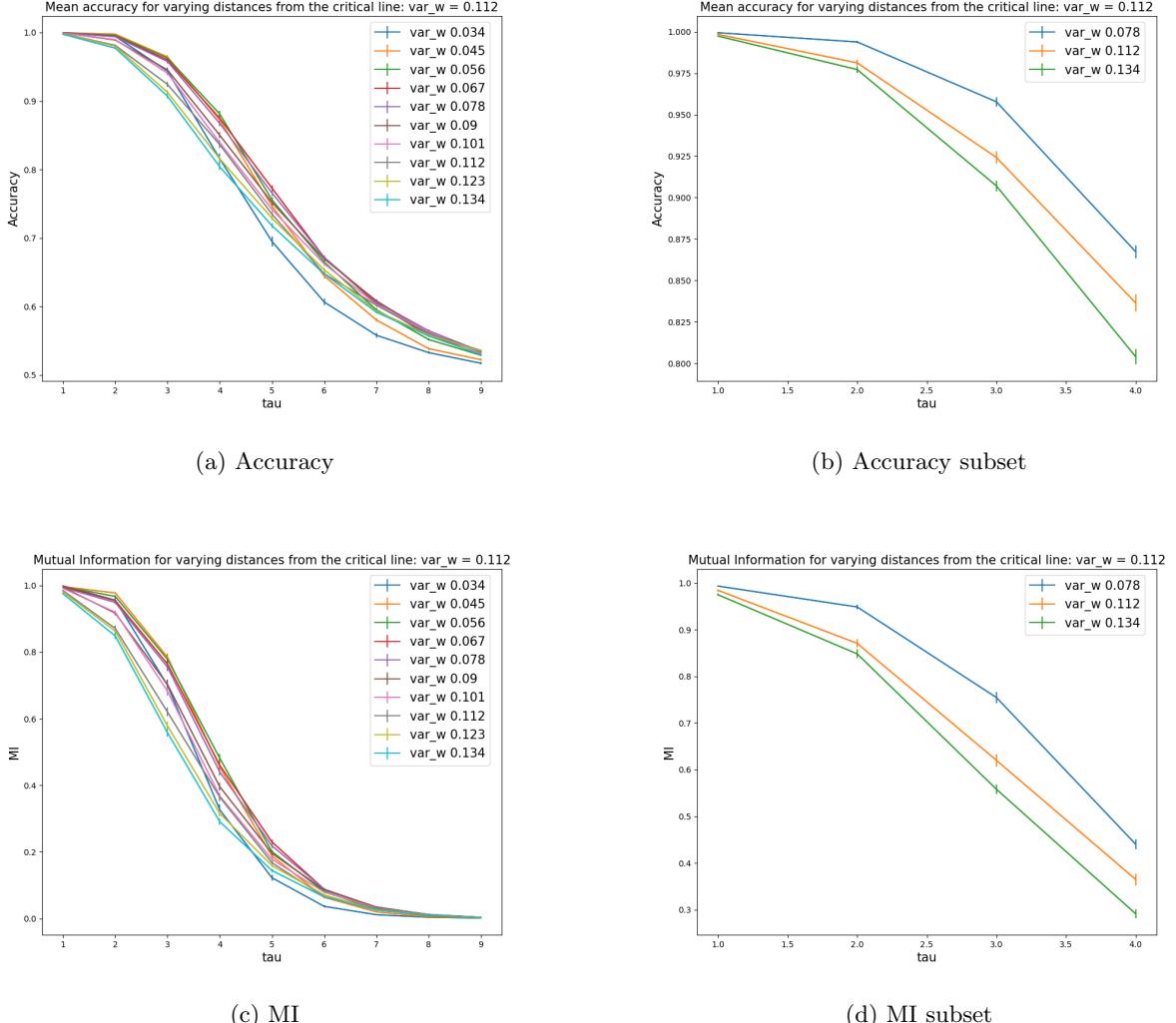


Figure 20: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 6$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 0.112$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

