Network modeling of complex systems: criticality, robustness, and computation

Vom Fachbereich für Physik und Elektrotechnik
der Universität Bremen

zur Erlangung des akademischen Grades eines
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation

von
Lorenz Tobias Baumgarten, M. Sc.
aus Buchholz in der Nordheide

Erstgutachter	Prof. Dr. Stefan Bornholdt
Zweitgutachter	Prof. Dr. Thilo Gross
Eingereicht am	10.02.2022
Tag des Promotionskolloquiums	20.05.2022

Abstract

Complex and adaptive networks are ubiquitous in many fields of scientific study, ranging from biological to social and communication networks, and can produce interesting and vital emerging phenomena, such as self-organized criticality. In this thesis, we study complex and adaptive networks in four different applications.

Our first field of study regards neural science, specifically brain criticality, which is hypothesized to be vital for the functioning of brains. In three papers, we study the presence of criticality in high-degree threshold networks and find a new critical point with dynamics more similar to real brain dynamics than previous high-degree critical points in such networks. Additionally, we develop algorithms that can tune networks towards such criticality, providing ideas for how criticality might be maintained in real brain networks.

Our second field of interest is epidemiology. Here, networks can be used to model contact between members of society and the spread of infectious diseases. We study the efficacy of a recursive contact tracing algorithm that attempts to predict the spread of a disease and quarantine possibly infectious people accordingly to combat a disease with a finite asymptomatic infection rate. We develop analytical calculations, supported by simulations, for the reduction of a disease's infections using this algorithm and find that recursive contact tracing can combat diseases that could not be controlled with classical tracing of only first contacts.

The third field is epigenetics, in which genetic networks are often modeled as simple Boolean networks. We hypothesize that genetic networks must be robust to noise, due to the environment in which they must function to facilitate life, and test this hypothesis by comparing the robustness of a number of real genetic networks to random networks. We find a higher robustness of the real networks compared to randomized variants and further trace the origins of this robustness to a combination of the networks' attractors themselves as well as the underlying network topology.

Finally, to spark ideas for amorphous computing, we develop a computing scheme using collision-based computing in an irregular, two-dimensional threshold network. We show that interactions of gliders of activity traversing the network can be used to create a universal set of Boolean gates that can be used to facilitate universal computing.

Kurzfassung

Komplexe und adaptive Netzwerke sind allgegenwärtig in vielen wissenschaftlichen Gebieten, von biologischen bis sozialen und Kommunikations-Netzen, und können interessante und lebenswichtige emergente Phänomene, wie zum Beispiel selbstorganisierte Kritizität, produzieren. In dieser Arbeit untersuchen wir komplexe und adaptive Netzwerke in vier verschiedenen Anwendungsbereichen.

Unser erster Forschungsbereich behandelt Neurowissenschaft, speziell Gehirn Kritizität, von der wir glauben, dass sie unerlässlich für ein funktionierendes Gehirn ist. In drei Veröffentlichungen untersuchen wir die Existenz von Kritizität in hochverbundenen Schwellennetzen und finden einen neuen kritischen Punkt mit ähnlicherer Dynamik zu realen Gehirn Netzwerken als zuvor bekannte kritische Punkte in solchen Netzen. Zusätzlich entwickeln wir Algorithmen, die Netzwerke durch Umverdrahtung zu einem kritischen Punkt leiten können. Dies soll Ideen dafür liefern, wie Kritizität in echten Gehirn Netzen aufrecht erhalten wird.

Unser zweites Forschungsgebiet ist die Epidemiologie. Hier können Netzwerke verwendet werden, um Kontakt zwischen Mitgliedern einer Gesellschaft und die Ausbreitung einer infektiösen Krankheit zu modellieren. Wir untersuchen die Wirksamkeit eines rekursiven Kontaktverfolgungsalgorithmus, der versucht, die Ausbreitung einer Krankheit vorherzusehen, und möglicherweise infektiöse Menschen entsprechend unter Quarantäne zu stellen, um eine Krankheit mit einer endlichen asymptomatischen Infektionsrate zu bekämpfen. Wir entwickeln analytische Berechnungen, unterstützt von Simulationen, für die Reduktion der Infektionen einer Krankheit unter Benutzung dieses Algorithmus und finden, dass rekursive Kontaktverfolgung Krankheiten, die durch klassische Kontaktverfolgung von nur direkten Kontakten nicht gestoppt werden könnten, effektiv bekämpfen kann.

Das dritte Gebiet ist die Epigenetik, in dem genetische Netze oft als simple Boolesche Netzwerke modelliert werden. Wir stellen die Hypothese auf, dass genetische Netzwerke robust gegenüber Rauschen sein müssen, da sie mit unzuverlässigen Bausteinen stabil funktionieren müssen, um Leben zu erzeugen, und testen diese Hypothese, indem wir die Robustheit von mehreren echten genetischen Netzwerken mit der von Zufallsnetzen vergleichen. Wir finden höhere Robustheit in den realen Netzwerken verglichen mit zufällig verdrahteten Varianten dieser Netze und führen den Ursprung dieser Robustheit auf eine Kombination der Form der Netzwerkattraktoren und der darunterliegenden Netzwerktopologie zurück.

Um Ideen für amorphe Computer zu wecken, entwickeln wir zuletzt ein kollisionsbasiertes Datenverarbeitungsschema in einem irregulären, zweidimensionalen Schwellennetz. Wir zeigen, dass Interaktionen von "Glidern"von Aktivität, die das Netzwerk durchqueren, genutzt werden können, um einen universellen Satz von Booleschen Gattern zu erstellen, die universelle Datenverarbeitung ermöglichen.

Contents

Preface

This work, as is often the aim of complex systems science, concerns itself with the attempt to simplify a complex system to a model with a, hopefully, minimal set of rules and parameters that still recreates interesting phenomena we observe in the original system, as well as the study of these models. Such a system might, for example, be a congeries of people—of course each with their own opinions, histories, and idiosyncrasies—but who, as a collective, act no differently than a set of soulless actors following a simple set of rules. One often finds that, even with only simple rules, surprisingly complicated phenomena can emerge from such a model.

While other fields often try to consider and faithfully recreate every little metaphorical gear and lever moving such systems in their models to predict the future as accurately as possible, physicists tend to agree with von Neumann's words: "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." In other words, with enough complexity, any model can describe any system to an arbitrary degree of accuracy; however, by reducing a model to the minimum required to recreate reality we can hope to understand how and why phenomena emerge instead of just predicting reality without understanding the underlying mechanisms.

Therefore, in this work, we will consider problems from vastly different fields—neural science, epidemiology, epigenetics, and amorphous computing—and use the tool of network science to develop parsimonious models to describe and study them. It is not relevant or necessary to understand, for example, how proteins interact with each other to turn certain genes in an organism on or off, how a signal is created inside a neuron, or to know the exact course of an infectious disease, as these are microscopic details that do not qualitatively change the resulting behavior of the system. This enables us physicists to be able to study these fields without having to be consummate experts in every single one of them. Naturally, if one's aim is to quantitatively predict anything, the types of models we will use here are insufficient, but our goal is to understand the underlying mechanisms behind the phenomena we observe and to create a foundation on which more quantitative research can be based.

Acknowledgements

I would like to thank Stefan Bornholdt for his excellent supervision and help with all of my projects as well as Gorm Gruner Jensen for helpful discussions. I also want to thank Klaus Bowe for his aid with any technological challenges during my time as a PhD student as well as Keno Krieger and Alex Jochim for their proofreading efforts and helpful comments for this work.

1 Introduction

Various real-life systems can be described as network models, e.g., networks of neurons [1], genetic regulation [2, 3], websites [4], scientific collaborations [5], or social interactions [6, 7]. These models consist of nodes representing elements of the network—such as neurons in a neuronal network, websites in a model of the world wide web, or people in social interaction networks—and links connecting two nodes, which might for example represent synapses connecting neurons, links between websites, or regular contact between people. These links can either be directed, such as in the case of neuronal networks in which a neuron sends a signal via its axons to dendrites of other neurons but not vice versa, or undirected, for example in social interaction network where, if a person has contact with another person, the inverse is also true.

In dynamic networks, nodes have states which are influenced by the states of nodes at the other end of a node's connections and the corresponding connection strengths. For example, in neuronal networks, a neuron would either be quiescent or fire, depending on its membrane potential. A firing neuron would then alter the membrane potentials of its neighboring neurons, potentially causing other neurons to also fire. In epigenetic networks, i.e., networks of gene regulation, a gene's activity may be suppressed or promoted by other genes, and in social networks, the spread of an infectious disease can be simulated with the individual nodes' states representing whether the state is susceptible to infection, infected or already recovered.

Such networks show myriad interesting dynamical properties that can be studied. In neuronal and epidemic network models, one of the most interesting properties of a network is criticality. For neuronal network models, this means that firings of neurons occur in avalanches whose size and duration distributions exhibit power-laws, i.e., avalanches of any arbitrary length can theoretically occur, but longer avalanches are increasingly less likely, resulting in many short avalanches and few long avalanches. In epidemic networks, the critical point represents the boundary between two phases in which, below the critical point, the number of disease carriers decreases with time and, above the critical point, increases with time.

These critical points are therefore of great importance to study because, for neuronal networks, it has been observed that real brain networks operate at or near this critical point [8–31], offering optimal information procession abilities [22, 32–35], while a divergence from the critical point can have devastating effects, such as epileptic seizures [36, 37]. For epidemic networks, naturally one wants to limit the spread of a disease, and it is, therefore, vital to understand what network properties cause sub- or supercriticality, enabling one to manipulate the network to push it into the subcritical regime and thus eliminate a disease over time, or to calculate

the point when a number of recovered agents can protect the population via herd immunity.

In genetic networks, the property of interest we discuss here is the robustness of genetic modulation networks under noise. Naturally, a cell is a very noisy environment making the reliable following of tasks via genetic switches a difficult challenge. Yet, it has been observed that the genetic network of the budding yeast cell cycle can reliably reach and follow a specific attractor, i.e., a repeating sequence of states, despite high noise levels [38, 39].

In this work, we study four separate problems in the realm of complex networks: criticality and its emergence in high-degree neuronal networks, pandemic mitigation via instantaneous recursive contact tracing, the stability of various genetic networks under signal transmission noise, and universal, collision-based computing in a threshold network.

Criticality in neuronal network models: In three papers, we study the critical transition in threshold models as a simple representation of neuronal networks.

The first paper discusses the existence of critical points at high degrees, i.e., a high average number of connections per node, in threshold models. Historically, the critical transition in such models has been studied for low average degrees, at which criticality could be reached by tuning a network's average degree [40–43]; however, recently a new critical point was discovered at high degrees, which is almost independent of average degree and can instead be tuned to by varying the ratio of excitatory to inhibitory connections in a network [44]. In our paper, we find the existence of a second critical point at high average degrees, which appears in networks with a high clustering coefficient, meaning that a node's neighbors are likely to also be neighbors of each other as opposed to random networks in which this is not the case, and nodes that either only have excitatory or only inhibitory outgoing connections. This new critical point has the interesting property of having a low average degree compared to the other critical point at high average degrees, which is closer to real neuronal networks.

