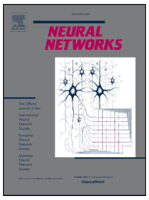
Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Complexity of brain-like signals in self-organised nanoscale networks

Jamie K. Steel [a], Ford Wagner [a], Edoardo Galli [a], Susant K. Acharya [a], Joshua B. Mallinson [a], Philip J. Bones [b], Matthew D. Arnold [c], Simon A. Brown [a,*]

[a] The MacDiarmid Institute for Advanced Materials and Nanotechnology, School of Physical and Chemical Sciences, University of Canterbury, Christchurch, 8140, New Zealand
[b] Electrical and Computer Engineering, University of Canterbury, Christchurch, 8140, New Zealand
[c] School of Mathematical and Physical Sciences, University of Technology Sydney, Sydney, Australia

ARTICLE INFO

Keywords:
Percolating networks of nanoparticles
Complexity
Self-assembly
Neural networks
Neuromorphic computing

ABSTRACT

The biological brain is comprised of a complex, interconnected, self-assembled network of neurons and synapses. This network enables efficient and accurate information processing, unsurpassed by any other known computational system. Percolating networks of nanoparticles (PNNs) are complex, interconnected, self-assembled systems that exhibit many emergent brain-like characteristics. Notably, neuron-like spiking patterns from PNNs have been shown to be critical, similar to signals from the cortex. PNNs are therefore an appealing candidate for neuromorphic computational systems. Here, the inherent complexity of the patterns of switching events generated by PNNs is explored using several different measures. We begin by defining qualitative measures of spatial, temporal, and spatio-temporal complexity, and then investigate a quantitative measure of complexity that was developed for analysis of patterns of spikes from neurons in the cortex. We discuss adaptations of the method that are required for data from the electronic devices of interest and the impact of various pre-processing procedures on the analysis. Through these measures, it is shown that the neuron-like spiking patterns from PNNs are indeed complex and are clearly distinct from random and ordered data.

1. Introduction

Traditional approaches to computing, rooted in the von Neumann model, have driven remarkable technological advancements over the last 50 years. These approaches are now, however, nearing fundamental physical and energy efficiency limits (Kaiser, 2024), motivating interest in alternative approaches. The brain demonstrates extraordinary computational capabilities, performing complex tasks, such as pattern recognition and decision-making, much more efficiently than artificial computing systems (Kandel et al., 2021). Brain-like (or neuromorphic) computing systems (Mead, 1990; Tanaka et al., 2019) therefore seek to bridge the gap between the biological and digital realms. A key challenge for neuromorphic computing is to move beyond rigid, regular architectures towards the intricate, complex, and critical architectures found within the brain (Christensen et al., 2022; Srinivasa et al., 2015).

Percolating networks of nanoparticles (PNNs, Fig. 1) and other self-organised nanoscale networks (comprising e.g. nanowires (Avizienis et al., 2012; Daniels et al., 2022; Hochstetter et al., 2021; Milano et al., 2022b) and nanotubes (Akai-Kasaya et al., 2022; Dale et al., 2016; Tanaka et al., 2022)) have been shown to be promising systems for neuromorphic computing (Bose et al., 2017; Mallinson et al., 2019). PNNs exhibit scale-free network architectures and intrinsic dynamical complexity, with long-range temporal correlations (LRTC), scale-invariant dynamics, and critical avalanching behaviour (Mallinson et al., 2019; Shirai et al., 2020). PNNs are made through a simple and scalable deposition process (Reichel et al., 2006), making them a cost-effective option for implementing novel computing paradigms such as reservoir computing (RC) (Mallinson et al., 2023, 2024) and probabilistic computing (Studholme & Brown, 2024; Studholme et al., 2025).

PNNs operate in two distinct regimes (a low-stimulus tunnelling regime and a high-stimulus switching regime) but in both regimes it is the gaps between connected groups of nanoparticles (Fig. 1a) that generate the useful dynamics. In the tunnelling regime, the input signals modulate the sizes of the tunnel gaps, leading to memristive behaviour that can be exploited for applications such as reservoir computing (Mallinson et al., 2023, 2024). In the switching regime, electrical stimulation causes atomic-scale filaments to form in the tunnel gaps (and later rupture) with corresponding step-wise changes in the measured output currents from the network (Fig. 1b). The switching events (or “spikes”) can be used as the basis for several computing schemes (Studholme & Brown, 2024; Studholme et al., 2023, 2025). On average, each switching event triggers one other switching event,

[*] Corresponding author.
E-mail address: simon.brown@canterbury.ac.nz (S.A. Brown).

https://doi.org/10.1016/j.neunet.2025.108031
Received 5 May 2025; Received in revised form 23 July 2025; Accepted 23 August 2025
Available online 27 August 2025

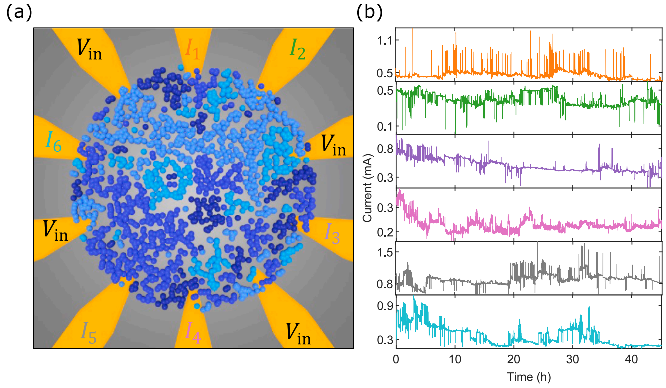


Fig. 1. (a) Schematic of the PNN with four input and six output electrodes (DC input voltages V_{in} and output currents I_i). (b) The measured signals from all six output electrodes of the PNN due to a 10 V DC stimulus applied simultaneously to all four input contacts. The line colours correspond to the output electrodes labelled with the same colours in (a). Complicated switching behaviour is clearly observed across all output electrodes for the duration of the measurement.

which leads to cascades of spatio-temporally correlated events, referred to as avalanches. These avalanches exhibit power-law statistics that are similar to those observed in the cortex (Friedman et al., 2012; Muñoz, 2018) and which are consistent with criticality (Mallinson et al., 2019). Criticality is associated with optimal information processing (Srinivasa et al., 2015).

While criticality and correlations in PNNs have been studied extensively (Mallinson et al., 2019; Shirai et al., 2020), the inherent *complexity* of the signals from PNNs and other self-assembled nanoscale systems has yet to be investigated. Complexity is important in a number of fields (Adami, 2002; Bar-Yam, 1999) and is of particular interest in neuroscience (Bullmore & Sporns, 2009; Green et al., 2018) where, for example, it has been conjectured that complexity correlates with higher states of consciousness (Tononi et al., 1994).

Here we investigate the inherent complexity of the signals measured from PNNs, and compare it to the complexity of signals observed in the cortex (Beggs & Plenz, 2003; Marshall et al., 2016; Timme et al., 2016). To achieve this, we analyse the complexity of data from a long-duration measurement from a multi-electrode PNN (Fig. 1). We begin with a qualitative analysis that demonstrates PNN data is clearly distinct from that produced by a random system. We then quantify these results using the neural complexity measure that was developed specifically for neural systems (Marshall et al., 2016; Tononi et al., 1994). The quantitative analysis follows closely a previous study of data from neural cultures (Timme et al., 2016). Note that throughout this report we present the key results and analysis in the main text and refer the reader to several appendices for technical details of the analysis.

2. Materials and methods

Device fabrication. PNNs were fabricated using straightforward nanoparticle deposition processes (Bose et al., 2017). Multi-contact metal electrodes were fabricated through photolithographic processes, thermal evaporation, and lift-off techniques on a Si wafer with a 200 nm thick Si_3N_4 passivation layer. The metal electrodes comprised a 5 nm NiCr adhesion layer underneath 45 nm of Au (Acharya et al., 2021). These electrodes are equally spaced around the circumference of a circle with a diameter of 600μm. As detailed by Bose et al. (2017), Mallinson et al. (2019), a beam of 7 nm Sn particles is deposited on the devices in a UHV environment. The particles coalesce to a degree, resulting in clusters of particles with diameters ∼20 nm. Electrical feedthroughs enable the device conductance to be continuously monitored during the deposition process. The deposition was terminated upon reaching the onset of conduction, which corresponds to the percolation threshold. The deposition occurs in a controlled environment and leads to devices that can be successfully operated for durations as long as one year (Mallinson et al., 2024).

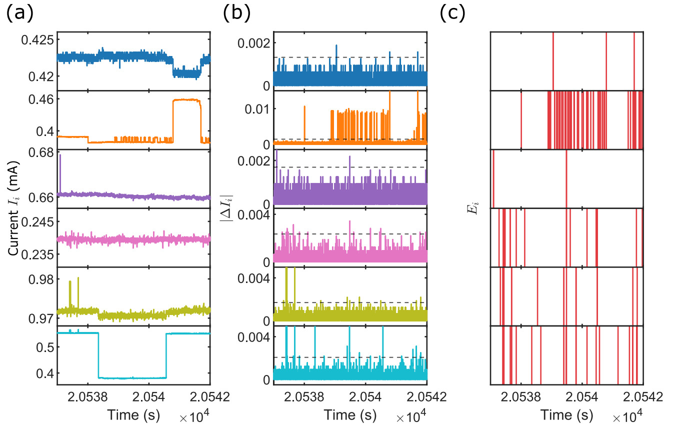


Fig. 2. Example showing the conversion of measured currents into event trains. (a) 0.5 s segment of the measured currents I_i from the six output electrodes shown in Fig. 1. The step-wise changes are caused by the formation and destruction of atomic filaments within the network. (b) Corresponding absolute value of the measured changes in current $|\Delta I_i|$. The horizontal dashed line indicates the threshold level; any changes in current greater than this level indicate that an event has occurred. (c) Corresponding event trains E_i for each output electrode. The red bars mark the events observed in panels a and b.