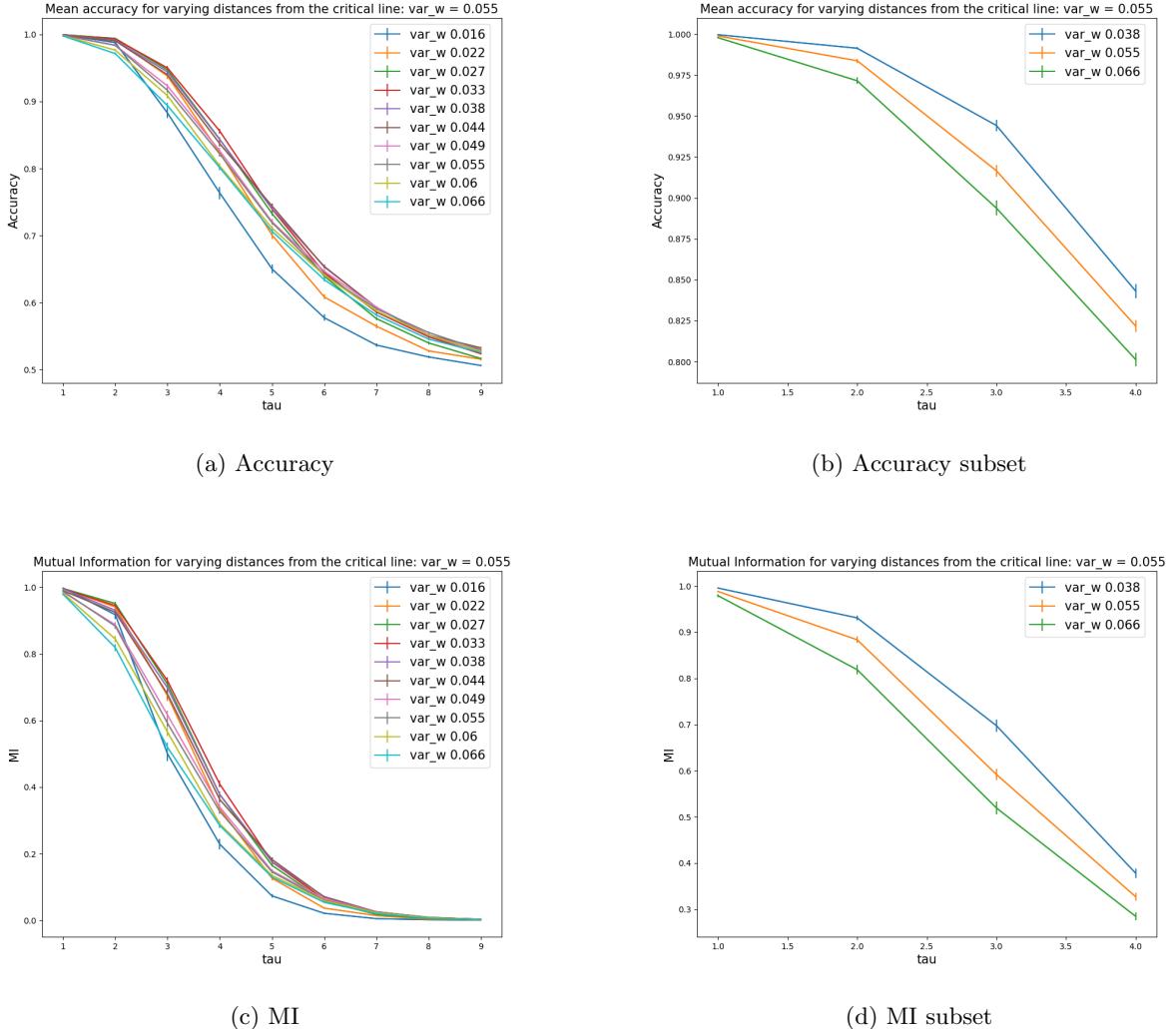


Figure 21: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 8$, $\sigma_e^2 = 1$ and the criticality value is given by $\sigma_w^2 = 0.055$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