In the second paper, we present an algorithm that tunes a threshold model towards criticality using only a node's locally available information. To move the network away from the subcritical quiescent regime, nodes receive incoming excitatory/inhibitory connections if they have exclusively inactive/active for a number of time steps, and to push the network away from the supercritical chaos, connections are removed from nodes that do change their state. This tunes towards the previously mentioned critical point at low average degrees.

The third paper combines the first two papers to create an algorithm that tunes towards the biologically more realistic high-degree critical point. In this paper, a similar algorithm to that in the second paper is applied with two changes: Firstly, nodes are either excitatory or inhibitory instead of connections, meaning that a node either has only excitatory or only inhibitory outgoing connections. A node's identity is not initially determined. Instead, if a node requires a connection from an excitatory node, a connection is either created to an excitatory node or to a node

without any outgoing connections which then becomes excitatory, or equivalently for inhibitory connections. Once a node loses all its outgoing connections, it returns to having no excitatory or inhibitory identity.

Secondly, the network is embedded in two-dimensional space which determines which nodes are connected to each other: Instead of randomly connecting nodes like in the previous paper, a connection is always formed to the closest available node or removed from the farthest connected node. This results in a high clustering coefficient, and overall the algorithm will then tune the network towards criticality at ever-increasing average degrees. Finally, we also present a variant of the model that is closer to real neuronal networks with nodes having a refractory period in which they cannot be active again directly after being active which also tunes towards criticality, indicating the potential biological relevance of our algorithm.

Epidemics with asymptomatic transmission: Sub-critical phase from recursive contact tracing: We study the capability of an app-facilitated, instantaneous, recursive contact tracing algorithm to stop an infectious disease with a nonzero asymptomatic rate in a toy SIR (susceptible, infected, removed) model. In this model, a set percentage of agents uses an app that will inform the people who have had contact with a symptomatically infected person, and, depending on the recursion depth, these people's contacts and their contacts' contacts and so on. These people will then be quarantined so they cannot infect anyone else. Depending on the fraction of people using the app, the recursion degree, the network topology, and the disease's asymptomatic rate, there is a phase transition between regimes in which the number of infected people either increases or decreases. Using both numerical simulations and an analytical approach, we can precisely calculate this critical transition point. We find that the network topology has almost no influence on this transition and that there is a critical value of asymptomatic rate above which even app usage by the entire population cannot push the system into the subcritical phase, with this critical value decreasing for increasing recursion depth.

Robustness to noisy signal transmission delays in genetic networks: Indicated by the remarkable robustness of the budding yeast cell cycle oscillator [39], we hypothesize that robustness to noise is an evolved property that many if not all genetic modulation networks should exhibit. To test this, we study Boolean network representations of various genetic networks provided by the cellcollective.org database. In numerical simulations, we compare the stability under signal transmission noise of genetic networks exhibiting cyclical attractors to randomized versions of these networks. We find that almost all of the networks we tested are more stable than their randomized variants and that this is a combined result of the form of the attractor, i.e., the sequence of states the network goes through, and the underlying topology facilitating this attractor.

Universal computing using localized activity in threshold networks: We present an algorithm that creates Boolean gates in two-dimensional threshold net-

work models. Signals in this model are represented by periodic "gliders", localized clusters of activity that travel through the network, where the presence of a glider indicates a TRUE signal and its absence a FALSE signal. In a Boolean gate, two paths of gliders collide, and, depending on which gliders are present, an output glider is produced or not produced. We show that a universal set of such Boolean gates can be built, enabling the construction of a universal computer by combining these gates.

2 Theoretical background

In this chapter, we will give an overview of the theory our research is based on. Firstly, in section 2.1, we will describe the theoretical foundation all of our models are based on: networks. We will discuss what networks are, their important properties, and how they can be constructed. Further, we will describe the network type we will be using later on, namely Boolean networks. In section 2.2, we will explain the phenomenon of criticality in networks, which four of our publications study in the contexts of neural science and epidemiology, starting from the theory of branching processes which can also be applied to networks, methods of calculating critical points, and indicators of criticality. Also, we will discuss the phenomenon of self-organized criticality, and the universality of different models at the critical point. Finally, we will turn to epidemiological modeling in section 2.3 and describe the effect of network topology on the critical point in such models.

2.1 Networks

Networks, also often referred to as graphs, are systems consisting of a number of nodes, also called vertices, connected by links, also called edges. Examples of networks can be physical objects, immediately recognizable as networks, like networks of neurons, roads, or a power grid, or can be more abstract, such as acquaintanceship between people or interactions between different genes. Links can also be directed: For example in a neuronal network, a neuron may send signals to another neuron but not vice versa. The number of connections a node i has is called the node's degree k_i.

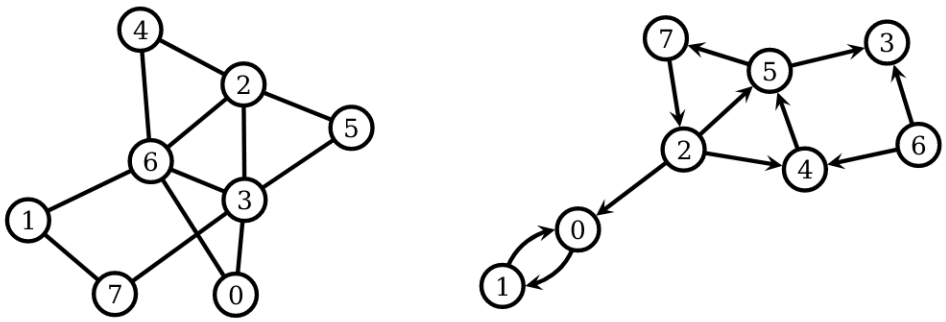


Figure 2.1: Left: An undirected network. Right: A directed network

In directed networks, a node has an in-degree and an out-degree, denoting the total number of its incoming or outgoing connections, respectively. Both nodes and links

may also have additional properties. Nodes may for example belong to certain categories: In a social network, one could distinguish between male and female nodes, e.g., when modeling sexually transmitted diseases [45], or in neuronal networks, one can distinguish between excitatory and inhibitory neurons. More often, nodes have an internal state or property that is influenced by the rest of the network, e.g. agents being infected or not in epidemic models or an intrinsic concentration of promoter proteins in an epigenetic system [39]. In this case, a network becomes a dynamical system whose next state is determined by the network's history. Often, networks are implemented as Markov systems, meaning that only the network's current state determines its next state.

Of course, in addition to dynamics affecting the states of nodes, one can also study dynamics of the network topology. In this case, the dynamics of the network consist of adding, removing, strengthening, or weakening links between nodes and adding or removing nodes. This could for example be the evolution of a scientific co-authorship network in which nodes are added as new authors publish their papers and links are added as authors collaborate on publications [46]. The most common application of these kinds of dynamics is network construction algorithms, such as the Watts-Strogatz model [47] or the Barabási-Albert model [48, 49].

Networks that include both dynamics of node states and dynamics of the underlying topology are called adaptive networks [50]. In such networks, node states are influenced by the underlying topology and the topology is in turn influenced by the nodes' states, creating a feedback loop. These networks can exhibit many interesting properties, most notably for this work self-organized criticality as we will discuss in section 2.2.5.

2.1.1 Network Topology

A network's topology describes the arrangement of all links and nodes in a network. It is most often mathematically described via an adjacency matrix $\mathbf{A}$ in the form of

$$A_{ij} = \begin{cases} 0 & \text{if there is no connection between nodes } i \text{ and } j \\ w_{ij} & \text{otherwise} \end{cases} \tag{2.1}$$

where w_{ij} denotes a connection's strength. In the simplest form $w_{ij} = 1$, which we will assume in this chapter. In this case, the adjacency matrix only describes whether a connection between two nodes exists or not, with no difference in strength between connections. Often, w_{ij} will be either $+1$ or -1 when nodes have promoting or impeding properties, such as in epigenetic [2] networks.

One of the simplest and most intuitive network topologies is a random graph, also called an Erdős-Rényi network [51]. Such a network consists of N nodes and any connection between two nodes i and j is formed with a probability p. This results in a network with an average degree $k = pN$, or commonly $k = p(N-1)$ if no self-connections are allowed, i.e., if a node i cannot have a link to itself.

The Erdős-Rényi network has the advantage over, for example, an ordered graph—

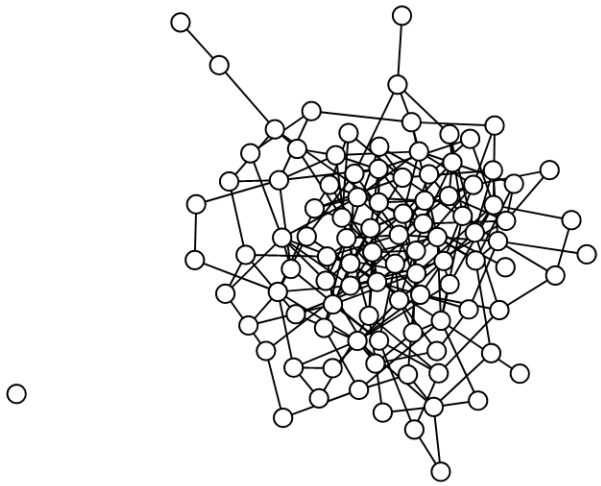


Figure 2.2: An undirected Erdős-Rényi network without self-connections with $N = 100$ nodes and connection probability $p = \frac{4}{99}$, and hence $k \approx 4$.

a grid in which every node is connected to its k nearest neighbors—of short average path lengths. The average path length L is the average distance between any two nodes in the network, where distance is defined as the minimal number of steps along edges one has to take to get from one node to another. For example, in the undirected network in figure 2.1, the distance between nodes 0 and 2 is two because one can get from one to the other over either node 3 or 6. Short average path lengths have been observed in many real-life networks, as has been found first by Milgram [52] and later quantified by Watts and Strogatz for a network of actor collaboration, a power grid, and the neuronal network of C. Elegans [47], and is, therefore, a desirable quality in a network model.