Electrical stimulus and measurement. Electrical measurements were conducted using a National Instruments multi-channel data acquisition system (NI PXIe-1082 and PXIe-6378 ADC/DAC) in conjunction with a custom-developed sample mounting system. A constant DC electrical stimulus was applied to four electrodes spaced around the periphery of the PNN. Currents were measured from the remaining six output electrodes as shown schematically in Fig. 1a (in practice, for each electrode a voltage is measured across a 1 kΩ resistor to ground). The sampling rate was fixed at 1 kHz as this provides high enough temporal resolution for robust event identification while allowing manageable data volumes for analysis of the long duration measurement.

3. Results

We focus here on one particularly long dataset (45 h) that is representative of the various key features observed in many shorter measurements made across multiple PNNs. For this dataset $V_{\text{in}} = 10$ V and the output currents I_i, where $i \in [1, 2, \ldots, 6]$, are shown in Fig. 1b. Extensive switching activity is observed at all output electrodes throughout the measurement, totalling some 16 million events across all electrodes. While the rich, scale-free signals (Mallinson et al., 2019) are only partially visible on the timescale of Fig. 1b, it is immediately clear that these signals *look* complex: many events are detected simultaneously at several different electrodes, but few events are detected at all electrodes, and the patterns of events are not repeated (see also Fig. 2). The challenge is to find meaningful measures of this complexity.

Fig. 2 shows the method by which the currents I_i are converted into event trains E_i for each electrode. Changes in the measured currents (Fig. 2a) are thresholded to remove noise (Acharya et al., 2021) (Fig. 2b) resulting in event trains E_i, i.e. arrays of 1's and 0's (Fig. 2c) corresponding to time-steps (1 ms each) for which there are events and no events, respectively. The six E_i arrays are vertically stacked to form a single matrix, referred to as the event raster **R**. The event trains E_i and event raster **R** are used for most of the complexity analyses, but additionally we define bipolar event trains F_i in which positive and negative ΔI_i are represented by 1's and −1's, respectively, and 0's still indicate where no event took place.

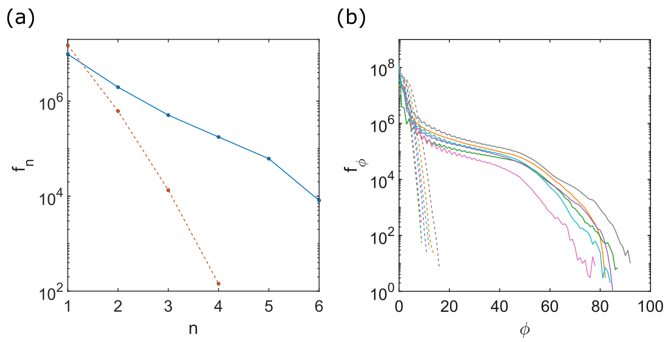


Fig. 3. (a) Spatial complexity measure, i.e. the number of events f_n that are simultaneously observed at n contacts, for the experimental event raster (solid blue line) and temporally randomised data (dashed orange line). The slopes of the two curves are clearly distinct, indicating that the experimental data is more spatially complex than the temporally randomised data. (b) Temporal complexity measure, i.e. the number of times f_ϕ that ϕ events are observed in a window-length $w = 100$, for the experimental event trains (solid lines) and temporally randomised data (dashed lines). Each colour corresponds to a specific electrode (the same colour code as in Fig. 2a and b). The experimental and temporally randomised data are clearly distinct, indicating the experimental data is more temporally complex than the randomised data.

3.1. Qualitative measures of complexity

We begin with a qualitative analysis of the complexity of the switching data contained in the event raster. Three measures of complexity are discussed: spatial, temporal, and spatio-temporal. To the best of our knowledge each of these qualitative measures is novel.

3.1.1. Spatial complexity, f_n

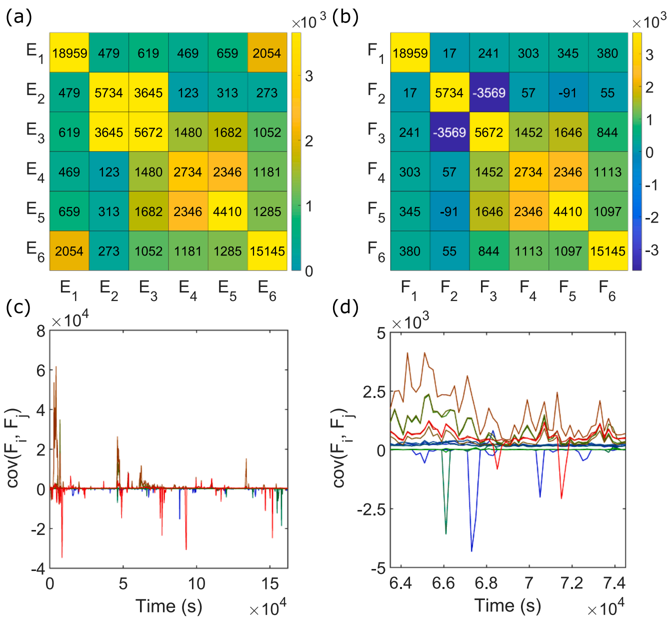
We begin by defining a spatial complexity measure f_n, which simply describes the number of time-steps for which an event is simultaneously detected at n electrodes. For example, $f_3 = 10$ means that events were detected at exactly three electrodes in only ten distinct time-steps (out of the full measurement). Note that the particular combination of electrodes is not important (one count is recorded in f_3 if an event is observed at electrodes 1, 4 and 6, *or* if an event is observed at electrodes 2, 3 and 6, etc.).

Fig. 3a shows the variation of f_n with n for the long dataset of interest. f_n decreases approximately exponentially with increasing n for the experimental event raster (solid blue line). Importantly, f_n is non-zero for all n (including $n = 6$), indicating that there are many time-steps for which events are simultaneously detected at multiple electrodes. When each event train is temporally randomised to destroy spatial correlations (see Appendix C), f_n also decreases exponentially, but with a much steeper gradient (dashed orange line). The steeper gradient means that $f_n = 0$ for $n > 4$, i.e. after temporal randomisation, events are never simultaneously recorded on five or six electrodes, confirming that the spatial correlations in the original event raster are destroyed.

3.1.2. Temporal complexity, f_ϕ

We now characterise the temporal complexity of an event train (i.e. of the signals measured at a *single* electrode). We define f_ϕ to be the number of segments of an event train, of window-length w time-steps, in which ϕ events are observed. For example, if an event train has total length $l = 100$ and a window-length $w = 10$ is used, there are $l - w + 1 = 91$ segments of the event train consisting of ten consecutive time-steps. A value of $f_4 = 12$ then means that 12 (out of a possible 91) distinct segments of the event train registered exactly four (out of a possible ten) events.

Fig. 3b shows the variation of f_ϕ with ϕ for $w = 100$ for the long dataset of interest. The solid lines correspond to the six experimental event trains and the dashed lines to the corresponding temporally randomised event trains. Each colour corresponds to a particular electrode. f_ϕ for the experimental event trains is non-zero for $\phi \lesssim 80$. In contrast, f_ϕ for the temporally randomised event trains decays exponentially and is zero for $\phi > 17$ in all cases. Large values of f_ϕ indicate bursty activity in the experimental event trains (many events occur in quick succession) (Mallinson et al., 2019; Shirai et al., 2020). In contrast, for temporally randomised event trains, the randomisation procedure destroys all temporal correlations and large values of f_ϕ are not observed.



	E_1	E_2	E_3	E_4	E_5	E_6
E_1	18959	479	619	469	659	2054
E_2	479	5734	3645	123	313	273
E_3	619	3645	5672	1480	1682	1052
E_4	469	123	1480	2734	2346	1181
E_5	659	313	1682	2346	4410	1285
E_6	2054	273	1052	1181	1285	15145

	F_1	F_2	F_3	F_4	F_5	F_6
F_1	18959	17	241	303	345	380
F_2	17	5734	-3569	57	-91	55
F_3	241	-3569	5672	1452	1646	844
F_4	303	57	1452	2734	2346	1113
F_5	345	-91	1646	2346	4410	1097
F_6	380	55	844	1113	1097	15145

Fig. 4. (a) Matrix of $cov(E_i, E_j)$ values for one 200 s segment of the unipolar event raster. The diagonal entries are the number of events detected at an electrode. The off-diagonal entries are the number of shared events between two event trains. The colour scale is chosen to highlight the off-diagonal elements. (b) Matrix of $cov(F_i, F_j)$ values for the same segment of the bipolar event raster. The off-diagonal entries are differences between the number of shared events with the same and opposite polarities. (c) Time-series of $cov(F_i, F_j)$ for all 200 s segments of the bipolar event raster. All 15 unique pairs of F_i are shown with different colours. (d) Magnified section of the $cov(F_i, F_j)$ curves from (c) illustrating dramatic changes in the correlations between event trains over time.

3.1.3. Spatio-temporal complexity

To characterise the spatio-temporal complexity of the data, we consider the covariance between all pairs of event trains. For any two time series, represented by arrays X_i and X_j, the covariance between them is defined as

$$cov(X_i, X_j) = \sum_{t=1}^{n} (X_i(t) - \mu_i)(X_j(t) - \mu_j) \tag{1}$$

where μ_i and μ_j are the mean values of X_i and X_j. For the covariance between two unipolar event trains E_i and E_j, the mean values are set to zero so that the calculated covariance value corresponds exactly to the event count. Fig. 4a shows the matrix of $cov(E_i, E_j)$ values for one example 200 s segment of data. The diagonal elements correspond to the number of events in E_i, while the 15 off-diagonal elements correspond to the number of events that occurred simultaneously in *both* event trains E_i and E_j (referred to as *shared events*). Fig. 4b shows the matrix $cov(F_i, F_j)$ for the same segment of data. Note that because bipolar event trains F_i include information about the sign of the event, the off-diagonal elements correspond to the *difference* in the number of events with positive and negative ΔI. A detailed comparison of Fig. 4a and b allows the total number of events with positive and negative ΔI to be deduced.