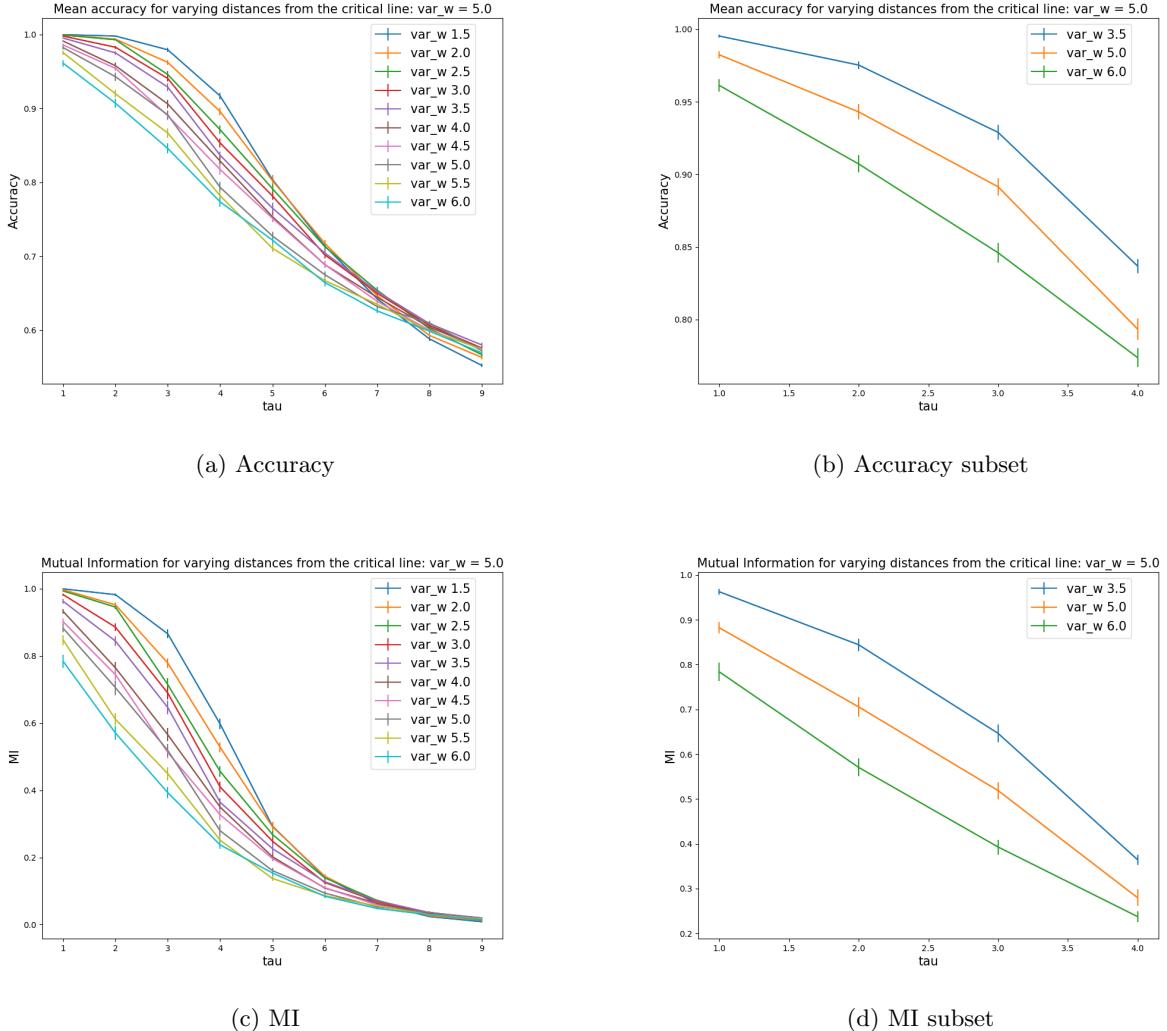


Figure 22: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 3$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 5$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