A for many purposes unrealistic property of random graphs is their low clustering coefficient. The local clustering coefficient C_i of a node i in an undirected graph was defined by Watts and Strogatz as the probability that two of the node's neighbors are also neighbors of each other [47],

$$C_i = \begin{cases} \frac{\sum_{j,k} A_{i,j} A_{i,k} A_{j,k}}{k_i(k_i-1)} = \frac{A^3_{ii}}{k_i(k_i-1)} & \text{if } k_i \neq 0 \\ 0 & \text{if } k_i = 0 \end{cases}, \tag{2.2}$$

and a global clustering coefficient $\overline{C}$ can be defined as

$$\overline{C} = \frac{1}{N} \sum_i C_i. \tag{2.3}$$

For directed graphs, a simple definition of the clustering coefficient simply ignores the directions of connections by using $A + A^\top$ instead of A [53]. One must also subtract reciprocal connections in the denominator because if node i has a reciprocal connection to node j, node j would, mathematically, appear as two different

neighbors in the denominator. The local clustering coefficient for a directed graph C_i^{D} then becomes

$$C_i^{\mathrm{D}} = \begin{cases} \frac{(A+A^\top)^3_{ii}}{2\left[k_i(k_i-1)-2k_i^{\mathrm{r}}\right]} & \text{if } 2\left[k_i(k_1-1)-2k_i^{\mathrm{r}}\right] \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{2.4}$$

where k_i^{r} is the number of reciprocal connections node i possesses. Note that in the case of a symmetric adjacency matrix A, this reduces back to the earlier definition of C_i

$$C_i^{\mathrm{D}} = \frac{(2A)^3_{ii}}{2\left[2k_i(2k_i-1)-2k_i\right]} = \frac{A^3_{ii}}{k_i(k_i-1)} = C_i \quad (\text{in case of } k_i(k_i-1) \neq 0)\,. \tag{2.5}$$

Because in this definition lower degree nodes have the same impact on the global clustering coefficient as higher degree nodes, a more commonly used definition of the global clustering coefficient is based on the number of open and closed triplets in the network. A triplet consists of three connected nodes, with either two or three connections between them. The global clustering coefficient is the number of closed triplets—which is equal to three times the number of fully connected triangles in a network as every node in the triangle can be considered the center of a closed triplet—divided by the number of open and closed triplets [54, 55],

$$C = \begin{cases} \frac{\mathrm{Tr}A^3}{\sum_{i\neq j}[A^2]_{i,j}} & \text{if } \sum_{i\neq j}[A^2]_{i,j} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

see figure 2.3.

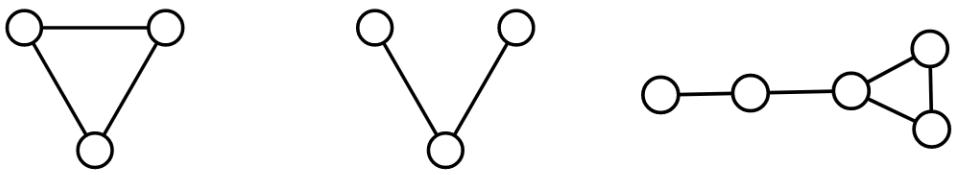


Figure 2.3: Left: A fully connected triangle, equal to three closed triplets. Center: An open triplet. Right: An example graph with one triangle and 6 connected triplets, and therefore a global clustering coefficient $C = 0.5$.

It is clear that many real-life networks will have a higher than random clustering coefficient. In social networks, for example, two friends of a person are more likely to be friends of each other than two random people. In most networks with a spatial component, nodes that are physically closer to each other will also be more likely to be connected than two random nodes. This is notably the case in neuronal networks, which have a significantly higher clustering coefficient than random graphs [56–59].

Probably the most commonly known network model to improve on this shortcom-

ing of random graphs is the Watts Strogatz model [47]. This model compromises between the high clustering coefficients of a network with local connections and the low average path length of a random graph. The Watts Strogatz model starts off as a ring of N nodes in which every node is connected to its k nearest neighbors. Then, connections are randomly rewired with a probability p. This introduces shortcut connections that lower the distances between nodes. Of course, for $p = 0$, the Watts Strogatz model is just a regular graph, and for $p = 1$ it is a random graph. However, for intermediate values of p, networks that combine the properties of these two graph types can be found. There exists a relatively large regime in p-space in which networks exhibit both high global clustering coefficients and low average path lengths, see figure 2.5.

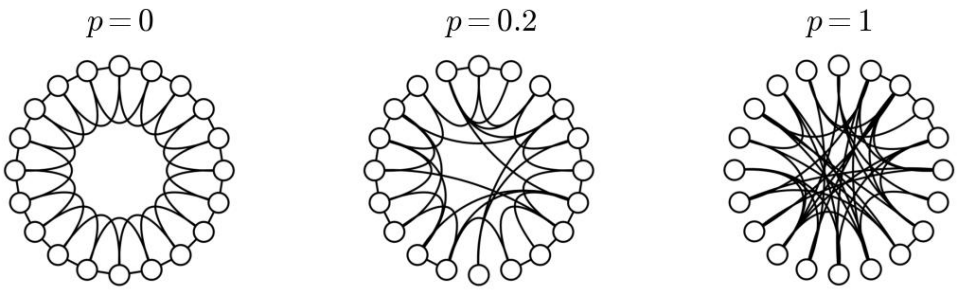


Figure 2.4: Watts Strogatz networks with $N = 20$, $k = 4$, and different values of the rewiring probability p. The leftmost network is a regular graph with high clustering and a high average path length. The rightmost network is a random graph with low clustering and a low average path length. The center network is an intermediate state between the two with both high clustering and a low average path length.

Another very important network property is the degree distribution, that is, the distribution describing the number of nodes $N(k)$ that have exactly k connections. The Erdős-Rényi graph has a binomial degree distribution [60]. However, for many networks found in the real world, such a degree distribution is unrealistic. Instead, real-life networks often show a power-law degree distribution [49]. Such networks are called scale-free networks. Examples include actor collaboration networks, the world wide web [61], electrical power grids, scientific citation networks [62], and—most importantly for this work—neuronal networks [63–70].

The Barabási-Albert model, first described in a more general form by Price [48] and later rediscovered by Barabási and Albert [49], constructs scale-free networks and also provides a possible mechanism for the creation of such network: a combination of growth and preferential attachment. Let us explain this mechanism with the example of a scientific citation network, a network in which nodes represent published research papers and a connection means that one paper cites another. Firstly, the number of nodes in such a network is not constant; new papers are constantly being published. Secondly, the probability that a paper will be cited will grow with

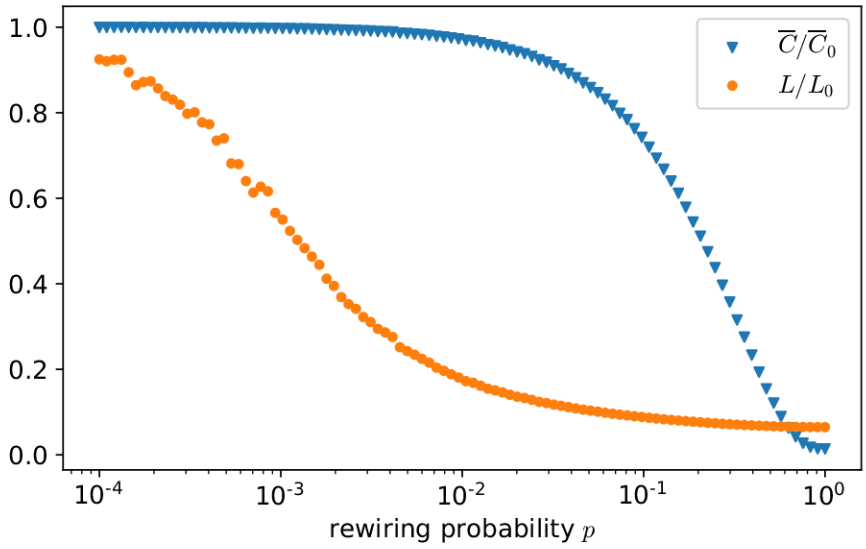


Figure 2.5: Average global clustering coefficients $\overline{C}$ (blue triangles) and path lengths L (orange circles) for Watts Strogatz networks as a function of the rewiring probability p, rescaled by values $\overline{C}_0$ and L_0 at $p = 0$. Every point is the average of 100 network realizations with $N = 1000$ nodes and average degree $k = 10$.

the number of citations that a paper already has because a frequently cited paper will be well known and therefore also be cited more often. Such a probability of forming connections that is proportional to the number of connections that a node already has is called preferential attachment.

In the model, a network initially consists of m_0 connected nodes. A new node is then added to the network in each time step and connected to $m \leq m_0$ other nodes already present in the network. The probability p_i of a node i being chosen for such a connection is proportional to the node's degree,

$$p_i = \frac{k_i}{\sum_j k_j}. \tag{2.7}$$

After a large number t of time steps, i.e., when the number of initial connections is negligible compared to the total number of connections in the network, the total number of connections becomes mt, and the sum of degrees in the denominator in equation 2.7 becomes $\sum_j k_j = 2mt$ as every connection counts towards the total degree of two nodes. The number of connections of a node i in the network therefore grows according to

$$\frac{\partial k_i}{\partial t} = m\frac{k_i}{\sum_j k_j} = m\frac{k_i}{2mt} = \frac{k_i}{2t}, \tag{2.8}$$

which results in

$$k(t) = m(t/t_i)^{1/2}, \tag{2.9}$$

where t_i is the time node i was added to the network. From this, we can calculate the probability that a node i has a degree k_i less than k

$$P\left[k_i(t) < k\right] = P\left(t_i > m^2 \frac{t}{k^2}\right) = 1 - P\left(t_i \leq \frac{m^2 t}{k^2}\right) \tag{2.10}$$

$$= 1 - \frac{m^2 t}{k^2(t + m_0)} \xrightarrow{t\to\infty} 1 - \frac{m^2}{k^2}, \tag{2.11}$$

and the degree distribution is

$$P(k) = \frac{\partial P\left[k_i(t) < k\right]}{\partial t} \approx \frac{2m^2}{k^3} \propto k^{-3} \tag{2.12}$$

This algorithm thus results in a network with a power-law governed degree distribution $P(k) \propto k^{-3}$.

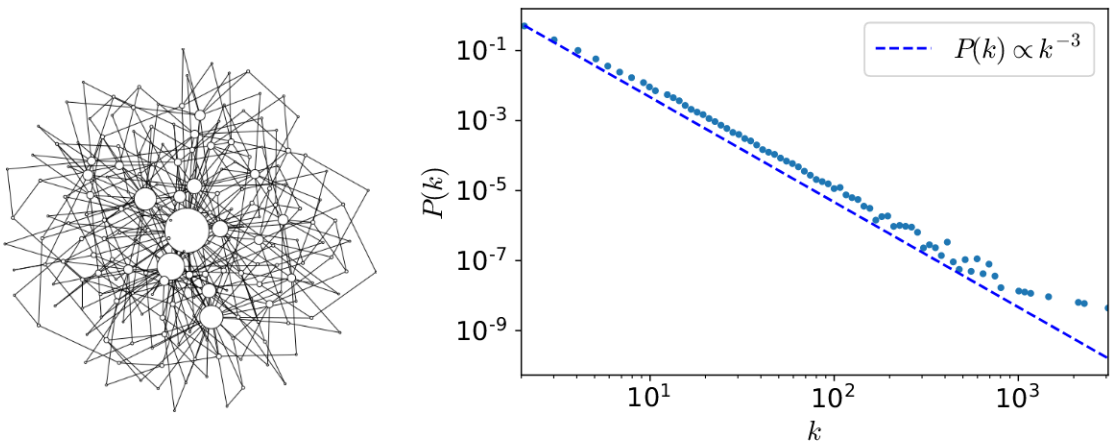


Figure 2.6: Left: A Barabási-Albert network with $N = 200$ nodes and an average degree of $k = 4$. Node sizes are proportional to the nodes' degrees, showing few large, highly connected nodes and many small nodes with few connections. Right: Logarithmically binned degree distribution of a Barabási-Albert network with $N = 10^6$ nodes and an average degree of $k = 4$. The dashed line shows the theoretical slope of $P(k) \propto k^{-3}$.