Fig. 4c shows a time-series of $cov(F_i, F_j)$, calculated for 200 s segments of the bipolar event trains. All 15 unique pairs are shown, each

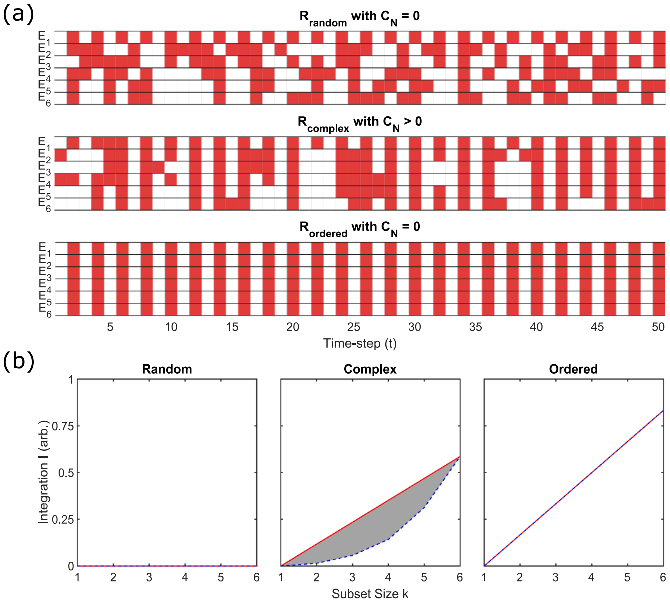


Fig. 5. (a) Three example event rasters comprised of six event trains (generated via the simple chain model provided in the NCC Toolbox (Marshall et al., 2016)) and corresponding neural complexity values demonstrating: random (top), complex (middle), and ordered (bottom) data. The red squares correspond to an event at a particular time-step; the white squares for no event. (b) Integration curves corresponding to the example event rasters in (a) for: random (left), complex (middle), and ordered (right) data. In essence, the neural complexity metric C_{N} calculates the area (shaded grey) between the linear approximation to the integration (solid red) and the average integration (dashed blue). Only complex data has a non-zero area. Note that these examples are illustrative only, and that much longer event trains are required to achieve meaningful values of C_{N}.

with a different colour. Remarkably, the patterns of correlations captured in the $cov(F_i, F_j)$ matrices often vary dramatically from one segment to the next. Moreover, the polarity of the correlation itself can change. For example, the brown curve shows over 6 × 10^4 shared events occurred in one segment, while the red curve shows over 3 × 10^4 shared events with opposite sign, demonstrating highly active and correlated switching data for different pairs of electrodes. The large number of shared events for some segments of data obscures weaker correlations and so we also show Fig. 4d which is a magnified section of Fig. 4c. The correlations clearly fluctuate between positive and negative polarity, consistent with complex spatio-temporal dynamics and the bursty nature of the data.

3.2. Quantitative measure of complexity

We now explore the application of the neural complexity measure C_{N} (Tononi et al., 1994) to data from PNNs. Neural complexity is a measure designed to describe the interplay of functional segregation and global integration, two fundamental characteristics of brain organisation that regulate information flow (Deco et al., 2015). Thus, the objective here is to apply the neural complexity measure to the event raster corresponding to the long PNN dataset of interest, and to compare the results to those from biology.

The neural complexity measure is designed such that both independent (random) and dependent (ordered) data yield low neural complexity values, whereas complex data result in comparatively high values. Fig. 5a illustrates three example event rasters (top: random; middle: complex; bottom: ordered), along with their respective (ideal) neural complexity values. Only the complex data yields a non-zero value of C_{N}.

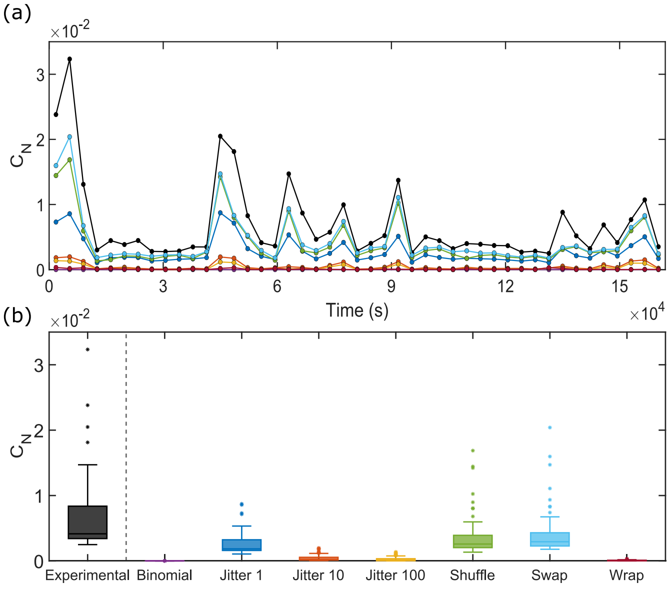


Fig. 6. Comparison of C_{N} between experimental and randomised versions (see Table C.2) of the event raster for consecutive 3600 s segments. (a) Time-series representation where each point is marked at the midpoint of the respective segment of data. The black curve corresponds to C_{N} for the experimental event raster (no randomisation), while the other colours indicate specific randomisation procedures (see panel (b) and Appendix C). For each randomisation procedure, the uncertainties are shown as the standard deviation of 10 independent realisations (negligible in most cases). The experimental C_{N} is consistently the highest. (b) A box and whisker graph showing the same data as (a). The stars indicate outliers calculated as greater than 1.5 × IQR (interquartile range).

The calculation of C_{N}, as utilised by Timme et al. (2016), and as implemented in the NCC Toolbox (Marshall et al., 2016), is summarised in Appendix A.

Pre-processing and sub-sampling. It is important to note that the implementation of neural complexity used in the neuroscience literature involves two pre-processing steps: binning the data on the time-scale of the average inter-event interval and removing all bins in which no events are observed in any event train (Marshall et al., 2016; Timme et al., 2016). Additionally, two sub-sampling corrections are employed.

The original rationale for these procedures and detailed explanations are provided in Appendix D. Both the binning procedure and the zero-removal procedure are found to be problematic for data from PNNs and so were not used in the analysis. Also, due to the small number of electrodes used here, the sub-sampling procedure was unnecessary. Furthermore it is shown in Appendix E that the zero removal process can lead to non-zero complexity values even for purely random data, and so cannot be applied for data that comprises a small number of event trains.

We emphasise the relative simplicity of the data processing (currents measured by the acquisition system are directly converted to point-to-point changes in current) and in particular that no further pre-processing is required.

3.3. Neural complexity results

Fig. 6a shows a time-series plot of C_{N} for the experimental event raster (black, along with seven randomised versions of the data, each shown with a different colour, as discussed below). Each point indicates a 3600 s segment of the data. The average value of C_{N} is shown for 10 realisations of each randomisation procedure. The statistical uncertainties are comparable to the size of the datapoints and so are not visible on this scale. Clearly the value of C_{N} fluctuates throughout the measurement, consistent with time-varying internal correlations despite a constant input stimulus.

To better elucidate the characteristic properties of the data that contribute to neural complexity, and to enable a comparison to biological analyses, the neural complexity values of experimental data are compared to values obtained after randomising the data in five distinct ways (one method has a tunable parameter and so three values are investigated, providing seven randomisations for the comparison in total). These randomisation procedures, which are provided in the NCC Toolbox (Marshall et al., 2016) and described in Appendix C, each alter different properties of the data, and to different extents.

The binomial randomisation procedure (purple data in Fig. 6a) leads to a significant decrease in C_N for all data segments, essentially reducing C_N to zero, as expected since all correlations are destroyed. The three jitter procedures (1, blue; 10, orange; 100, yellow) decrease C_N in proportion to the amount of jitter (corresponding to the extent correlations have been destroyed). Each of the three curves corresponding to the jitter procedure traces a similar shape as the experimental curve. The shuffle and swap procedures (green and light blue) result in only small decreases in C_N for most data segments, and the shape of each curve mostly correlates to the experimental curve. Finally, the wrap procedure (maroon) produces a low value of C_N for most data segments, irrespective of the corresponding experimental value, as most spatial correlations are destroyed.

To facilitate an easy qualitative comparison with the neuronal data, Fig. 6b shows a box and whisker plot of the values of C_N from Fig. 6a, in the same format as originally presented by Timme et al. (2016). The experimental event raster has the highest median C_N value. The shuffle and swap procedures result in median C_N values approximately aligned with the minimum of the lower whisker for the experimental event raster. In neural culture data (Timme et al., 2016) the median values were only slightly lower than the experimental values. The difference between the experimental and jittered data for the PNNs is also greater. Hence, the C_N measure suggests that the PNN data is more complex than the data from neural cultures.

Direct comparison between the values of C_N obtained here and in the neuroscience literature is not possible due to the vastly different number of electrodes and the different pre-processing procedures (see Appendices A and D). However, it is clear that, as was observed for data from neural cultures, C_N is highest for the experimental data (in all cases randomisation removes complexity). Interestingly, the relative values of C_N for the experimental data (compared to the various randomised datasets) are significantly greater than those obtained from neural cultures. This result suggests that the PNN data is actually more complex than the data analysed by Timme et al. (2016).