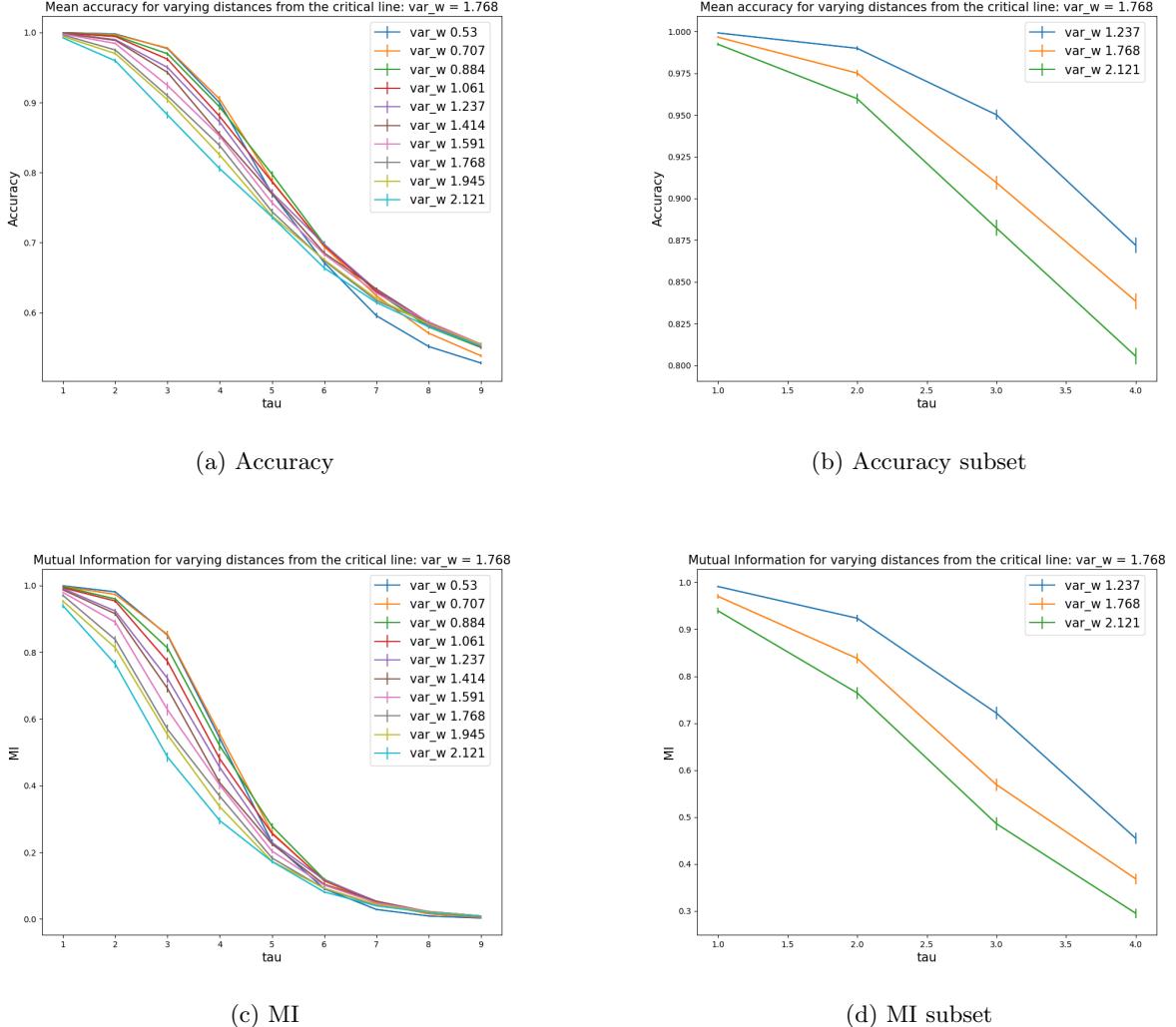


Figure 23: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 4$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 1.768$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

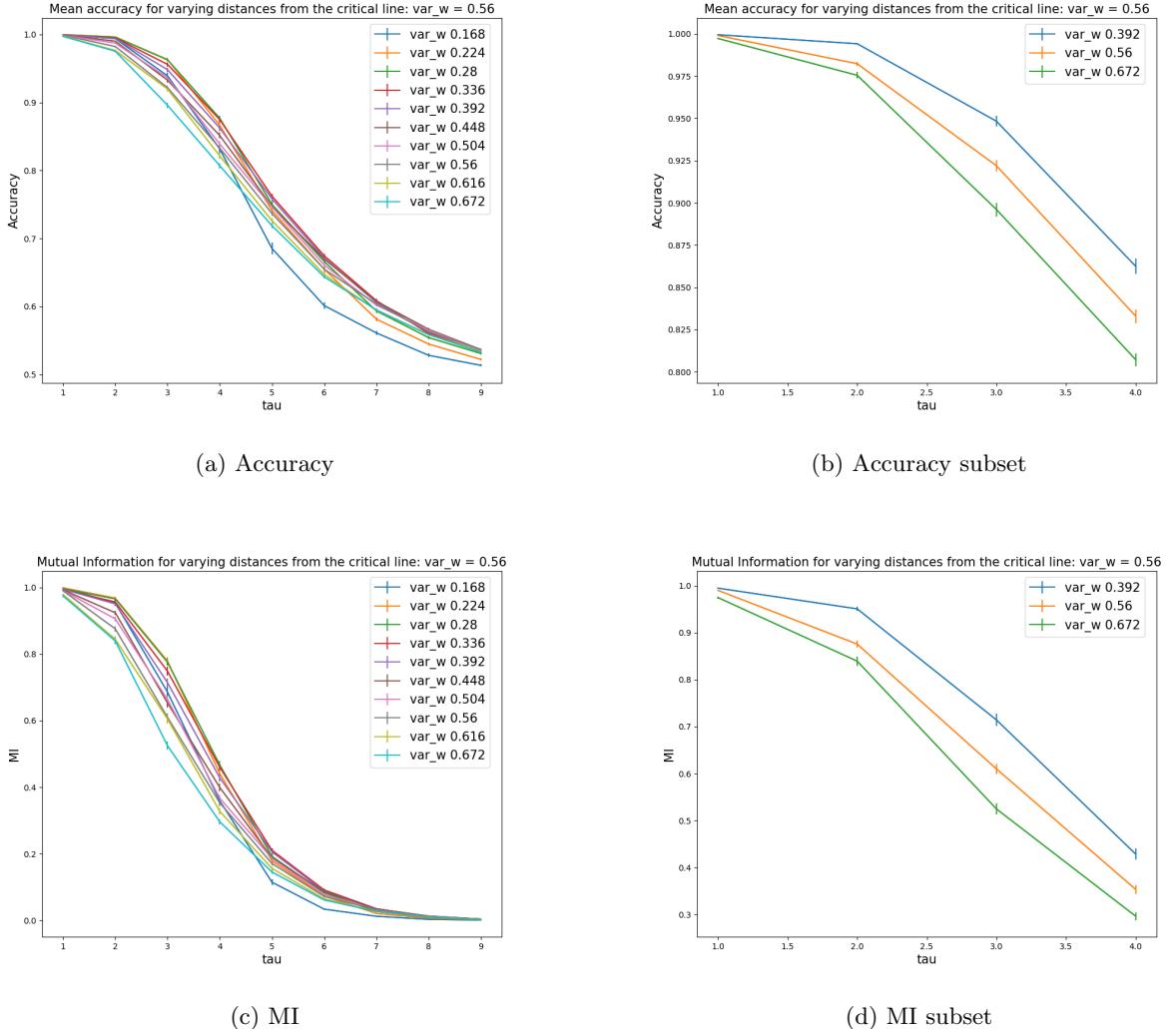


Figure 24: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 6$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 0.56$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