2.1.2 Boolean Networks

Boolean networks, first introduced by Kauffman [2] are commonly used in epigenetics [2, 3, 71–79], in which a node's state determines whether a gene is being transcribed, or in neural networks [80–82], in which the node state determines whether a neuron is firing or not, and this network archetype is used in most of our publications. In such networks, a node i's state is defined by a binary variable $\sigma_i \in \{0, 1\}$. In their

most basic form, node states of Boolean networks are synchronously updated in discrete time steps, where every node state σ_i follows a Boolean update rule

$$\sigma_i(t+1) = f_i\left(\{\sigma_j(t)\}\right), \tag{2.13}$$

where $f_i(\{\sigma_j(t)\}$ is a Boolean function depending on the node states of all nodes at time t. In neural contexts, the Boolean functions f_i are often chosen to be a simple threshold function

$$f_i\left(\{\sigma_j(t)\}\right) = f\left(S_i(t)\right) = \Theta\left(S_i(t) - h\right) \tag{2.14}$$

$$\text{with } S_i(t) = \sum_j^N A_{ij}(t)\sigma_j(t), \tag{2.15}$$

where $S_i(t)$ is the incoming signal node i receives at time t, Θ is the Heaviside step function, and h is a threshold. In such threshold networks, a node is thus only active if the sum of its inputs, possibly weighted by A_{ij}, is greater than some threshold h.

It is clear that, in the absence of noise, such a network is a deterministic system with a finite number of states, and therefore its dynamics have to eventually lead to a repeating set of states, called an attractor. An attractor consisting of only one stationary state is called a steady state, and one consisting of multiple states is called a limit cycle.

These attractors are of immense importance in epigenetic networks. A steady state may determine the type a cell becomes, therefore enabling cells with the same DNA to fulfill different functions [83, 84], and limit cycles—as well as transients, the states leading up to an attractor—often describe states a cell goes through while fulfilling a function. For example, the fission yeast cell cycle can be described by a simple Boolean model [85], as shown in figure 2.7.

2.2 Criticality in complex networks

In this section, we will discuss the concept of criticality in networks, which, alongside many occurrences of this phenomenon that we do not concern ourselves with, is a vital feature of brain networks and, in epidemiology, separates phases in which an infectious disease can or cannot be stopped, as will be discussed further in the publications.

It has been discovered early on, that the dynamics of Boolean network models—or similar spin glass models in which $\sigma_i \in \{-1, 1\}$—can be sectioned into two different regimes: a quiescent, frozen, or ordered regime in which the network's dynamics are subdued and the response to a change in a node's state dies out quickly, characterized by small limit cycles whose average length scales with a sublinear power of system size N, and a chaotic regime in which such a perturbation is amplified and propagates through the network and the average length of limit cycles grows exponentially with system size [41, 86–90]. In between these two regimes, at the "edge of chaos", is a

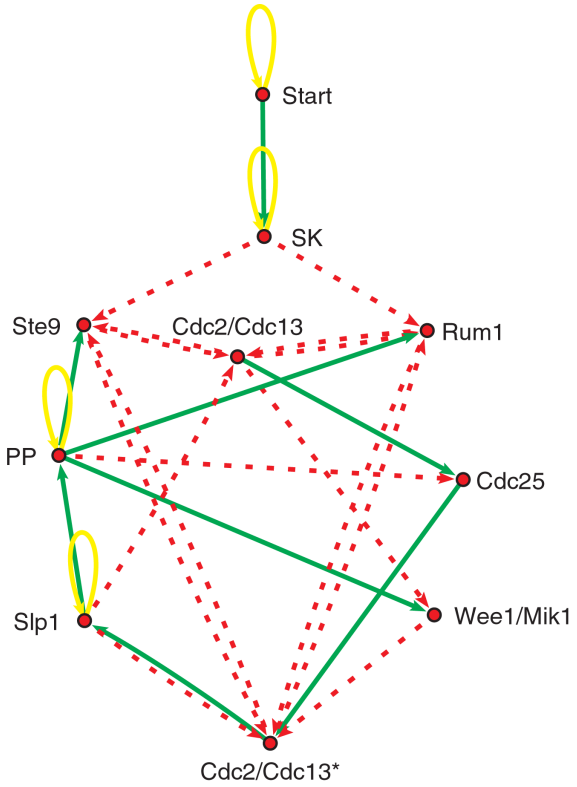


Figure 2.7: Boolean network describing the fission yeast cell cycle. Arrows described interactions between genes. Green arrows are activating connections, red arrows are inhibiting. Yellow arrows are self-inhibiting loops. Taken from [85].

critical point whose properties we will discuss in this section.

2.2.1 Galton-Watson branching processes

Before returning to networks, let us first discuss the theory of branching processes [91, 92], in which we can make analytical calculations regarding criticality. We will later discuss how and why the findings of this section are also relevant for networks.

A branching process is a tree of nodes that can spawn child nodes. The tree starts with an initial 0-th generation; these nodes' children are the first generation, and so on. Let Z_n denote the number of nodes in the n-th generation, and let us for simplicity's sake assume that $Z_0 = 1$. We now also assume that all nodes act independently of each other and that Z_{n+1} only depends on Z_n. The creation of such a tree can thus be described as a Markov process. The number of nodes in successive generations is determined via the probabilities p_k, with $\sum_{k=0}^{\infty} p_k = 1$, of a node having exactly k child nodes. We now introduce the probability generating function

$$f(w) = \sum_{k=0}^{\infty} p_k w^k, \tag{2.16}$$

that encodes the probabilities p_k as coefficients of a power series, where $w \in \mathbb{C}$ is a complex number. If we have a generating function of a probability distribution $f(w)$, the generating function for the sum of k' random variables is simply $[f(w)]^k$, since

$$[f(w)]^k = \left(\sum_{k=0}^{\infty} p_k w^k\right)^{k'} = \sum_{k=0}^{\infty} w^k \prod_{\substack{i=k_1,..,k_{k'} \\ \sum_i = k}} p_i, \tag{2.17}$$

where the product produces all possible combinations of k' probabilities for random variables i that sum up to k.

If we now assume that we know the generating function $f_n(w)$ for the number of nodes Z_n in the n-th generation, the generating function for Z_{n+1} of a tree with $Z_1 = k$ is

$$f_{n+1}\left(w|Z_1 = k\right) = \left[f_n(w)\right]^k, \tag{2.18}$$

because each of the k nodes in the first generation independently produces a tree whose n-th generation corresponds to the entire tree's $(n+1)$-th generation. If we remove the restriction that $Z_1 = k$, the generating function of Z_{n+1} is therefore

$$f_{n+1}(w) = \sum_{k=0}^{\infty} P(Z_1 = k)\left[f_n(w)\right]^k = \sum_{k=0}^{\infty} p_k \left[f_n(w)\right]^k = f\left[f_n(w)\right]. \tag{2.19}$$

We can now see that the generating functions of Z_n are given by

$$f_0(w) = w \tag{2.20}$$
$$f_1(w) = f(w) \tag{2.21}$$
$$f_2(w) = f\left[f(w)\right] \tag{2.22}$$
$$\dots$$
$$f_{n+1}(w) = f\left[f_n(w)\right]. \tag{2.23}$$

The expected value m of a probability distribution p_k is

$$m = \sum_{k=0}^{\infty} p_k k = f'(1), \tag{2.24}$$

and with this, the expected value of Z_n is

$$\langle Z_n \rangle = f_n'(1) = f'\left[f_{n-1}(1)\right] f_{n-1}'(1) = f'(1) f_{n-1}'(1), \tag{2.25}$$
$$\text{with } f_1(1) = \sum_{k=0}^{\infty} p_k 1^k = 1, \tag{2.26}$$
$$\text{and therefore } f_{n-1}(1) = 1. \tag{2.27}$$

If we iterate this result, we receive

$$\langle Z_n \rangle = f'(1) f_{n-1}'(1) = f'(1) f'(1) f_{n-2}(1) = \dots = \left[f'(1)\right]^n = m^n. \tag{2.28}$$

Extinction probability

We see now that, if $m < 1$, the expected value of Z_∞ tends towards zero, in other words, the tree will die out. To show this more rigorously, we consider the extinction probability

$$P_{\text{ext}} = \lim_{n\to\infty} P(Z_n = 0) = \lim_{n\to\infty} f_n(0), \tag{2.29}$$

with the convention that $0^0 = 1$. Since

$$\lim_{n\to\infty} f_{n+1}(0) = \lim_{n\to\infty} f\left[f_n(0)\right] = \lim_{n\to\infty} f_n(0) = P_{\text{ext}}, \tag{2.30}$$

we receive

$$P_{\text{ext}} = f(P_{\text{ext}}). \tag{2.31}$$

For $m \leq 1$, we consider that

$$f(1) = \sum_{k=0}^{\infty} p_k 1^k = 1 \tag{2.32}$$

$$\text{and } f'(w) < f'(1) \Rightarrow f'(w) < 1 \text{ for } 0 \leq w < 1. \tag{2.33}$$

Therefore,

$$f(w) > 1 - f'(1)(1-w) = w \text{ for } 0 \leq w < 1 \tag{2.34}$$

$$\Rightarrow P_{\text{ext}} = 1. \tag{2.35}$$

Hence, all trees with $m \leq 1$ die out with probability 1. For $m > 1$, conversely, $f'(w) > 1$ for $s \lesssim 1$, and $f(0) \geq 0$, see figure 2.8. Therefore, there is a solution for equation (2.31) with $P_{\text{ext}} < 1$, meaning that there is a nonzero probability of trees surviving infinitely long. We have consequently shown that there is a critical transition at $m = 1$.