4. Discussion

4.1. Complexity in PNNs and related systems

The complexity of the event trains obtained from a 10 electrode PNN with $m = 6$ outputs was analysed using several different measures of complexity. Measures of spatial (f_n) and temporal (f_ϕ) complexity both showed that the PNN data differed significantly from temporally randomised data. A measure of spatio-temporal complexity based on the pairwise covariance between event trains revealed complex time-varying correlations. Collectively, these three qualitative measures of complexity are consistent with intricate and complex spatio-temporal dynamics within the network.

The neural complexity metric C_N provides a quantitative measure of complexity, based on the local segregation and global integration of the data. Comparison with randomised versions of the data confirmed that the PNN data exhibits significant spatial complexity and is neither random nor ordered. [We note that complex data exists somewhere on a spectrum between random/independent data and ordered/dependent data, and that it would be interesting to develop methods to introduce additional correlations (spatial, temporal, and/or spatio-temporal) that allow further comparisons with data that is more ordered.]

The measures of complexity developed here will be valuable in allowing characterisation of other neuromorphic systems such as SNNs, and we hope that in subsequent study we will be able to compare directly with the complexity of other artificial systems such as SNNs (Chen et al., 2025; Li et al., 2024; Shen et al., 2024), PNNs fabricated with other materials (Pal et al., 2025; van der Ree et al., 2024), and self-assembled nanowire networks (Hochstetter et al., 2021; Milano et al., 2022b).

4.2. Comparison with biological data

Direct comparison with biological data is challenging because the number of electrical contacts to PNNs is currently much smaller than is possible with the commercial microelectrode array technology that is typically used in neuroscience experiments (Friedman et al., 2012). Since only $m = 6$ event trains were available here, it would be useful to explore complexity in PNNs with more electrodes. Devices with ~ 100 electrodes would be required to allow direct comparison with results from neuroscience—such devices are possible but challenging from a fabrication point of view.

We note that it was necessary to adapt the method developed for biological data in Timme et al. (2016) because there are some significant differences in the characteristics of the signals measured from PNNs and from the cortex. Two differences stand out. Firstly, the signals from PNNs are measured with microsecond-level time resolution which means that individual spiking events are much sharper than those measured in the biological system, and that there are often much larger numbers of timesteps between events. Secondly, the noise in the measured signals from PNNs is extremely low, and the signal to noise ratio is therefore much higher than is possible in the case of signals from the cortex. This means that, for example, power-law distributions of inter-event intervals from PNNs can be measured over more than six decades of time (Acharya et al., 2021). These differences underly the need to adapt the analysis of Timme et al. (2016) as described in Section 3.2 and Appendices D and E.

Investigations of complexity are ongoing in the neuroscience literature. For example, quantification of neural complexity has facilitated comparisons of brain functionality between healthy and impaired states. In these studies, neuronal data were measured through electroencephalography (EEG) (van Putten & Stam, 2001) and magnetoencephalography (MEG) (van Cappellen et al., 2003), under the hypothesis that a reduction in consciousness correlates with a decrease in neural complexity. The EEG study found an increase in neural complexity among patients with severe brain disorders (specifically epileptic seizures and postanoxic encephalopathy) compared to a healthy brain state. Similarly, the MEG study found an increase in neural complexity in patients with Alzheimer's disease relative to a healthy brain state. These findings challenge the originally posited hypothesis that neural complexity C_N is elevated during optimal information processing states (Tononi et al., 1994). Other investigations of neural complexity include those reporting (i) higher complexity in gifted students than average students (Jin et al., 2009), (ii) a decrease in complexity when subjects were subjected to anaesthetics that lower brain function (Varley et al., 2020), and (iii) elevated complexity in networks with higher densities of neurons (Heiney et al., 2022).

It should be further noted that it has been demonstrated that the neural complexity measure is ineffective when the system exhibits high causal density. Causal density is a distinct measure of dynamical complexity (Shanahan, 2008) that evaluates the influence of one event train on others, taking into account local integration and global segregation that is smeared out over time (Seth et al., 2006). Causal density may provide a useful alternative complexity metric in future investigations.

4.3. Complexity and computation

Since one of the aims of understanding and quantifying complexity is to facilitate computation, we note that self-organised nanoscale networks like PNNs have already been used for a variety of styles of computation. In physical reservoir computing schemes PNNs have been used for experimental demonstrations of time series prediction, nonlinear transformation, waveform discrimination, memory capacity, and speech recognition (Mallinson et al., 2023, 2024), as well as speaker and Braille recognition (Steel et al., 2025) and autonomous swarm generation (Heywood et al., 2024). In the case of nanowire networks, a wide range of computing tasks have been simulated including pattern recognition (Kuncic et al., 2020; Milano et al., 2022b), time series prediction (Fu et al., 2020; Milano et al., 2023, 2022b; Zhu et al., 2020), speech recognition (Milano et al., 2022a), and regression (Hochstetter et al., 2021; Loeffler et al., 2021; Zhu et al., 2021). In addition, simulations that have been used to investigate a range of issues related to computation, including the scalability of these systems (Daniels et al., 2022), the architecture of the reservoir (Daniels et al., 2023; Daniels & Brown, 2021; Mallinson et al., 2023), the impact of the positioning and number of electrodes (Heywood et al., 2022), and the dimensionality of the reservoir (Daniels et al., 2022).

We now consider the power consumption required for computation. The devices measured here typically operate with power consumption at ~mW levels, due to the relatively high conductance of the Sn nanoparticles. However, it has recently been shown that the power consumption in Mo nanoparticle networks is at ~nW levels (van der Ree et al., 2024), suggesting that the choice of nanoparticle material will ultimately be a factor in minimising power consumption in applications.

5. Conclusion

Previous research has unequivocally demonstrated that PNNs exhibit criticality (Mallinson et al., 2019). The literature suggests that criticality maximises both complexity (Timme et al., 2016) and information processing capability (Srinivasa et al., 2015). The demonstration of complexity presented here further motivates the use of spiking PNNs (i.e. operating in a high voltage regime Mallinson et al., 2019) as computational devices. [When operated in a low voltage regime PNNs do not exhibit spiking. In that regime, very different kinds of spatio-temporal correlations were successfully utilised in reservoir computing schemes (Heywood et al., 2024; Mallinson et al., 2023, 2024; Steel et al., 2025).] In fact, spiking PNNs were recently used to perform Boolean logic and classification of hand-written digits (Studholme et al., 2023), as well as probabilistic computation (Studholme & Brown, 2024). We note however that it is still not clear that complex or critical systems provide an advantage for specific computational tasks (Studholme et al., 2025), and so further research on this topic is urgently needed.

We conclude by highlighting that a variety of SNNs are under intense investigation, and in particular that both simulated and hardware implementations of SNNs exhibit a variety of important properties such as the ability to flexibly process many types of data and for learning via evolution of network structures (see Chen et al., 2025; Li et al., 2024; Shen et al., 2024 and refs therein). These systems are extremely interesting and are an important next target for application of the complexity metrics proposed here. It is important to investigate, for example, the way that complexity changes as the sparsity of the (spiking) network changes or is evolved during learning. In this context it is worth noting that even at the beginning of the development of PNNs (Bose et al., 2017) the aim was to develop applications through integration of PNNs with CMOS circuitry. Such hybrid systems might benefit from both low-cost self-assembly intrinsic physical processing of signals in PNNs and the standardised electronic circuitry and flexibility of deliberately engineered SNNs.

CRediT authorship contribution statement

Jamie K. Steel: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation; **Ford Wagner:** Writing – review & editing, Methodology, Investigation, Formal analysis, Data curation; **Edoardo Galli:** Writing – review & editing, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Susant K. Acharya:** Writing – review & editing, Investigation, Formal analysis; **Joshua B. Mallinson:** Writing – review & editing, Methodology, Investigation, Formal analysis; **Philip J. Bones:** Writing – review & editing, Supervision, Formal analysis; **Matthew D. Arnold:** Writing – review & editing, Supervision, Conceptualization; **Simon A. Brown:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Simon Brown reports financial support was provided by Royal Society of New Zealand Marsden Fund. Simon Brown has patent #US Patent 11,856,877 issued to University of Canterbury. Simon Brown has patent #US patent US11281968 pending to University of Canterbury. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Calculation of neural complexity C_N

The calculation of C_N comprises three distinct steps: calculation of entropy, integration, and complexity (Marshall et al., 2016; Tononi et al., 1994). Each of these steps is explained below, and a detailed example calculation is provided in Appendix B.

Note that the calculation requires consideration of subsets of the event raster, and in particular it is important to understand the notation used to label the subsets. Here, R^k_j refers to the jth subset of k event trains and, in particular, $R^m_1 \equiv \mathbf{R}$ (the entire event raster) and $R^1_j \equiv E_j$ (a single event train). Examples are given in Appendix B.

The first step of the neural complexity calculation involves computation of the entropy. For a particular subset R^k_j, the entropy is

$$H(R^k_j) = -\sum_{\mathbf{r}_i \in R^k_j} p(\mathbf{r}_i)\log_2(p(\mathbf{r}_i)) \tag{A.1}$$

where $p(\mathbf{r}_i)$ is the probability of a particular *combined state* $\mathbf{r}_i$ (i.e. a particular combination of electrodes at which events are observed for a single time-step) occurring in the subset R^k_j. The binary logarithm produces an entropy in units of bits (Shannon, 1948).

The integration is next defined as a measure of the entropy of a subset (joint entropy) compared with the sum of the individual entropies of the constituent event trains for that particular subset. Integration measures the degree to which the activity of the event trains are coordinated and, for a particular subset, is defined as

$$I(R^k_j) = \left(\sum_{j' \in k} H(R^1_{j'})\right) - H(R^k_j) \tag{A.2}$$

where the summation $j' \in k$ refers to the values of j' that correspond to the constituent event trains of the subset R^k_j.