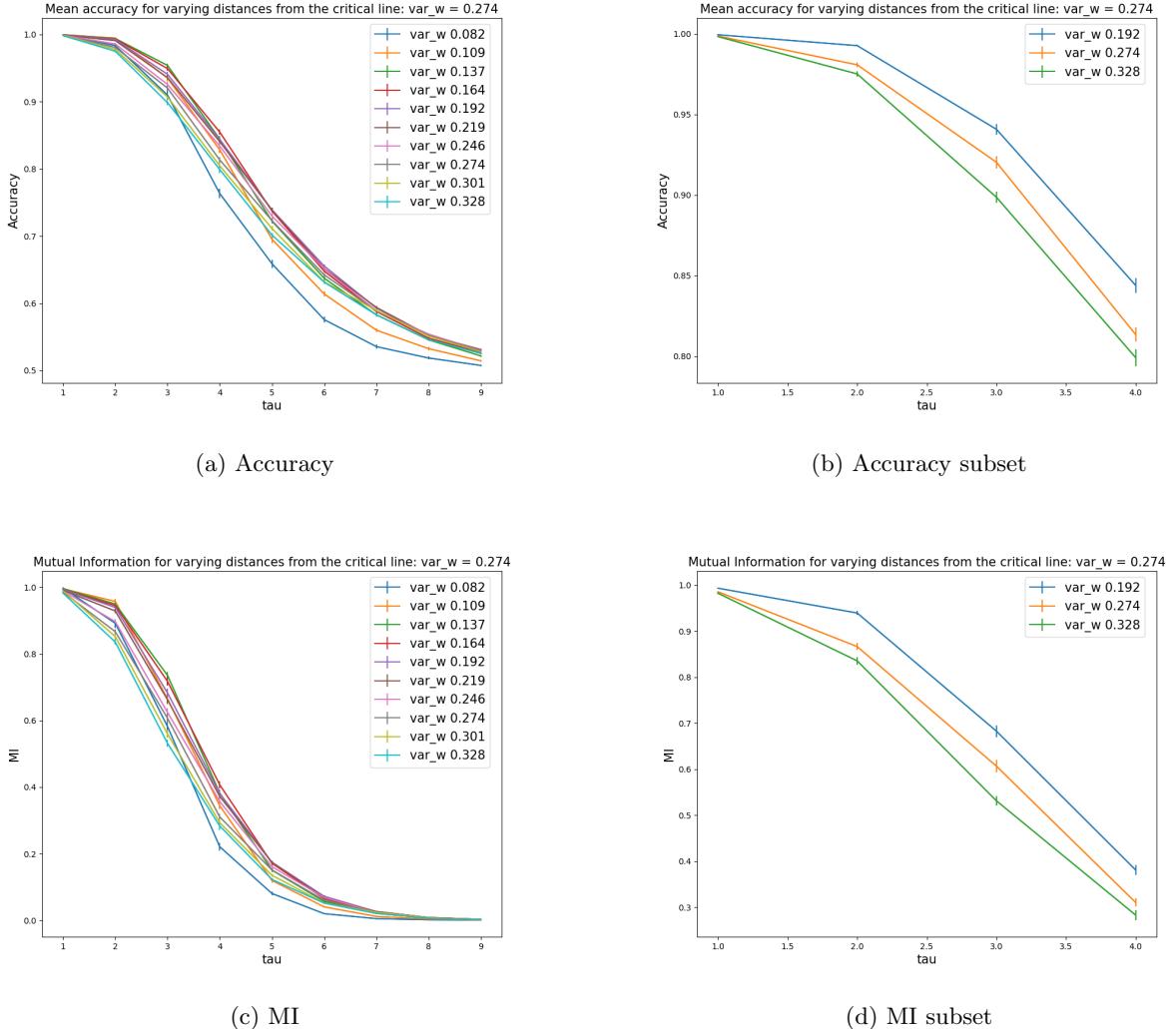


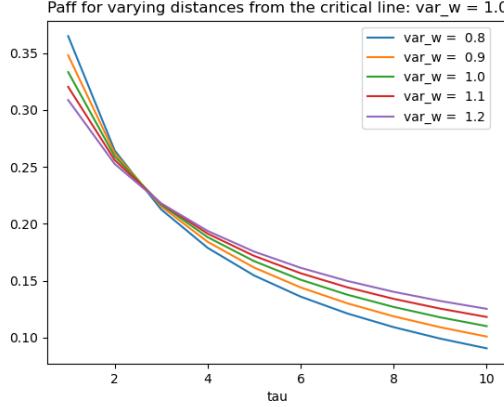
Figure 25: Random network Mutual Information and mean accuracy scores for the 3-bit parity task for varying τ , where τ denotes how many time steps ago the input one wishes to decode was given. The network parameters are given by $K = 8$, $\sigma_e^2 = 5$ and the criticality value is given by $\sigma_w^2 = 0.274$. (a) Mean accuracy score for all values of τ and σ_w^2 tested. (b) Mean accuracy of a subset of τ and σ_w^2 (to more clearly see the difference in behavior at different distances from criticality). (c) Mutual Information score for all values of τ and σ_w^2 tested. (d) Mutual Information score of a subset of τ and σ_w^2 .