Size distribution

We now want to calculate the probability distribution of the number of nodes $Z = \sum_{i=0}^{\infty} Z_i$ in a tree with $p_0 \neq 0$ (since otherwise the branching process would continue indefinitely),

$$P_n = P(Z = n), \tag{2.36}$$

for which we will follow [93]. Some mathematical details will be omitted that can be found in the original work. Let us define the generating function of this distribution as

$$F(z) = \sum_{n=0}^{\infty} P_n z^n, \tag{2.37}$$

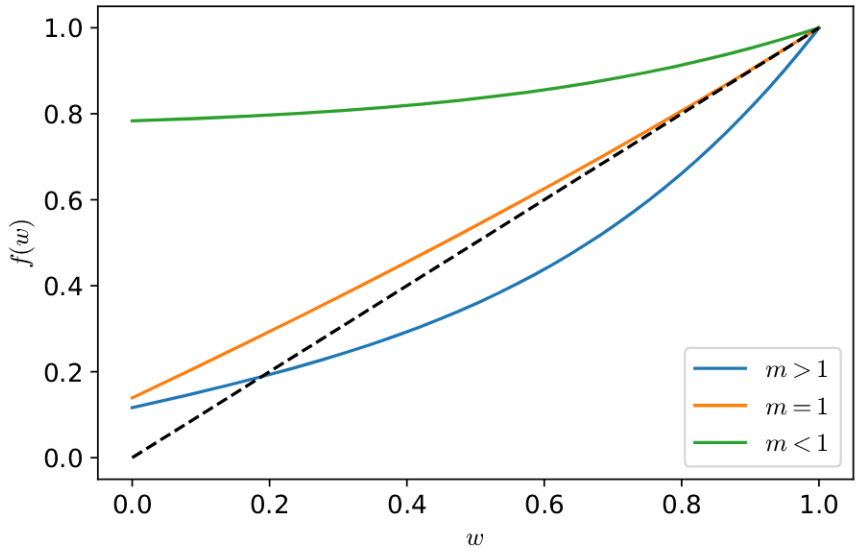


Figure 2.8: Probability generating function $f(w)$ for real values of w and randomly generated example probability distributions p_k with $m > 1$, $m = 1$, and $m < 1$. The dashed line shows the identity function.

with $z \in \mathbb{C}$, and the generating function of the sum of nodes up to the n-th generation

$$F_n(z) = \sum_{i=0}^{\infty} P(\sum_{j=0}^{n} Z_j = i) z^i. \tag{2.38}$$

Imagine that we know the generating function F_n for the number of nodes in a tree with n generations. Now, we know that, given such a generating function, we can calculate the number of descendants of this generating function by applying the generating function $f(w)$, as was done in (2.20–2.23). Therefore, $f\left[F_n(z)\right]$ gives us a generating function for the total number of descendants for all nodes in an existing tree. Effectively, this creates a new tree starting at generation one up to generation $n+1$. For the total number of nodes in this tree, we must now add the one initial node in the zeroth generation, which can be achieved by multiplying with z, hence

$$F_{n+1}(z) = z f\left[F_n(z)\right]. \tag{2.39}$$

For the total number of nodes in a tree, the condition

$$\lim_{n\to\infty} F_n(z) = \lim_{n\to\infty} F_{n+1}(z) = \lim_{n\to\infty} z f\left[F_n(z)\right] \tag{2.40}$$

$$\Rightarrow F(z) = z f\left[F(z)\right] \tag{2.41}$$

must be fulfilled. This can also be written in the form of an implicit equation

$$G(z, w) = z f(w) - w = 0, \tag{2.42}$$

for which $w = F(z)$ is the solution, and the inverse function of $F(w)$ can be constructed

$$F^{-1}(w) = z = \frac{w}{f(w)} \tag{2.43}$$

if $|z| \leq \alpha$ and $|w| \leq \rho$, where α and ρ are the radii of convergence of $F(z)$ and $f(w)$, respectively. Now, let q be the greatest common denominator of all ν with $p_\nu \neq 0$, and let $\Theta = \frac{2\pi}{q}$. Then, w^{kq} is a positive real number if $\arg(w)$ is a multiple of Θ. If this is the case, the equality in

$$|f(z)| = \left| \sum_{k=0}^{\infty} p_{kq} w^{kq} \right| \leq \sum_{k=0}^{\infty} p_{kq} |w|^{kq} = f(|z|) \tag{2.44}$$

holds, and therefore, using (2.43), we obtain

$$|F(z)| = |w| = F\left[z(|w|)\right] = F\left(\frac{|w|}{f(|w|)}\right) = F\left(\frac{|w|}{|f(w)|}\right) = F(|z|), \tag{2.45}$$

which implies that, since $w = F(z)$, and therefore also $\arg(w) = \arg\left[F(z)\right]$, $P_n = 0$ if $n \not\equiv 1 (\mathrm{mod}\ q)$, so that

$$w = |w| \exp\left(\frac{2\pi i}{q} m\right) = P(z) = \sum_{k=0}^{\infty} P_{kq+1} \left(\frac{w}{f(w)}\right)^{kq+1} \tag{2.46}$$

$$= \sum_{k=0}^{\infty} P_{kq+1} \left|\frac{w}{f(w)}\right|^{kq+1} \exp\left(\frac{2\pi i}{q} m(kq+1)\right) \tag{2.47}$$

$$= \exp\left(\frac{2\pi i}{q} m\right) P(|z|) \tag{2.48}$$

$$= \exp\left(\frac{2\pi i}{q} m\right) |P(z)| \tag{2.49}$$

$$= \exp\left(\frac{2\pi i}{q} m\right) |w| = w, \tag{2.50}$$

with $m \in \mathbb{Z}$.

Since all coefficients of the power series $F(z)$ are non-negative, $F(z)$ has a singularity at $F(\alpha)$. Therefore, we can use the implicit function theorem, which states that there is an analytic function that solves the implicit equation (2.42) if

$$\frac{\partial}{\partial w} G(z, w) \neq 0. \tag{2.51}$$

Since $F(z)$ has a singularity at $z = \alpha$, and is thus not analytic, we know that

$$\frac{\partial}{\partial w} G(z, w)\Big|_{z=\alpha, w=a} = \alpha f'(a) - 1 = 0, \tag{2.52}$$

with $a = F(\alpha)$. Now, we define the points

$$z_\nu = \alpha \mathrm{e}^{i\nu\Theta} \tag{2.53}$$
$$w_\nu = a\mathrm{e}^{i\nu\Theta}, \tag{2.54}$$

with $\nu = 0, 1, \ldots, q-1$. One can easily see that

$$F(z\mathrm{e}^{i\nu\Theta}) = \sum_{k=0}^{\infty} P_{kq+1} z^{kq+1} \mathrm{e}^{i\nu\Theta(kq+1)} \tag{2.55}$$
$$= \sum_{k=0}^{\infty} P_{kq+1} z^{kq+1} \mathrm{e}^{i\nu\Theta} \tag{2.56}$$
$$= \mathrm{e}^{i\nu\Theta} F(z), \tag{2.57}$$

and therefore $w_\nu = F(z_\nu)$. If we test the condition of the implicit function theorem (2.51) at these points, we find that

$$\left.\frac{\partial}{\partial w} G(z,w)\right|_{z=z_\nu, w=w_\nu} = z_\nu f'(w_\nu) - 1 \tag{2.58}$$
$$= z_\nu \sum_{k=0}^{\infty} p_{kq} kq w_\nu^{kq-1} - 1 \tag{2.59}$$
$$= \alpha \mathrm{e}^{i\nu\Theta} \sum_{k=0}^{\infty} p_{kq} kq a^{kq-1} \mathrm{e}^{i\nu\Theta(kq-1)} - 1 \tag{2.60}$$
$$= \alpha \mathrm{e}^{i\nu\Theta} \sum_{k=0}^{\infty} p_{kq} kq a^{kq-1} \mathrm{e}^{-i\nu\Theta} - 1 \tag{2.61}$$
$$= \alpha \sum_{k=0}^{\infty} p_{kq} kq a^{kq-1} - 1 \tag{2.62}$$
$$= \alpha f'(a) - 1 = 0, \tag{2.63}$$

(so long as $a < \rho$ so that $G(z,w)$ is analytic here), which means that z_ν are singularities of $F(z)$. In fact, by calculating the derivatives of the inverse function $F^{-1}(w)$,

$$\left.\frac{d}{dw} F^{-1}(w)\right|_{w=w_\nu} = \frac{1 - z_\nu f'(w_\nu)}{f(w_\nu)} = 0 \tag{2.64}$$
$$\left.\frac{d^2}{dw^2} F^{-1}(w)\right|_{w=w_\nu} = -\frac{z_\nu f''(w_\nu)}{f(w_\nu)} = -\frac{\alpha^2 f''(a)}{w_\nu} \neq 0, \tag{2.65}$$

we see that $F(z)$ has first order branch points at z_ν, whereas for $|z| = \alpha$ but $z \neq z_\nu$ we find

$$|G(z,w)| > 1 - \alpha\,|f'(w)| > 1 - \alpha f'(|w|) > 1 - \alpha f'(a) = 0, \tag{2.66}$$

meaning that $F(z)$ is analytic at these points. We can therefore define an analytic

continuation for $F(z)$ as a function of $(z - z_\nu)^{1/2}$, due to the branch point being of first order. If we make branch cuts radiating outwards from z_ν, this function is analytic in the resulting domain D within a circle $|z| \leq \beta$ with $\beta > \alpha$. Now, we use Cauchy's integral formula which gives the n-th derivative of a holomorphic function f at a point a via

$$f^{(n)}(a) = \frac{n!}{2\pi i} \oint_\gamma \frac{f(z)}{(z-a)^{n+1}} dz, \tag{2.67}$$

where a lies in the domain that is by γ (running counterclockwise). Thus

$$P_n = F^{(n)}(0) = \frac{1}{2\pi i} \oint_\Gamma \frac{F(z)}{z^{n+1}} dz, \tag{2.68}$$

where Γ is the counterclockwise oriented boundary of the domain D, as shown in figure 2.9. Since $F(\mathrm{e}^{i\nu\Theta} z) = \mathrm{e}^{i\nu\Theta} F(z)$, see equation (2.57), we can write

$$P_n = \frac{1}{2\pi i} \oint_\Gamma \frac{F(z)}{z^{n+1}} dz = \frac{A}{2\pi i} \int_\gamma \frac{F(z)}{z^{n+1}} dz, \tag{2.69}$$

where γ is the part of Γ that lies in $\frac{-\pi}{q} \leq \arg(z) \leq \frac{\pi}{q}$ and A is

$$A = \sum_{\nu=0}^{q-1} \mathrm{e}^{-i\nu\Theta(n-1)} = \begin{cases} 0 & \text{if } n \not\equiv 1 (\mathrm{mod}\ q) \\ q & \text{if } n \equiv 1 (\mathrm{mod}\ q). \end{cases} \tag{2.70}$$

We now write the analytic continuation of $F(z)$ as

$$F(z) = a + b(z-\alpha)^{1/2} + c(z-\alpha) + (z-\alpha)^{3/2} \zeta(z), \tag{2.71}$$

where $\zeta(z)$ is analytic in D. Clearly, the end result P_n cannot depend on β so the only part of the integral we need to concern ourselves with is the path around the branch cut, which we call γ', see figure 2.9. Since the parts that are linear in $z - \alpha$ do not branch, their integral along γ' is zero. Therefore,

$$\int_\gamma \frac{F(z)}{z^{n+1}} dz = b \int_{\gamma'} \frac{(z-\alpha)^{1/2}}{z^{n+1}} dz + \int_{\gamma'} \frac{(z-\alpha)^{3/2} \zeta(z)}{z^{n+1}} dz + \mathcal{O}(\beta^{-n}). \tag{2.72}$$