Finally, neural complexity is calculated as

$$C_\mathrm{N}(\mathbf{R}) = \frac{1}{m}\sum_{k=2}^{m}\left[\left(\frac{k-1}{m-1}\right)I(\mathbf{R}) - \langle I(R^k_j)\rangle_j\right] \tag{A.3}$$

where the summation begins from $k = 2$ because Eq. (A.2) is always zero

Table B.1
The entropy and integration values for all subsets of the example event raster in Fig. B.1a. The mean integration over j is calculated independently for each subset of k event trains. Eq. (A.3) gives the neural complexity as $C_\mathrm{N}(\mathbf{R}) \simeq 0.0870$ bit/event train.

Subset size k	Event raster subset R_j^k	Entropy H (bit)	Integration I (bit)	Mean integration $\langle I(R_j^k)\rangle_j$ (bit)
1	$R_1^1 \equiv E_1$	0.8813	0	0.000
	$R_2^1 \equiv E_2$	0.9710	0	
	$R_3^1 \equiv E_3$	0.8813	0	
	$R_4^1 \equiv E_4$	1.0000	0	
2	R_1^2	1.7610	0.0913	0.123
	R_2^2	1.7610	0.0016	
	R_3^2	1.4855	0.3958	
	R_4^2	1.7610	0.0913	
	R_5^2	1.8464	0.1245	
	R_6^2	1.8464	0.0349	
3	R_1^3	2.5219	0.2116	0.4160
	R_2^3	2.3219	0.5303	
	R_3^3	2.2464	0.5161	
	R_4^3	2.4464	0.4058	
4	$R_1^4 \equiv \mathbf{R}$	2.8464	0.8871	0.8871

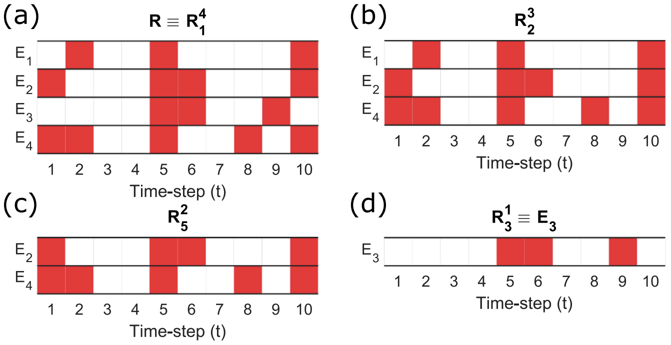


Fig. B.1. Four instances of the subset notation for an example event raster with four event trains. The red squares correspond to an event at a particular time-step; the white squares for no event. (a) The complete event raster, or equivalently, the first unique set of four event trains, $\mathbf{R} \equiv R_1^4$; (b) the second unique set of three event trains, R_2^3; (c) the fifth unique set of two event trains, R_5^2; (d) the third unique set of one event train, or equivalently, the third event train, $R_3^1 \equiv E_3$. This figure is provided for clarity to complement the example neural complexity calculation.

for $k = 1$. Here, $\langle I(R_j^k)\rangle_j$ refers to the average integration over all j for some k, while $I(\mathbf{R})$ denotes the integration of the entire event raster (total integration) and is constant throughout the summation. Thus, the first term in the square parentheses is a linear approximation to the integration for a subset size k, given the total integration. Therefore, neural complexity is the difference between the linear approximation to the integration and the average integration, summed across all subset sizes k. High neural complexity is found when the average integration differs greatly from the linear approximation. Conversely, low neural complexity is found when those two terms are approximately equal.

In summary, each step of the neural complexity calculation performs a distinct function. The first step (Eq. (A.1)) assigns an entropy value to each event raster subset. The integration step (Eq. (A.2)) differentiates between dependent (high value) and independent (low value) data. The complexity step (Eq. (A.3)) also differentiates between dependent (low value) and independent (high value) data; however, the type of data yielding a low value is reversed in comparison to the integration step. Consequently, complex data can be differentiated from dependent and independent data. For dependent/ordered data, the complexity step yields a low value, resulting in a low C_N (left panel of Fig. 5b). For independent/random data, the integration step yields a low value, resulting in a low C_N (right panel of Fig. 5b). For complex data, neither the integration step nor the complexity step yields a low value, resulting in a high C_N (middle panel of Fig. 5b).

Appendix B. Example neural complexity C_N calculation

The neural complexity calculation embodied in Eqs. (A.1) to (A.3) is now illustrated step by step for the example event raster shown in Fig. B.1a. This example event raster is comprised of four event trains. Only one event is observed at all electrodes, and there are three time-steps at which no event is observed. The entropy, integration, and average integration values for all subsets R_j^k for this example are listed in Table B.1.

In the first step, the entropy H of the individual event trains (i.e. $R_j^1 \equiv E_j$) is calculated from Eq. (A.1). For example, in R_3^1 (the event train E_3, see Fig. B.1d) there are two possible combined states $\mathbf{r}_i$: no event (0) and event (1)[1]. These have probabilities of $p(0) = \frac{7}{10}$ and $p(1) = \frac{3}{10}$, respectively, i.e. the proportion of time-steps for which each combined state occurs. Thus, the entropy is

$$\begin{aligned} H(R_1^1 \equiv E_1) &= -\left(\frac{7}{10}\log_2\left(\frac{7}{10}\right) + \frac{3}{10}\log_2\left(\frac{3}{10}\right)\right) \\ &\simeq 0.88\text{bit}. \end{aligned}$$

The remaining entropy values for R_j^1 with $j = 2, 3, 4$ are found in a similar fashion (i.e. the other individual event trains, see Table B.1). Entropy is easily extended to larger subsets. For example, for the entire event raster (i.e. $\mathbf{R} \equiv R_1^4$) there are $2^4 = 16$ possible combined states, corresponding to the combinations of four binary options at each time-step. Eight combined states do not appear in the event raster in Fig. B.1a and so have probability $p(\mathbf{r}_i) = 0$. Of the eight states that do appear, one appears three times (all 0's) and so has probability $p([0,0,0,0]) = \frac{3}{10}$ while the remaining seven appear only once with $p(\mathbf{r}_i) = \frac{1}{10}$. Thus, the total entropy can be obtained from Eq. (A.1) and the above eight probabilities and is found to be $H(\mathbf{R} \equiv R_1^4) \simeq 2.85$bit.

In the second step, the integration I is found from Eq. (A.2). Given that there are four event trains in the example event raster, there are four subset sizes k (and for each k there are $\binom{4}{k}$ different j values) to consider. For example, for R_2^3 (the second set of $k = 3$ event trains, E_i for $i = 1, 2, 4$, see Fig. B.1b) the integration is

$$\begin{aligned} I(R_2^3) &= \left(H(R_1^1) + H(R_2^1) + H(R_4^1)\right) - H(R_2^3) \\ &\simeq (0.881 + 0.971 + 1.000) - 2.322 \\ &\simeq 0.530\text{bit}. \end{aligned}$$

The remaining integration values for R_j^3 with $j = 1, 3, 4$ are found in a similar fashion (see Table B.1). Now, the average of the integration values for all four j values is

$$\begin{aligned} \langle I(R_j^k)\rangle_j &= \frac{1}{4}\sum_{i=1}^{4} I(R_i^3) \\ &\simeq \frac{1}{4}(0.212 + 0.530 + 0.516 + 0.406) \\ &\simeq 0.416\text{bit}. \end{aligned}$$

Finally, in the third step, the neural complexity C_N is found from Eq. (A.3). C_N is the normalised difference between a linear approximation for the expected integration (across all subset size scales k given the total integration) and the average integration. For the example event

[1] For single event trains, the combined states collapse to scalar values. Additionally, with only two possible combined states, the entropy reaches a maximum H = 1 when p(ri) = 0.5 and a minimum H = 0 when p(ri) = 1.

raster in Fig. B.1a, this is

$$
\begin{aligned}
C_{\mathrm{N}}(\mathbf{R}) &= \frac{1}{4}\Big[\Big(\frac{1}{3}I(\mathbf{R}) - \langle I(R_j^2)\rangle_j\Big) + \Big(\frac{2}{3}I(\mathbf{R}) - \langle I(R_j^3)\rangle_j\Big) \\
&\quad + \Big(\frac{3}{3}I(\mathbf{R}) - \langle I(R_j^4)\rangle_j\Big)\Big] \\
&\simeq \frac{1}{4}\Big[\Big(\frac{1}{3}(0.887) - (0.123)\Big) + \Big(\frac{2}{3}(0.887) - (0.416)\Big) \\
&\quad + \Big(\frac{3}{3}(0.887) - (0.887)\Big)\Big] \\
&\simeq 0.087\mathrm{bit/eventtrain}.
\end{aligned}
$$

The non-zero C_{N} result means there is a clear difference between the linear approximation to the integration terms $\frac{k-1}{3}I(\mathbf{R})$ and the corresponding average integration $\langle I(R_j^k)\rangle_j$. This indicates that the data contained in the example event raster is neither random nor ordered (see middle panel of Fig. 5a). Note that because of the small size of the example event raster, it is not possible to say definitively that the example in Fig. B.1a is complex.

Appendix C. Randomisation procedures

There are several randomisation procedures defined in the neuroscience literature and provided in the NCC Toolbox (Marshall et al., 2016). Each of these procedures alters the positions of events in the event raster in various distinct ways, allowing for an in-depth analysis of how disruptions to the data impact the complexity. Table C.2 presents, in context with the particular structure and notation of the event raster data, a brief description of the five procedures. Fig. C.1 illustrates the effect that each procedure has on an example event raster. Table C.1 shows how each procedure disrupts three characteristic properties of the event raster.