Table 32: p-value for mean accuracy difference on the 3-bit parity task with $\sigma_e^2 = 5$, K = 8 and 300 neurons. Marked in yellow are the respective values of σ_w^2 and τ that are significantly greater than the mean accuracy at criticality at the $\alpha = 0.05$ confidence level. The critical value for σ_w^2 is highlighted in red.

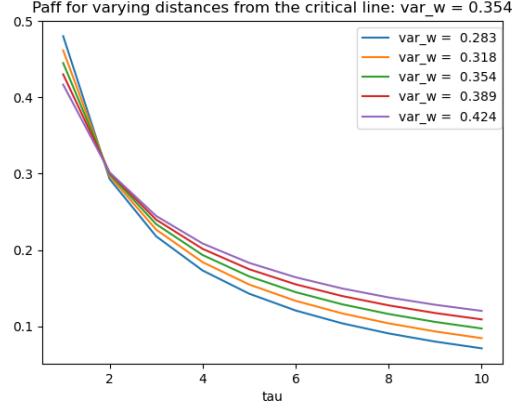
tau var _w	1	2	3	4	5	6	7	8	9
0.082	0.000	0.183	0.909	1.000	1.000	1.000	1.000	1.000	1.000
0.109	0.000	0.000	0.009	0.010	1.000	1.000	1.000	1.000	1.000
0.137	0.000	0.000	0.000	0.000	0.555	0.897	0.950	0.993	1.000
0.164	0.001	0.000	0.000	0.000	0.002	0.062	0.608	0.953	0.900
0.192	0.001	0.000	0.000	0.000	0.001	0.000	0.068	0.537	0.818
0.219	0.020	0.000	0.002	0.000	0.001	0.003	0.040	0.474	0.113
0.246	0.101	0.004	0.157	0.001	0.098	0.012	0.424	0.230	0.371
0.274	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500	0.500
0.301	0.047	0.891	0.991	0.908	0.990	0.997	0.386	0.702	0.735
0.328	0.863	0.996	1.000	0.977	1.000	0.999	0.978	0.996	0.975

8 APPENDIX C: ADDITIONAL RESULTS FOR P_{aff} AND N_{aff}

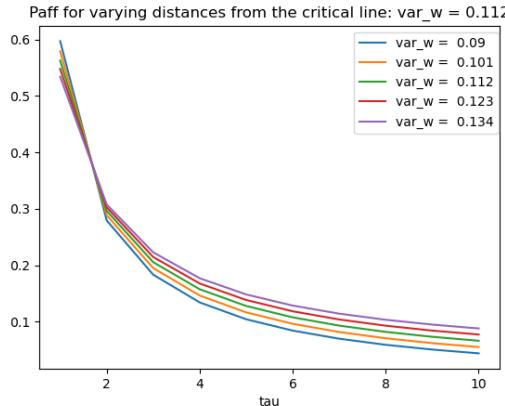
This section contains additional results for the numerical simulations of P_{aff} and N_{aff} described in Section 2.2.2. For completeness purposes, the same simulations were run but for a different value of σ_e^2 and its corresponding values of σ_w^2 and K , that should render sub-critical, critical and super-critical regimes. The results should match those in Section 2.2.2, as the behavior across the critical line should be invariant, which is indeed the case as seen in Figures 26 and 27 when compared to 2 and 3, respectively.



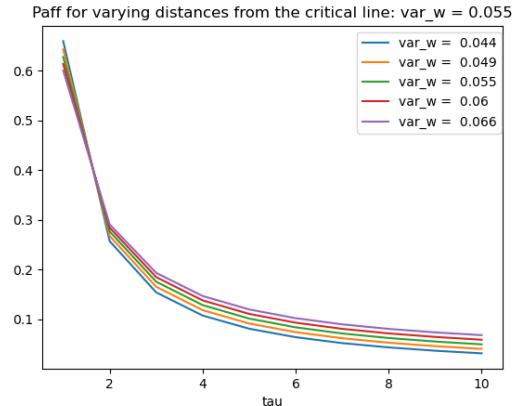
(a) $K = 3$



(b) $K = 4$



(c) $K = 6$



(d) $K = 8$

Figure 26: Probability of a neuron being affected by the input τ time steps ago. Different connectivity degrees are displayed, which individually determine what is the critical value of σ_w^2 . Sub-critical (blue and orange), critical (green) and super-critical (red and purple) are depicted. All cases are considered at $\sigma_e^2 = 1$.