When integrating along the branch cut, we choose the branch of the square root function that has $\mathrm{Im}(\sqrt{z}) \geq 0$. When crossing the branch cut, this branch changes the sign of its real part, which means that

$$\int_{\gamma'} \frac{(z-\alpha)^{1/2}}{z^{n+1}} dz = 2 \int_\alpha^\beta \frac{(x-\alpha)^{1/2}}{x^{n+1}} dx. \tag{2.73}$$

To solve this integral, we use the generalized binomial theorem

$$(x+y)^r = \sum_{k=0}^{\infty} \binom{r}{k} x^{r-k} y^k, \tag{2.74}$$

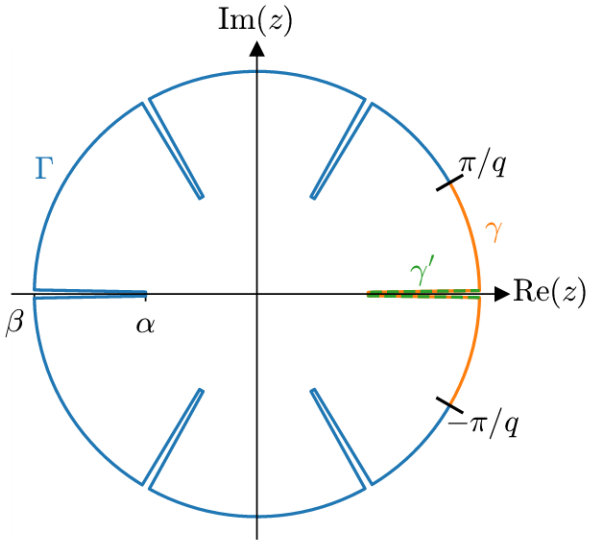


Figure 2.9: Integration paths Γ, γ, and γ' for $q = 6$. Note that Γ includes γ, and γ includes γ'.

which holds for $|x| > |y|$ and, with $r = \frac{1}{2}$, also holds for $|x| = |y|$, i.e., our entire integration path with $x = x$ and $y = \alpha$. The integral then becomes

$$\int_{\gamma'} \frac{(z-\alpha)^{1/2}}{z^{n+1}} dz = 2 \sum_{k=0}^{\infty} \binom{1/2}{k} (-1)^k \alpha^k \int_{\alpha}^{\beta} x^{-(1/2+n+k)} dx \tag{2.75}$$

$$= 2\sqrt{\alpha}\alpha^{-n} \sum_{k=0}^{\infty} \binom{1/2}{k} \frac{(-1)^k}{1/2-n-k} + \mathcal{O}\left(\beta^{-n}\right). \tag{2.76}$$

Finally, we use the identity

$$\sum_{k=0}^{\infty} \binom{1/2}{k} \frac{(-1)^k}{1/2-n-k} = \binom{1/2}{n} (-1)^n \pi \tag{2.77}$$

to arrive at

$$\int_{\gamma'} \frac{(z-\alpha)^{1/2}}{z^{n+1}} dz = 2\pi \binom{1/2}{n} \sqrt{\alpha}\alpha^{-n}(-1)^n + \mathcal{O}\left(\beta^{-n}\right), \tag{2.78}$$

and analogously

$$\int_{\gamma'} \frac{(z-\alpha)^{3/2}\zeta(z)}{z^n+1} dz = \mathcal{O}\left(\left|\int_{\gamma'} \frac{(z-\alpha)^{3/2}}{z^n+1} dz\right|\right) \tag{2.79}$$

$$= \mathcal{O}\left(\left|\alpha^{-n}\binom{3/2}{n}\right|\right) + \mathcal{O}\left(\beta^{-n}\right). \tag{2.80}$$

The constant b can be determined by using

$$w - a = F(z) - F(\alpha) = b(z - \alpha)^{1/2} + \mathcal{O}(z - \alpha) \tag{2.81}$$

and a Taylor approximation of $F^{-1}(z)$, using equation 2.65,

$$z - \alpha = F^{-1}(w) - F^{-1}(a) = -\frac{\alpha^2 f''(a)}{2a}(w - a)^2 + \mathcal{O}\left[(w - a)^3\right], \tag{2.82}$$

which results in

$$b = i\left(\frac{2a}{\alpha^2 f''(a)}\right)^{1/2} \tag{2.83}$$

Putting the results in equations (2.78), (2.80), and (2.83) back into the equation (2.68) for P_n, and approximating the binomial coefficients with

$$\left|\binom{1/2}{n}\right| = (4\pi n^3)^{-1/2} + \mathcal{O}(n^{-5/2}) \quad \text{and} \tag{2.84}$$

$$\left|\binom{3/2}{n}\right| = \mathcal{O}(n^{-5/2}), \tag{2.85}$$

we receive the result

$$P_n = \begin{cases} q\left(\frac{a}{2\pi\alpha f''(a)}\right)^{\frac{1}{2}} \alpha^{-n} n^{-\frac{3}{2}} + \mathcal{O}(\alpha^{-n} n^{-5/2}) & \text{if } n \equiv 1 (\text{mod } q) \\ 0 & \text{if } n \not\equiv 1 (\text{mod } q). \end{cases} \tag{2.86}$$

For all practical purposes, q can be assumed to be one, since $p_1 \neq 0$ implies $q = 1$, and for any branching process with $p_1 = 0$, p_1 can be infinitesimally increased such that $q = 1$. The probabilities P_n go faster than exponentially towards zero with increasing n, except in the case of $\alpha = 1$. In this case, we can calculate that $F(1) = a = 1$, and therefore, from the implicit function theorem (2.63), we know that $f'(1) = 1 = m$. In other words, $\alpha = 1$ is the case at the critical point $m = 1$ (it can more rigorously be shown that $\alpha = 1$ iff $m = 1$ [91]). In short, at the critical point, for large values of n the probabilities of a branching process consisting of n nodes scale as

$$P_n \propto n^{-\frac{3}{2}}. \tag{2.87}$$

Length distribution

It is difficult to provide a general expression for the number of generations before extinction in a branching process [91], but it can be shown via Taylor expanding $f(s)$ that for large n at the critical point, if $f'''(1) < \infty$,

$$P(Z_n > 0) \approx \frac{2}{n f''(1)}, \tag{2.88}$$

which can be used to calculate the probability of a generation having zero members [94]

$$P(Z_n = 0) \approx 1 - \frac{2}{n} \tag{2.89}$$

for a branching process with $p_0 = p_2 = 1/2$. With this, we calculate that the distribution of times to extinction

$$Q_n \propto \frac{d}{dn} P(Z_n = 0) \propto n^{-2} \tag{2.90}$$

also follows a power-law with an exponent of -2.

Connection to networks

It is easy to see that the propagation of a signal through a directed network—so long as clustering can be dismissed, i.e., if it is infinitely improbable that signals taking different paths through the network will ever meet each other—can be described as a branching process. For example, activation spreading in an infinitely large random regular graph, i.e., a random graph in which every node has precisely k outgoing connections, with $k = 2$ and a probability of $p = 0.5$ of activating a node at the end of a connection can be described as a branching process with $p_0 = 0.25$, $p_1 = 0.5$, and $p_2 = 0.25$. It is therefore clear that the distributions of the total number of activated nodes and lengths of activation cascades will follow the power-laws calculated above.

Of particular interest for this work is that the exponents of $-3/2$ and -2 calculated here for the distributions of sizes and lengths of critical branching processes can be found in experimental data of neural activity avalanches, indicating that these avalanches may result from a process belonging to the same universality class as branching processes,[8–11, 95, 96], although power-laws with exponents differing from these have also been observed depending on the cortical region, observed animal, or context, e.g., in vivo versus in vitro, [97, 98].

2.2.2 Annealed approximation

The first analytical calculation of the critical point in networks was done by Derrida and Pomeau for random regular Boolean networks [40]. Consider two network configurations $\Sigma_1(t = 0)$ and $\Sigma_2(t = 0)$, that is specific states for all nodes σ_i, of a network with a distance

$$d\left[\Sigma_1(t = 0), \Sigma_2(t = 0)\right] = n, \tag{2.91}$$

meaning that n nodes have different states in the two configurations. We now want to calculate the probability $P_1(m, n)$ that the distance between these two configurations after one time step will be

$$d\left[\Sigma_1(t = 1), \Sigma_2(t = 1)\right] = m. \tag{2.92}$$

Any nodes whose k inputs are all identical in $\Sigma_1(t=0)$ and $\Sigma_2(t=0)$ will also be identical in $\Sigma_1(t=1)$ and $\Sigma_2(t=2)$. The probability of this applying to N_0 nodes is

$$Q(N_0) = \binom{N}{N_0} \left[\left(\frac{N-n}{N}\right)^k\right]^{N_0} \left[1 - \left(\frac{N-n}{N}\right)^k\right]^{N-N_0}. \tag{2.93}$$

For all other nodes, we assume that the functions governing node states, see equation (2.13) are 1 with probability $p = 1/2$ and 0 with probability $1-p = 1/2$. Therefore, there is a probability of $1/2$ that these nodes will be identical in $\Sigma_1(t=1)$ and $\Sigma_2(t=1)$. Therefore, we can calculate

$$P_1(m,n) = \sum_{N_0=0}^{N} Q(N_0) \left(\frac{1}{2}\right)^{N-N_0} \binom{N-N_0}{m} \tag{2.94}$$

$$= \frac{N!}{m!(N-m)!} \frac{1}{2^N} \left[1 + \left(1 - \frac{n}{N}\right)^k\right]^{N-m} \left[1 - \left(1 - \frac{n}{N}\right)^k\right]^{m}. \tag{2.95}$$

To calculate the probabilities of distances at subsequent time steps $P_t(m,n)$, one would have to consider that states of subsequent time steps are correlated with each other. Here, Derrida and Pomeau introduce the idea of an annealed approximation. Instead of assuming that the functions governing node activity $f_i(\sigma_j(t))$, see equation (2.13), are constant throughout time, we now assume that these functions are randomized in each time step which removes the correlations of node configurations in subsequent time steps. We can therefore write

$$P_t^{\mathrm{a}}(m,n) = \sum_{q_1=0}^{N} \cdots \sum_{q_{t-1}=0}^{N} P_1(m,q_{t-1}) P(q_{t-1},q_{t-2}) \ldots P(q_1,n), \tag{2.96}$$

where P_t^{a} is the annealed approximation for P_t. After introducing continuous variables $x = \frac{n}{N}$ and $y = \frac{m}{N}$ for large N, one finds that the probabilities $P_t(m,n)$ are peaked around values

$$y_t = \frac{1-(1-y_{t-1})^k}{2}, \tag{2.97}$$

$$\text{with } y_1 = \frac{1-(1-x)^k}{2}, \tag{2.98}$$

as shown for P_1 in figure 2.10. For $N \to \infty$, two configurations Σ_1 and Σ_2 with a distance Nx at $t=0$ will have a distance Ny_t after t time steps. The map 2.98 has an attractive fixed point at $x=0$ for $k \leq 2$ which becomes unstable for $k > 2$, where a new fixed point at $y^* > 0$ appears. We see therefore that $k=2$ is the critical point at and below which any perturbation to a system will die out eventually with probability one, just like a branching process will end for $m \leq 1$.