Table C.1
The characteristic properties of the event raster that are affected by the randomisation procedures described in Table C.2. Adapted from the original of Timme et al. (2016).

Procedure	Event rate	Inter-event interval (IEI)	Avalanche profile
Binomial	No	Yes	Yes
Jitter	No	Yes[a]	Yes[a]
Shuffle	Yes	Yes	No
Swap	No	Yes	No
Wrap	No	No[b]	Yes

[a] Jitter only partially disrupts this property. The disruption is proportional to the standard deviation used.

[b] Wrap only disrupts a single IEI at the point of the cut.

Table C.2
Brief descriptions of the five randomisation procedures provided in code in the NCC Toolbox (Marshall et al., 2016).

Procedure	Description
Binomial[a]	Independently convert each event train to a binomial point process.
Jitter[b]	Independently relocate each event in each event train using a Gaussian distribution, with an adjustable standard deviation, centred on the time-step at which the event occurred.
Shuffle[c]	Independently convert each combined state to a binomial point process.
Swap[a]	Swap each event with an event from a different event train such that the number of events at a time-step t remains unchanged.
Wrap	Independently slice each event train into two pieces at a random time-step t which are subsequently recombined in the opposite order.

[a] Designated *Poisson* in the NCC Toolbox (Marshall et al., 2016) and referred to as *temporal randomisation* throughout Section 3.1.

[b] The original description by Rolston et al. (2007) contains further insightful visual explanations.

[c] Designated *Swap2* in the NCC Toolbox (Marshall et al., 2016).

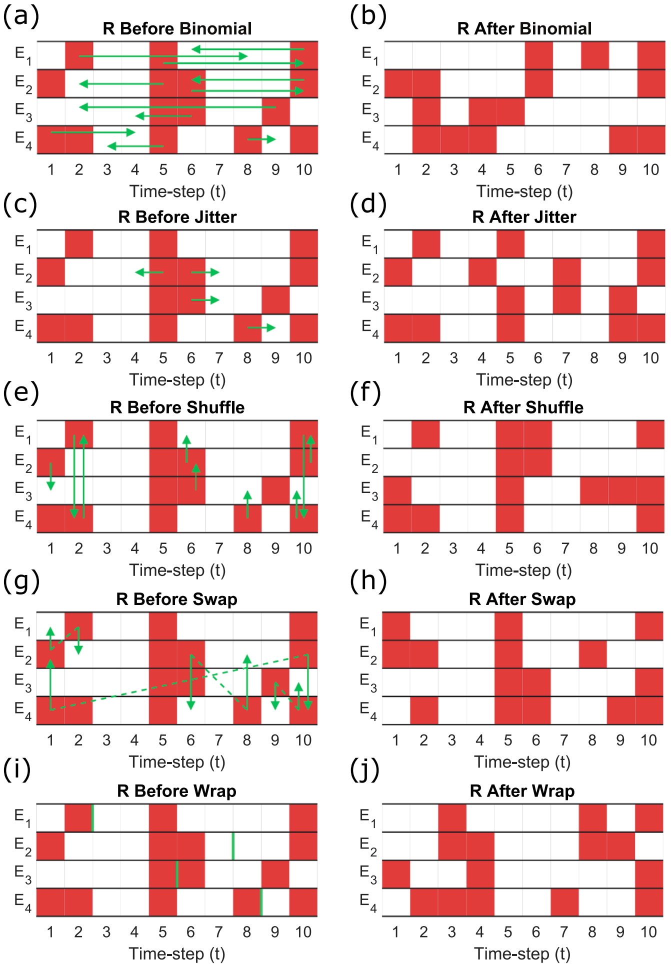


Fig. C.1. Visual representations of the five randomisation procedures described in Table C.2 for an example event raster. The red squares correspond to an event at a particular time-step; the white squares for no event. The green arrows in (a, c, e, and g) indicate where an event is shifted to. In (c), the amount of time-steps an event is shifted depends on the standard deviation of the Gaussian distribution used, i.e. a larger standard deviation gives higher probability to events shifting more time-steps. The dashed lines in (g) indicate the two events that are swapped between event trains. The vertical green lines in (i) indicate the slicing position. Panels (b, d, f, h, and j) show the resultant event rasters corresponding to panels (a, c, e, g, and i), respectively.

Appendix D. Data pre-processing & sub-sampling

In this appendix we discuss the pre-processing and sub-sampling procedures that were implemented by Marshall et al. (2016), Timme et al. (2016) as additional steps in the calculation of the neural complexity measure described in Section 3.2. The two pre-processing steps—binning of the data and the removal of all instances of the combined state in which no events occur—as well as the two sub-sampling corrections are described below, along with explanations as to why they are not applicable to the data analysed in this work.

Binning. It is common in measurements of neuronal spiking in biological systems that single events occur over multiple time-steps, i.e. in typical biological experiments the duration of events is much longer than the sampling interval. In contrast, for PNNs, events occur at only a single time-step, i.e. the duration of events is much shorter than the sampling interval. The binning procedure creates a new (shorter) event raster. For each event train, if one or more events occur within a bin of length b_l, the bin contains a count. Fig. D.1b illustrates the effect of the binning procedure with $b_l = 2$ applied to the example event raster

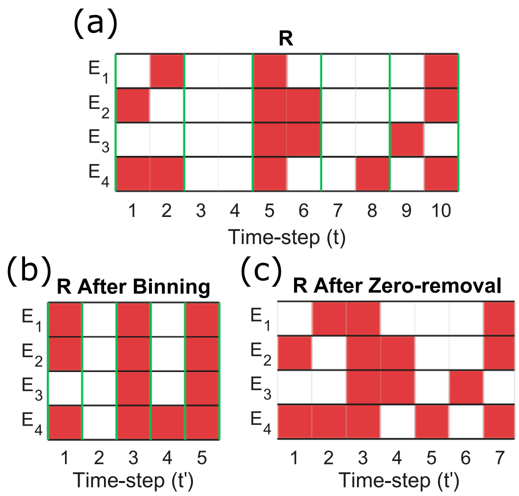


Fig. D.1. The two pre-processing procedures applied to an example event raster. The red squares correspond to an event at a particular time-step; the white squares for no event. (a) The event raster before applying either procedure. (b) The event raster after applying the binning procedure (without zero-removal). Two consecutive time-steps (bin length of $b_l = 2$, vertical green lines in (a)) have been merged together. (c) The event raster after applying the zero-removal procedure (without binning). Time-steps 3, 4, and 7 in (a) are the same combined state with no observed events on any event train and thus have been removed.

in Fig. D.1a. Marshall et al. (2016), Timme et al. (2016) used $b_l = \langle \mathrm{IEI} \rangle$ (the average inter-event interval) as the best representative timescale of the system. In contrast, for PNNs, because the sampling interval is greater than the duration of an event, the only appropriate choice for the complexity analysis is $b_l = 1$.

Zero-removal. Zero-removal refers to the removal of all instances of the combined state in which no events are observed on any of the event trains from the event raster (Marshall et al., 2016; Timme et al., 2016). This ensures that only the complexity associated with avalanches is considered, and aims to prevent the numerical neural complexity measure (Section 3.2) being biased by periods where no activity is present. If a majority of the time-steps have no events, the event raster would be described as ordered rather than complex, and C_N would contain little information about the patterns of observed events.

Fig. D.1c illustrates the effect of the zero-removal procedure applied to the example event raster in Fig. D.1a. Simply, the columns of the event raster corresponding to time-steps 3, 4, and 7 are removed. However, as demonstrated in Appendix E, zero-removal leads to dramatic adverse effects for the neural complexity calculation. Specifically, truly independent data produces non-zero neural complexity values that are comparable to, or even exceed, those derived from experimental data that necessarily exhibits a non-zero degree of complexity. This observation might be less significant in data involving hundreds of outputs, which could explain its absence in previous studies. Therefore, the zero-removal procedure, as established in the neuroscience literature (Marshall et al., 2016; Timme et al., 2016) was not used for the complexity analysis detailed here.

Sub-sampling corrections. In the neuroscience literature (Marshall et al., 2016; Timme et al., 2016) the use of multi-electrode arrays makes it possible to measure spiking at 100 electrodes simultaneously. This means that there are $2^{100} \simeq 10^{30}$ possible states (as defined in Section 3.2), which is more than is computationally feasible to analyse. Hence, a sub-sampling correction was developed to address a possible underestimation of the joint entropy (Marshall et al., 2016). A further correction is then necessitated in the case that there is a negative second derivative in the subset size k. For the PNNs with six electrodes used here, there are only $2^6 = 64$ possible states, which can all be easily analysed and thus no sub-sampling correction is needed. Sub-sampling corrections would only be needed if a much larger number of electrodes were fabricated ($30+$).

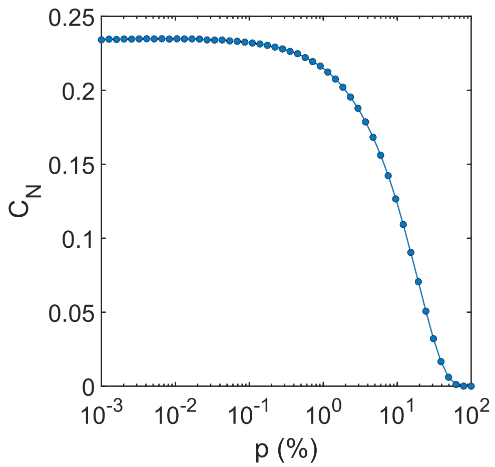


Fig. E.1. Comparison of C_N for event rasters populated with independent random events generated by a Poisson point process with probability p. The zero-removal procedure has been applied (but there is no binning). The event rasters are created from $m = 6$ event trains with the equivalent number of time-steps to a 3600 s segment of experimental data with a 1 kHz sampling rate. Each point represents the average C_N over ten independent realisations, however, the variance is negligible.