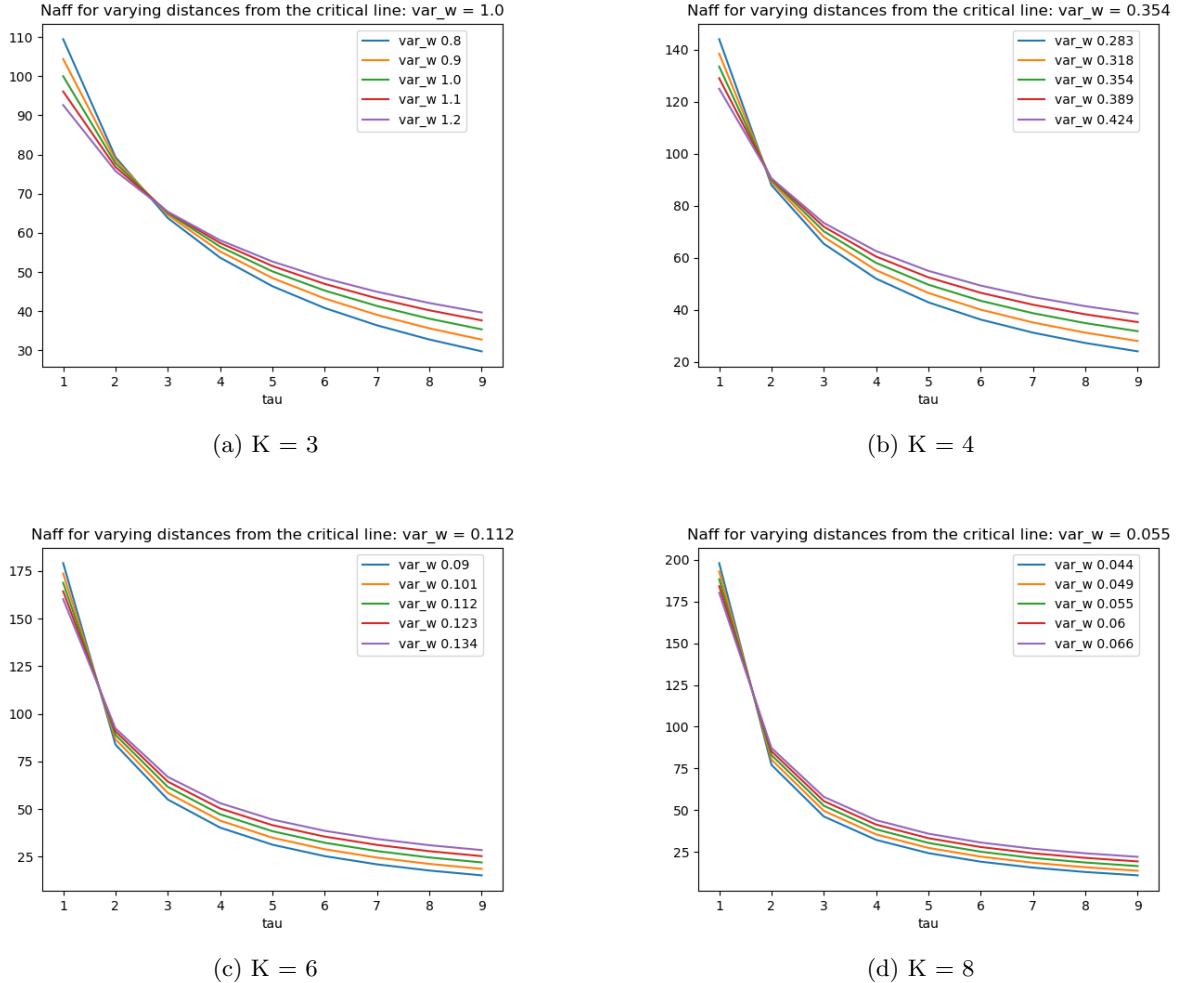


Figure 27: Expected number of neurons affected by the input τ time steps ago. Different connectivity degrees are displayed, which individually determine what is the critical value of σ_w^2 . Sub-critical (blue and orange), critical (green) and super-critical (red and purple) are depicted. All cases are considered at $\sigma_e^2 = 1$ and the number of neurons is 300.

9 ACKNOWLEDGEMENTS

I would like to thank my supervisors, Prof. Remi Monasson, Prof. Benjamin Grewe and Dr. Pau Vimellis Aceituno for their guidance in this MSc thesis. I would particularly like to thank Prof. Monasson for introducing me into the world of statistical physics and Dr. Vimellis for guiding me through criticality research. Both supervisors have constantly challenged me intellectually and made this thesis a fascinating learning experience. Lastly, I would like to thank my supervisor and mentor Prof. Benjamin Grewe for an excellent mentorship throughout my Masters' degree.