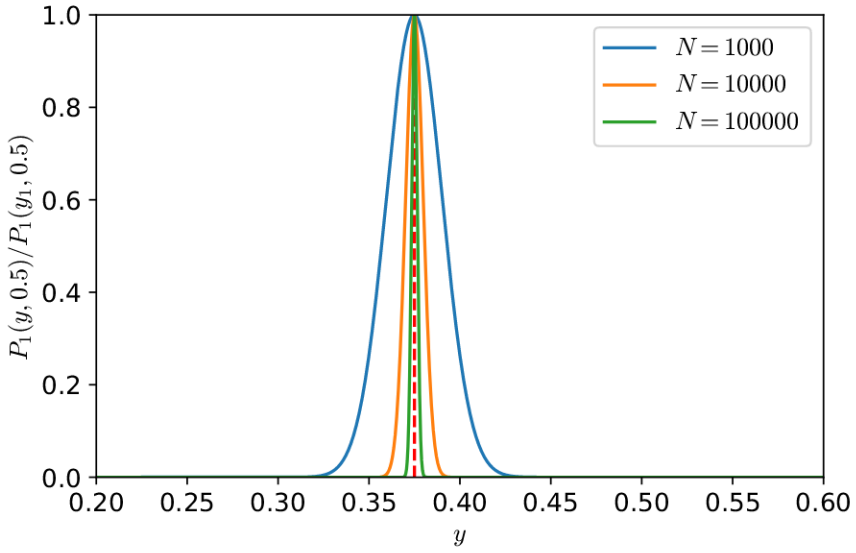


Figure 2.10: Probabilities $P_1(y, x)$, rescaled by $P_1(y_1, x)$ for visibility, for different values of N, $x = 0.5$, and $k = 2$. The red dashed line shows the value $y_1 = \frac{1-(1-x)^k}{2}$.

2.2.3 Sensitivity

A simple, albeit not universally accurate, indicator of the critical point is the sensitivity, also called the branching parameter, λ [99, 100]. The sensitivity measures the average response of a network to the perturbation of one node. To measure the sensitivity of a network at time t, a copy of the network is created and the state of one node in the network is inverted. The sensitivity is then the average distance between the original network and its perturbed copy in the next time step $t + 1$. In other words, it measures how many nodes on average will change their state in response to a perturbation of one node state. If the sensitivity is less than one, a perturbation of one node will on average create a perturbation in the next time step of less than one node, and the perturbation will therefore with high probability die out quickly. This marks the quiescent regime. If, on the other hand, the sensitivity is larger than one, a perturbation will instead grow with time, marking the chaotic regime. Between these two, at a sensitivity of one, the critical point can be found.

The sensitivity is of course closely related to the parameter m in a branching process, as described in section 2.2.1: just like the sensitivity, the parameter m measures the average propagation of a signal and indicates criticality at $m = 1$. From this, we can also see when the sensitivity will fail as an indicator of criticality, namely if the path a signal takes cannot be described as a tree. This will be significant if there are small loops present in a network, e.g., in a clustered network. Of course, most finite networks will have loops, but as long as their influence on signal transmission is not too large, the sensitivity can still be useful to gauge criticality.

2.2.4 Additional criteria of criticality

As mentioned above, the sensitivity is merely an indicator of criticality, but a sensitivity of one does not necessarily imply true criticality. Even the power-laws described above can also be observed in non-critical regimes [101]. A stronger criterion for criticality results from universal scaling theory [102], namely that, if the distribution of avalanche sizes S follows a power-law $\text{Prob}(S) \propto S^{-\tau}$ and the distribution of avalanche durations T follows a power-law $\text{Prob}(T) \propto T^{-\alpha}$, the average avalanche sizes $\langle S \rangle$ as a function of avalanche durations T must also follow a power-law

$$\langle S \rangle \propto T^{\gamma} \tag{2.99}$$

$$\text{with } \gamma = \frac{\alpha - 1}{\tau - 1}. \tag{2.100}$$

Another criterion resulting from universal scaling theory is that the average temporal profiles of avalanches, that is, the number of active nodes in an activity avalanche as a function of time, when properly rescaled, collapse onto each other for all avalanche sizes. This is a result of dynamics at a critical point being scale-invariant, as we will discuss in subsection 2.2.6.

2.2.5 Self-organized Criticality

While the critical point can be reached by tuning a system's parameters, i.e., raising the temperature until a phase transition occurs, many natural systems reliably reach this point without any outside tuning. These systems essentially tune their own parameters to reach criticality; in other words, the critical point is an attractor of the system's dynamics [103]. This phenomenon is called self-organized criticality, or SOC for short, and can be observed in a variety of different systems, such as sandpiles [103], forest fires [104, 105], earthquakes [106], or extinction models [107].

These systems can be viewed as networks in which criticality is encoded in the node states while the connectivity remains constant. It is often, especially in the context of neural networks, of more interest to have criticality emerge from the network's topology instead. Bornholdt and Rohlf [108] introduced a model that evolves a network's topology to reach criticality instead.

This model is a threshold model with binary nodes taking values $\sigma_i = \pm 1$ according to

$$\sigma_i(t) = \text{sgn} f_i(t) \tag{2.101}$$

$$\text{with } f_i(t) = \sum_{j=1}^{N} A_{ij}\sigma_j(t), \tag{2.102}$$

where A_{ij} are the elements of the adjacency matrix $\mathbf{A}$, with $A_{ij} = \pm 1$ if nodes i and j are connected and $A_{ij} = 0$ otherwise. The system is initialized as a random network with random states and the dynamics are run until an attractor is reached. Then a node is randomly selected and a local rewiring rule is applied to it: If the node does

not change its state during the attractor, it receives a new connection from a random node; if it does change its state, it loses a random connection. Additionally, the sign of a random nonzero entry of the adjacency matrix is reversed. The system's state is then randomized again and the process is repeated. For an infinite number of nodes, this algorithm evolves a network towards the critical point at $\langle k \rangle = 2$.

2.2.6 Universality classes

One can find many systems that, although they are governed by different microscopic rules, behave similarly at the critical point, meaning that they have the same critical exponents. Two such systems that share critical exponents are said to belong to the same universality class. Examples include liquid-gas coexistence curves of fluids or magnetic systems [109].

This behavior can be explained by the renormalization group transformation [109, 110]. This is a mathematical method in which a system is coarse-grained, removing degrees of freedom on short scales, and its parameters are rescaled to reproduce the behavior of the original system. This can be described as a flow in the system's parameter space, which in many cases will end in a fixed point. This fixed point maps onto itself under coarse-graining, meaning that it is self-similar, i.e., it has a fractal structure. We recognize this behavior from the power-laws in avalanche duration and size distributions, which are called scale-free, i.e., invariant under coarse-graining, due to their lack of a characteristic scale. Different systems, with different microscopic properties, in a fixed point's vicinity will flow towards that same fixed point under coarse-graining and therefore share the same behavior on long length scales and thus belong to the same universality class.

Notably, it is believed, although still debated, that brain dynamics are part of the same universality class as branching processes [8, 111] which is commonly referred to as the mean-field directed percolation universality class [112]. Critical exponents for various SOC models are shown in table 2.1.

Table 2.1: Critical exponents for avalanche distributions for various SOC models.

		α	τ
Sandpile model [103]	2D	0.42	0.98
	3D	0.90	1.35
Forest fire model [105]		1	2
Earthquake model [106]	2D		1
	3D		1.35
Bak-Sneppen model [107]		1.074 ± 0.004 [113]	1.2911 [114]
Branching process (MF-DP) [91]		2	1.5

2.3 Epidemiological models

In this section, we will explain the basics of epidemiological network models which are utilized by one of our publications. One of the simplest and most popular models for the spread of epidemic diseases is the SIR model, see e.g. [115] for a review. In this model, a population is split into three groups: susceptible (S), infected (I), and removed (R), which may represent states such as recovered and immune or diseased. These models were at first studied in a fully mixed ensemble, in which all members of the population have equal probabilities of meeting and thus infecting others. In this case, the model's evolution can be described using simple nonlinear differential equations [116].

This approach, however, neglects all network effects that might play a role in the spread of a disease. To study these effects, we follow [45] and imagine an infinitely large, random, undirected network with an arbitrary degree distribution. Like in previous sections, the infinite size in this network means that a disease spreading across the network encounters no loops. Every node represents a person who is in one of the three states (susceptible, infected, or removed), and connections represent contact between two people. An infected node i may now infect any susceptible node j that is connected to with probability

$$T_{ij} = 1 - \mathrm{e}^{r_{ij}\tau_i}, \tag{2.103}$$

where r_{ij} is the average rate of disease-causing contacts between the two and τ_i is the time that node i is infective. Assuming that r_{ij} and τ_i are independently and identically distributed random variables pulled from the distributions $P(r)$ and $P(\tau)$, the a priori probability T of infection is simply the average of T_{ij},

$$T = \langle T_{ij} \rangle = 1 - \int_0^\infty dr d\tau P(r) P(\tau) \mathrm{e}^{r\tau}, \tag{2.104}$$

and this average transmissibility T is sufficient to describe the spread of a disease in a network, as demonstrated in [45].