Appendix E. Discussion on zero-removal

In this appendix we demonstrate that for a small number of event trains the zero-removal process in itself is problematic because it leads to non-zero values of C_N even when the data is random.

Poisson-generated random data. Random data is generated via a Poisson point process. The probability for an event to occur at each time-step of each event train is p. The length of the event trains is 3.6×10^6 time-steps, i.e. the same as a 3600 s segment of experimental data recorded with a 1 kHz sampling rate. The random event raster is then created from m independent event trains. The zero-removal procedure is then applied to the random event raster.

Fig. E.1 shows C_N as a function of p. Each point represents the average of ten independent realisations, however, the uncertainty is negligible. Clearly C_N is highly dependent on p, and $C_N \neq 0$ for $p \lesssim 50\,\%$, which demonstrates that the neural complexity measure is providing an erroneous impression of the level of the complexity in the data. As emphasised in the discussion of Fig. 5, random data should lead to $C_N = 0$. Note that the values of C_N for low p are notably larger than for any calculated for experimental data from the PNNs (typically $C_N \simeq 0.15$).

This problematic result is a direct consequence of the zero-removal procedure. [Recall that the zero-removal procedure removes all instances of the combined state in the event raster for which no events are recorded on any event train (see Appendix D).] The zero-removal procedure was originally justified on the grounds that it mitigates biases associated with varying numbers of avalanches (Marshall et al., 2016). In the next section we show that if only a small number of electrodes are available this leads to a non-zero value of C_N for low p.

Effect of the number of electrodes. The experimental event rasters are limited to $m = 6$ event trains, which is a consequence of the ten-electrode device geometry used here. [To allow direct comparison, the Poisson-generated rasters contain the same number of event trains.] For each event train, there is then a $1 - p$ probability that no event occurs in any time step, and thus a probability $(1 - p)^m$ that no event occurs in any of

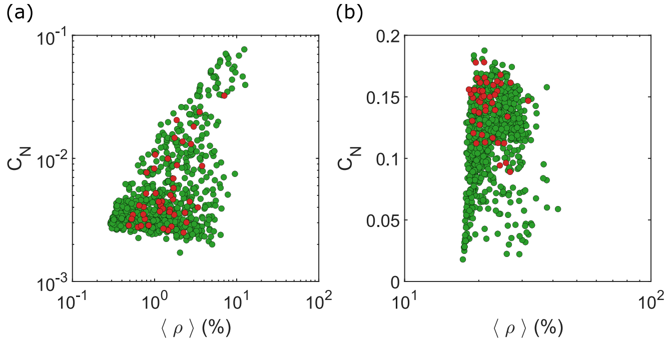


Fig. E.2. Scatter-plot between neural complexity C_N and mean event rate $\langle \rho \rangle$ for consecutive segments of the data. The red and green markers correspond to $s_l = 200\,\mathrm{s}$ and $3600\,\mathrm{s}$, respectively. (a) Without the zero-removal procedure a moderate positive correlation is present. (b) With the zero-removal procedure there is instead a weak negative correlation that weakens further as the segment length increases. Note that C_N is shown on a logarithmic scale in (a) and a linear scale in (b).

the m event trains. Hence, for $p = 1\,\%$ and $m = 6$, ∼94 % of the time-steps are instances of the combined state to be removed, i.e. the zero-removal procedure has a large effect on the event raster and hence on the neural complexity calculation. In contrast, in neuroscience measurements there can be $m > 100$ constituent event trains (Marshall et al., 2016), and only ∼37 % of the time-steps are this same combined state are removed *from a random event raster*. Hence we conclude that the zero-removal procedure is appropriate when m is large but is inappropriate for small m because it means that the neural complexity metric is unable to distinguish between random and complex data.

Effect of event rate. The dependence of C_N on p for zero-removed data is equivalent to a dependence on the event rate (total number of events over time). Hence, the zero-removal procedure means that it is not possible to directly compare C_N values for segments of data that have different event rates (e.g. to conclude one segment is more complex than another). Fig. E.2a and b show C_N versus the mean event rate $\langle \rho \rangle$ for segments of the experimental data before (a) and after (b) the zero-removal procedure is applied, respectively. The colours correspond to different segment lengths s_l. Fig. E.2a (note the logarithmic C_N vertical scale) shows a moderate positive correlation between C_N and $\langle \rho \rangle$, while Fig. E.2b shows a weak negative correlation. This effect is observed because the zero-removal procedure has a greater effect on data with a low event rate compared with data with a high event rate (as there are more instances of the combined state that is to be removed). Thus, the impact on neural complexity is more severe. It is worth noting, however, that the variance in event rate decreases as the segment length s_l increases. The zero-removal procedure has a more comparable effect on different segments of data as the event rate tends towards the dataset average.

References

Acharya, S. K., Galli, E., Mallinson, J. B., Bose, S. K., Wagner, F., Heywood, Z. E., Bones, P. J., Arnold, M. D., & Brown, S. A. (2021). Stochastic spiking behavior in neuromorphic networks enables true random number generation. *ACS Applied Materials and Interfaces*, *13*(44), 52861–52870.

Adami, C. (2002). What is complexity? *BioEssays*, *24*, 1085–1094.

Akai-Kasaya, M., Takeshima, Y., Kan, S., Nakajima, K., Oya, T., & Asai, T. (2022). Performance of reservoir computing in a random network of single-walled carbon nanotubes complexed with polyoxometalate. *Neuromorphic Computing and Engineering*, *2*(1), 014003.

Avizienis, A. V., Sillin, H. O., Martin-Olmos, C., Shieh, H. H., Aono, M., Stieg, A. Z., & Gimzewski, J. K. (2012). Neuromorphic atomic switch networks. *PloS One*, *7*(8), e42772.

Bar-Yam, Y. (1999). Dynamics of complex systems. (1st ed.). Cambridge, MA, United States: CRC Press.

Beggs, J. M., & Plenz, D. (2003). Neuronal avalanches in neocortical circuits. *Journal of Neuroscience*, *23*, 11167–11177.

Bose, S. K., Mallinson, J. B., Gazoni, R. M., & Brown, S. A. (2017). Stable self-assembled atomic-switch networks for neuromorphic applications. *IEEE Transactions on Electron Devices*, *64*(12), 5194–5201.

Bullmore, E., & Sporns, O. (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, *10*, 186–198.

van Cappellen, A. M., van Walsum, A., Pijnenburg, Y. A. L., Berendse, H. W., Dijk, B. W. V., Knol, D. L., Scheltens, P., & Stam, C. J. (2003). A neural complexity measure applied to MEG data in Alzheimer's disease. *Clinical Neurophysiology*, *114*, 1034–1040.

Chen, P., Zhang, B., He, E., Xiao, Y., Liu, F., Lin, P., Wang, Z., & Pan, G. (2025). Towards scalable memristive hardware for spiking neural networks. *Materials Horizons*, *12*, 2820–2839.

Christensen, D. V., Dittmann, R., Linares-Barranco, B., Sebastian, A., Le Gallo, M., Redaelli, A., Slesazeck, S., Mikolajick, T., Spiga, S., Menzel, S., Valov, I., Milano, G., Ricciardi, C., Liang, S. J., Miao, F., Lanza, M., Quill, T. J., Keene, S. T., Salleo, A., … Pryds, N. (2022). 2022 roadmap on neuromorphic computing and engineering. *Neuromorphic Computing and Engineering*, *2*(2), 022501.

Dale, M., Miller, J. F., Stepney, S., & Trefzer, M. A. (2016). Evolving carbon nanotube reservoir computers. In M. Amos, & A. Condon (Eds.), *UCNC 2016*. Lecture notes in computer science (pp. 49–61). Manchester, UK (vol. *9726*).

Daniels, R. K., Arnold, M. D., Heywood, Z. E., Mallinson, J. B., Bones, P. J., & Brown, S. A. (2023). Brain-like networks of nanowires and nanoparticles: A change of perspective. *Physical Review Applied*, *20*, 034021.

Daniels, R. K., & Brown, S. A. (2021). Nanowire networks: How does small-world character evolve with dimensionality? *Nanoscale Horizons*, *6*, 482.

Daniels, R. K., Mallinson, J. B., Heywood, Z. E., Bones, P. J., Arnold, M. D., & Brown, S. A. (2022). Reservoir computing with 3D nanowire networks. *Neural Networks*, *154*, 122–130.

Deco, G., Tononi, G., Boly, M., & Kringelbach, M. L. (2015). Rethinking segregation and integration: Contributions of whole-brain modelling. *Nature Reviews Neuroscience*, *16*, 430–439.

Friedman, N., Ito, S., Brinkman, B. A. W., Shimono, M., DeVille, R. E. L., Dahmen, K. A., Beggs, J. M., & Butler, T. C. (2012). Universal critical dynamics in high resolution neuronal avalanche data. *Physical Review Letters*, *108*, 208102.

Fu, K., Zhu, R., Loeffler, A., Hochstetter, J., Diaz-Alvarez, A., Stieg, A., Gimzewski, J., Nakayama, T., & Kuncic, Z. (2020). Reservoir computing with neuromemristive nanowire networks. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–8).

Green, S., Şerban, M., Scholl, R., Jones, N., Brigandt, I., & Bechtel, W. (2018). Network analyses in systems biology: New strategies for dealing with biological complexity. *Synthese*, *195*, 1751–1777.

Heiney, K., Huse Ramstad, O., Fiskum, V., Sandvig, A., Sandvig, I., & Nichele, S. (2022). Neuronal avalanche dynamics and functional connectivity elucidate information propagation in vitro. *Frontiers in Neural Circuits*, *16*, 980631.