REFERENCES

- [1] Joschka Boedecker, Oliver Obst, Joseph T. Lizier, N. Michael Mayer, and Minoru Asada. Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, 131(3):205–213, 9 2012.
- [2] Thierry Mora and William Bialek. Are Biological Systems Poised at Criticality? *Journal of Statistical Physics*, 144(2):268–302, 7 2011.
- [3] Oren Shriki and Dovi Yellin. Optimal Information Representation and Criticality in an Adaptive Sensory Recurrent Neuronal Network. *PLOS Computational Biology*, 12(2):e1004698, 2 2016.
- [4] Guillermo B. Morales and Miguel A. Muñoz. Optimal Input Representation in Neural Systems at the Edge of Chaos. *Biology*, 10(8):702, 7 2021.
- [5] Nils Bertschinger and Thomas Natschläger. Real-Time Computation at the Edge of Chaos in Recurrent Neural Networks. *Neural Computation*, 16(7):1413–1436, 7 2004.
- [6] Robert DiPietro and Gregory D. Hager. Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*, pages 503–519. Elsevier, 2020.
- [7] Robert Legenstein and Wolfgang Maass. Edge of chaos and prediction of computational performance for neural circuit models. *Neural Networks*, 20(3):323–334, 4 2007.
- [8] Igor Farkaš, Radomír Bosák, and Peter Gergel. Computational analysis of memory capacity in echo state networks. *Neural Networks*, 83:109–120, 11 2016.
- [9] T. Toyoizumi and L. F. Abbott. Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime. *Physical Review E*, 84(5):051908, 11 2011.
- [10] Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep Information Propagation. 11 2016.
- [11] Kian Katanforoosh and Daniel Kunin. Initializing neural networks, 2019.
- [12] John M. Beggs and Dietmar Plenz. Neuronal Avalanches in Neocortical Circuits. *The Journal of Neuroscience*, 23(35):11167–11177, 12 2003.
- [13] John M. Beggs and Nicholas Timme. Being Critical of Criticality in the Brain. *Frontiers in Physiology*, 3, 2012.
- [14] Jonathan Touboul and Alain Destexhe. Power-law statistics and universal scaling in the absence of criticality. *Physical Review E*, 95(1):012413, 1 2017.
- [15] Michael P. H. Stumpf and Mason A. Porter. Critical Truths About Power Laws. *Science*, 335(6069):665–666, 2 2012.
- [16] J Wilting and V Priesemann. Between Perfectly Critical and Fully Irregular: A Reverberating Model Captures and Predicts Cortical Spike Propagation. *Cerebral Cortex*, 29(6):2759–2770, 6 2019.
- [17] Miguel A. Muñoz. *Colloquium: Criticality and dynamical scaling in living systems*. *Reviews of Modern Physics*, 90(3):031001, 7 2018.
- [18] Woodrow L. Shew and Dietmar Plenz. The Functional Benefits of Criticality in the Cortex. *The Neuroscientist*, 19(1):88–100, 2 2013.
- [19] Marzieh Zare and Paolo Grigolini. Criticality and avalanches in neural networks. *Chaos, Solitons & Fractals*, 55:80–94, 10 2013.
- [20] Izzet B. Yildiz, Herbert Jaeger, and Stefan J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 11 2012.

- [21] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 4 1982.
- [22] B. Derrida, E. Gardner, and A. Zippelius. An exactly solvable asymmetric neural network model. *EPL*, 4(2), 1987.
- [23] Maximilian Uhlig, Anna Levina, Theo Geisel, and J. Michael Herrmann. Critical dynamics in associative memory networks. *Frontiers in Computational Neuroscience*, 7, 2013.
- [24] Bruno Del Papa, Viola Priesemann, and Jochen Triesch. Criticality meets learning: Criticality signatures in a self-organizing recurrent neural network. *PLOS ONE*, 12(5):e0178683, 5 2017.
- [25] Benjamin Cramer, David Stöckel, Markus Kreft, Michael Wibral, Johannes Schemmel, Karlheinz Meier, and Viola Priesemann. Control of criticality and computation in spiking neuromorphic networks with plasticity. *Nature Communications*, 11(1):2853, 12 2020.
- [26] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, 11 2002.
- [27] Daniel J. Gauthier, Erik Boltt, Aaron Griffith, and Wendson A. S. Barbosa. Next generation reservoir computing. *Nature Communications*, 12(1):5564, 12 2021.
- [28] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake Vanderplas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. 9 2013.
- [29] B Derrida. Dynamical phase transition in nonsymmetric spin glasses. *Journal of Physics A: Mathematical and General*, 20(11):L721–L725, 8 1987.
- [30] L. de Almeida, M. Idiart, and J. E. Lisman. Memory retrieval time and memory capacity of the CA3 network: Role of gamma frequency oscillations. *Learning & Memory*, 14(11):795–806, 11 2007.
- [31] Claudio Gallicchio. Sparsity in Reservoir Computing Neural Networks. 6 2020.
- [32] Nicolas Cazin, Martin Llofriu Alonso, Pablo Scleidorovich Chiodi, Tatiana Pelc, Bruce Harland, Alfredo Weitzenfeld, Jean-Marc Fellous, and Peter Ford Dominey. Reservoir computing model of prefrontal cortex creates novel combinations of previous navigation sequences from hippocampal place-cell replay with spatial reward propagation. *PLOS Computational Biology*, 15(7):e1006624, 7 2019.
- [33] Stefano Recanatesi, Ulises Pereira-Obilinovic, Masayoshi Murakami, Zachary Mainen, and Luca Mazzucato. Metastable attractors explain the variable timing of stable behavioral action sequences. *Neuron*, 110(1):139–153, 1 2022.