Heywood, Z., Mallinson, J., Galli, E., Acharya, S., Bose, S., Arnold, M., Bones, P., & Brown, S. A. (2022). Self-organized nanoscale networks: Are neuromorphic properties conserved in realistic device geometries? *Neuromorphic Computing and Engineering*, *2*, 024009.

Heywood, Z. E., Mallinson, J. B., Bones, P. J., & Brown, S. A. (2024). From 'follow the leader' to autonomous swarming: physical reservoir computing in two dimensions. *Neuromorphic Computing and Engineering*, *4*(3), 034011.

Hochstetter, J., Zhu, R., Loeffler, A., Diaz-Alvarez, A., Nakayama, T., & Kuncic, Z. (2021). Avalanches and edge-of-chaos learning in neuromorphic nanowire networks. *Nature Communications*, *12*(1), 4008.

Jin, S. H., Kim, S. Y., Park, K. H., & Lee, K. J. (2009). Differences in EEG between gifted and average students: Neural complexity and functional cluster analysis. *International Journal of Neuroscience*, *117*, 1167–1184.

Kaiser, W. (2024). The evolvement of lithography optics towards advanced EUV lithography: Enabling the continuation of Moore's law for six decades. *IEEE Electron Devices Magazine*, *2*(1), 23–34.

Kandel, E. R., Koester, J. D., Mack, S. H., & Siegelbaum, S. A. (2021). Principles of neural science. (6th ed.). New York, NY, United States: McGraw Hill.

Kuncic, Z., Kavehei, O., Zhu, R., Loeffler, A., Fu, K., Hochstetter, J., Li, M., Shine, J. M., Diaz-Alvarez, A., Stieg, A., Gimzewski, J., & Nakayama, T. (2020). Neuromorphic information processing with nanowire networks. In *2020 IEEE international symposium on circuits and systems (ISCAS)* (pp. 1–5).

Li, Y., Xu, Q., Shen, J., Xu, H., Chen, L., & Pan, G. (2024). Towards efficient deep spiking neural networks construction with spiking activity based pruning. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, & F. Berkenkamp (Eds.), *Proceedings of the 41st international conference on machine learning* (pp. 29063–29073). PMLR (vol. *235*). Proceedings of Machine Learning Research.

Loeffler, A., Zhu, R., Hochstetter, J., Diaz-Alvarez, A., Nakayama, T., Shine, J. M., & Kuncic, Z. (2021). Modularity and multitasking in neuro-memristive reservoir networks. *Neuromorphic Computing and Engineering*, *1*, 014003.

Mallinson, J. B., Heywood, Z. E., Daniels, R. K., Arnold, M. D., Bones, P. J., & Brown, S. A. (2023). Reservoir computing using networks of memristors: effects of topology and heterogeneity. *Nanoscale*, *15*(22), 9663–9674.

Mallinson, J. B., Shirai, S., Acharya, S. K., Bose, S. K., Galli, E., & Brown, S. A. (2019). Avalanches and criticality in self-organized nanoscale networks. *Science Advances*, *5*, eaaw8438.

Mallinson, J. B., Steel, J. K., Heywood, Z. E., Studholme, S. J., Bones, P. J., & Brown, S. A. (2024). Experimental demonstration of reservoir computing with self-assembled percolating networks of nanoparticles. *Advanced Materials*, *36*(29), 2402319.

Marshall, N., Timme, N. M., Bennett, N., Ripp, M., Lautzenhiser, E., & Beggs, J. M. (2016). Analysis of power laws, shape collapses, and neural complexity: New techniques and MATLAB support via the NCC toolbox. *Frontiers in Physiology*, *7*, 00250.

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

Milano, G., Agliuzza, M., De Leo, N., & Ricciardi, C. (2022a). Speech recognition through physical reservoir computing with neuromorphic nanowire networks. In *2022 international joint conference on neural networks (IJCNN)* (pp. 1–6). IEEE.

Milano, G., Chakrabarty, T., & Ricciardi, C. (2023). Mackey-glass time series forecasting by nanowire networks. In *2023 IEEE international conference on metrology for extended reality, artificial intelligence and neural engineering (metroXRAINE)* (pp. 989–994). IEEE.

Milano, G., Pedretti, G., Montano, K., Ricci, S., Hashemkhani, S., Boarino, L., Ielmini, D., & Ricciardi, C. (2022b). In materia reservoir computing with a fully memristive architecture based on self-organizing nanowire networks. *Nature Materials*, *21*(2), 195–202.

Muñoz, M. A. (2018). Colloquium: Criticality and dynamical scaling in living systems. *Reviews of Modern Physics*, *90*, 031001.

Pal, M., Vidhyadhiraja, N. S., & Kulkarni, G. U. (2025). Realising bio-synapse analogous paired-pulse facilitation (PPF) and release inactivation (RI) and emulating “tip of the tongue” experience through conductance and retention dynamics in ag nano-labyrinth-based neuromorphic device. *Advanced Functional Materials*, *35*, 2425635.

van Putten, M. J. A. M., & Stam, C. J. (2001). Application of a neural complexity measure to multichannel EEG. *Physics Letters A*, *281*, 131–141.

van der Ree, A. J. T., Ahmadi, M., Ten Brink, G. H., Kooi, B. J., & Palasantzas, G. (2024). Stable millivolt range resistive switching in percolating molybdenum nanoparticle networks. *ACS Applied Materials and Interfaces*, *16*(47), 65157–65164.

Reichel, R., Partridge, J. G., Dunbar, A. D. F., Brown, S. A., Caughley, O., & Ayesh, A. (2006). Construction and application of a UHV compatible cluster deposition system. *Journal of Nanoparticle Research*, *8*, 405–416.

Rolston, J. D., Wagenaar, D. A., & Potter, S. M. (2007). Precisely timed spatiotemporal patterns of neural activity in dissociated cortical cultures. *Neuroscience*, *148*, 294–303.

Seth, A. K., Izhikevich, E., Reeke, G. N., & Edelman, G. M. (2006). Theories and measures of consciousness: An extended framework. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 10799–10804.

Shanahan, M. (2008). Dynamical complexity in small-world networks of spiking neurons. *Physical Review E*, *78*, 041924.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*, 379–423.

Shen, J., Ni, W., Xu, Q., & Tang, H. (2024). Efficient spiking neural networks with sparse selective activation for continual learning. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 611–619). (vol. *38*).

Shirai, S., Acharya, S. K., Bose, S. K., Mallinson, J. B., Galli, E., Pike, M. D., Arnold, M. D., & Brown, S. A. (2020). Long-range temporal correlations in scale-free neuromorphic networks. *Network Neuroscience*, *4*(2), 432–447.

Srinivasa, N., Stepp, N. D., & Cruz-Albrecht, J. (2015). Criticality as a set-point for adaptive behavior in neuromorphic hardware. *Frontiers in Neuroscience*, *9*(12), 00449.

Steel, J. K., Mallinson, J. B., Heywood, Z. E., Studholme, S. J., Bones, P. J., Müller-Cleve, S. F., Bartolozzi, C., & Brown, S. A. (2025). Speaker recognition and braille classification with a physical reservoir computing system. In *Proceedings of the 2025 international joint conference on neural networks (IJCNN)*. Rome, Italy.

Studholme, S. J., & Brown, S. A. (2024). Integer factorization with stochastic spiking in percolating networks of nanoparticles. *ACS Nano*, *18*, 28060–28069.

Studholme, S. J., Heywood, Z. E., Mallinson, J. B., Steel, J. K., Bones, P. J., Arnold, M. D., & Brown, S. A. (2023). Computation via neuron-like spiking in percolating networks of nanoparticles. *Nano Letters*, *23*(22), 10594–10599.

Studholme, S. J., Mallinson, J. B., Steel, J. K., & Brown, S. A. (2025). Probabilistic computing with percolating nanoparticle networks using experimental data with signatures of criticality. *Neuromorphic Computing and Engineering*, *5*(1), 014017.

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., & Hirose, A. (2019). Recent advances in physical reservoir computing: A review. *Neural Networks*, *115*, 100–123.

Tanaka, H., Azhari, S., Usami, Y., Banerjee, D., Kotooka, T., Srikimkaew, O., Dang, T.-T. T., Murazoe, S., Oyabu, R., Kimizuka, K., & Hakoshima, M. (2022). In-materio computing in random networks of carbon nanotubes complexed with chemically dynamic molecules: A review. *Neuromorphic Computing and Engineering*, *2*(2), 022002.

Timme, N. M., Marshall, N. J., Bennett, N., Ripp, M., Lautzenhiser, E., & Beggs, J. M. (2016). Criticality maximizes complexity in neural tissue. *Frontiers in Physiology*, *7*, 00425.

Tononi, G., Sporns, O., & Edelman, G. M. (1994). A measure for brain complexity: Relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences of the United States of America*, *91*(11), 5033–5037.

Varley, T. F., Sporns, O., Puce, A., & Beggs, J. (2020). Differential effects of propofol and ketamine on critical brain dynamics. *PLoS Computational Biology*, *16*(12), e1008418.

Zhu, R., Hochstetter, J., Loeffler, A., Diaz-Alvarez, A., Nakayama, T., Lizier, J. T., & Kuncic, Z. (2021). Information dynamics in neuromorphic nanowire networks. *Scientific Reports*, *11*(1), 13047.

Zhu, R., Hochstetter, J., Loeffler, A., Diaz-Alvarez, A., Stieg, A., Gimzewski, J., Nakayama, T., & Kuncic, Z. (2020). Harnessing adaptive dynamics in neuro-memristive nanowire networks for transfer learning. In *2020 international conference on rebooting computing* (p. 102).