We now once again introduce generating functions

$$f_0(x) = \sum_{k=0}^{\infty} p_k x^k, \tag{2.105}$$

where p_k describes the network's degree distributions, i.e., p_k is the probability of a node having k edges. A crucial property of the spread of a disease is that nodes are not equally likely to be infected; instead, an infecting connection is more likely to lead to a node with a high degree than one with a low degree, which leads to high degree nodes being more likely to become infected. Thus, we define the generating

function for the degree of a node at the end of a random link

$$\frac{\sum_{k=0}^{\infty} k p_k x^k}{\sum_{k=0}^{\infty} k p_k} = x \frac{f_0'(x)}{f_0'(1)}. \tag{2.106}$$

Since we want to know the number of edges that can continue the infection chain, we want to exclude the one edge the node itself was infected by, which we can do by dividing equation (2.106) by x, arriving at

$$f_1(x) = \frac{f_0'(x)}{f_0'(1)} = \frac{1}{\langle k \rangle} f_0'(x), \tag{2.107}$$

where $\langle k \rangle = f_0'(1)$ is the average degree. Further, we also define the generating functions for the number of infective connections originating from a random node $f_0(x;T)$ and for a node at the end of a random connection $f_1(x;T)$,

$$f_0(x;T) = \sum_{l=0}^{\infty} \sum_{k=l}^{\infty} p_k \binom{k}{l} (1-T)^{k-l} (xT)^l \tag{2.108}$$

$$= \sum_{k=0}^{\infty} p_k \sum_{l=0}^{k} \binom{k}{l} (1-T)^{k-l} (xT)^l \tag{2.109}$$

$$= \sum_{k=0}^{\infty} p_k (1 - T + xT)^k \tag{2.110}$$

$$= f_0[1 + (x-1)T], \tag{2.111}$$

where we used the binomial theorem between lines (2.109) and (2.110), and

$$f_1(x;T) = f_1[1 + (x-1)T]. \tag{2.112}$$

To determine the sizes of infected clusters, we define the generating functions

$$H_0(x;T) = \sum_{n=0}^{\infty} P_n(T) x^n \tag{2.113}$$

of the sizes of clusters of infected nodes that a random node is part of and $H_1(x;T)$ for a node at the end of a random edge. Analogously to equation (2.39), we find that

$$F_1(x;T) = x f_1(F_1(x;T);T) \tag{2.114}$$

$$\text{and } F_0(x;T) = x f_0(F_1(x;T);T). \tag{2.115}$$

The average size of an infected cluster is then

$$\langle n \rangle = F_0'(1;T) = 1 + f_0'(1;T)F_1'(1;T) \tag{2.116}$$

$$\text{with } F_1'(1;T) = 1 + f_1(1;T)F_1'(1;T) = \frac{1}{1 - f_1'(1;T)}, \tag{2.117}$$

and thus

$$\langle n \rangle = 1 + \frac{f_0'(1;T)}{1 - f_1'(1;T)} = 1 + \frac{T f_0'(1)}{1 - T f_1'(t)}. \tag{2.118}$$

Note that $\langle n \rangle$ diverges for $T f_1'(1) = 1$. In epidemiology, this quantity is often called the basic reproduction number $R_0 = T f_1'(1)$ and is the equivalent to the parameter m in branching theory. The value $R_0 = 1$ marks the critical point past which a disease can infect a finite fraction of an infinitely large network. We can calculate a critical transmissibility

$$T_c = \frac{1}{f_1'(1)} = \frac{f_0'(1)}{f_0''(1)} = \frac{\langle k \rangle}{\sum_{k=0}^{\infty} k(k-1)p_k}. \tag{2.119}$$

Finding this critical point and developing strategies for reducing the transmissibility to below this point are some of the main interests of epidemiological research. Note therefore in particular, that for a power-law governed degree distribution $p_k \propto k^{-\gamma}$ with $\gamma < 3$, the denominator in equation (2.119) diverges, and the critical transmissibility T_c becomes zero. Therefore, in a scale-free network—so long as $\gamma < 3$, which is the case for most real-world networks—there is no critical transition [117], meaning that any disease is supercritical, which is of particular interest since many real-life networks can approximately be described by power-laws, albeit with cutoffs, as discussed in section 2.1.1.

3 Results

3.1 Criticality in neuronal network models

In high-degree Boolean threshold network models, in addition to a recently discovered critical branch that is almost independent of the network's average degree and exists in a high activity-state, we find a new critical branch with low average activity, see figure 3.1. This critical branch appears in networks with high clustering and in which nodes have either only excitatory or only inhibitory outgoing connections. This critical branch is more realistic for describing neural dynamics due to its low

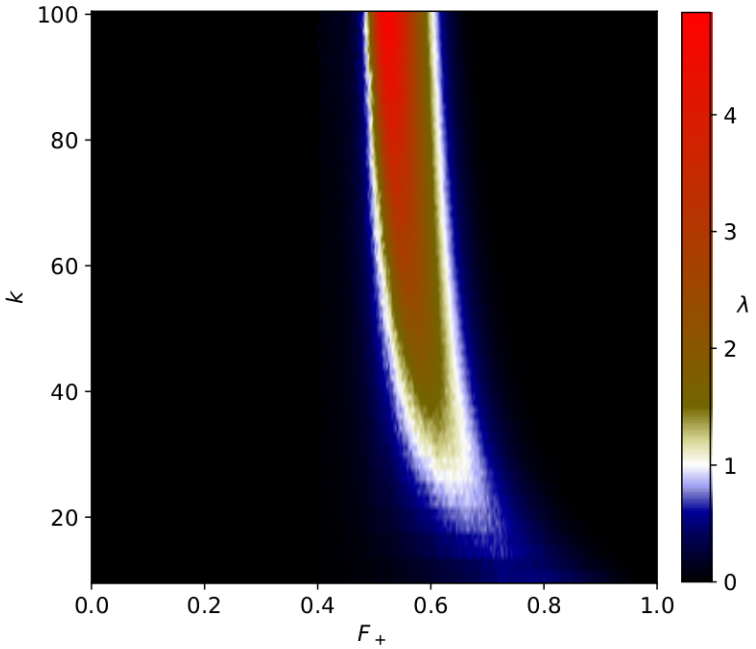


Figure 3.1: Sensitivity λ as a function of the average degree k and the fraction of excitatory connections F_+ in a network in which nodes have either only excitatory or only inhibitory outgoing connections with threshold $h = 2$, clustering coefficient $C = 0.65$. The critical point at $\lambda = 1$ is shown in white. Two separate critical branches can be seen: one at a higher F_+, which is the previously known high-activity critical branch, and one at lower F_+, which is the newly discovered low-activity critical branch.

average activity. Also, both criteria for the existence of this critical branch, high clustering and nodes having either only excitatory or only inhibitory outgoing connections, are present in brains. As the critical point is largely independent of the average degree k, this critical point will still be relevant at the very high average degrees present in brains, unlike the previously discussed critical point at $k = 2$, see section 2.2.2. We also provide an analytical approximation for the fraction of excitatory connections at the high-activity critical branch as a function of the average degree k using annealed approximation.

In another paper, we describe an algorithm to tune a Boolean threshold network towards the critical point at low average degrees using only locally available information. Every W time steps, a node is picked and, if its state has not changed within the last W time steps, it gains an incoming link from a random node. Otherwise, it loses a random incoming connection. Newly created connections are excitatory if the receiving node has been inactive in the previous W time steps or inhibitory if it has been active. Therefore, adding links moves the network away from a quiescent state, and removing links moves the network away from chaos. The result is that this algorithm, regardless of a network's initial topology, quickly tunes a network towards criticality for a large part of the parameter space. Further, we also tested the algorithm for different possible network implementations, namely networks with a fixed number of inhibitory and excitatory nodes and networks with continuous link weights. The algorithm led to criticality regardless of the exact network implementation, indicating the universality of the underlying mechanism. We verified this criticality using avalanche profiles as well as universal scaling theory.

Finally, we combined the two previous results to create an algorithm that tunes to the critical point at high average degrees. For this, we modified the previously explained algorithm such that new connections are always formed to the nearest available node and when an incoming connection of a node is removed, the longest connection is chosen. This ensures that the resulting network will have a high clustering coefficient. Further, nodes are either excitatory or inhibitory, and their identity, instead of being set according to a set fraction of inhibitory nodes as is usually done, is decided by the algorithm. Nodes that have no outgoing connections are undecided, i.e., neither excitatory nor inhibitory. When a node needs to receive for example an inhibitory connection, it can receive this connection from either an inhibitory node or an undecided one. If it receives a connection from an undecided node, that node then becomes inhibitory. The ratio of inhibitory to excitatory nodes is therefore dynamically attained by the algorithm. Since the network now fulfills both criteria for the existence of the low-activity critical branch at high degrees, it should now be able to be tuned to this critical branch, and we do find that the algorithm creates a critical high-degree network for a large part of the parameter space. Once again, we verify the criticality using avalanche profiles as well as universal scaling theory, see figure 3.2. Finally, we introduce a variant of the algorithm operating on a more biologically realistic network model in which nodes can only be active for one time step and are then inactive for a refractory period. This imitates the spiking activity of real neuronal networks. Here too, we find that

the algorithm can create critical networks.

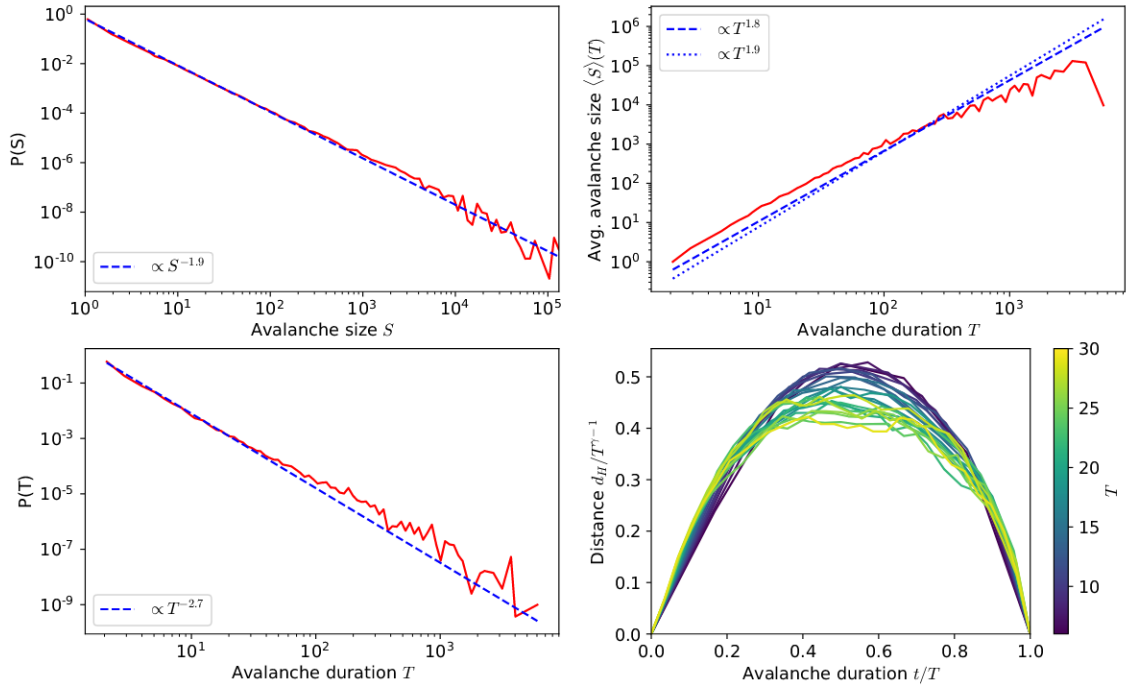


Figure 3.2: Avalanche size and duration distributions (left), average avalanche size as a function of avalanche duration (top right), and avalanche profile collapse (bottom right) for networks at high average degrees created by the algorithm described above. Dashed lines show power-law fits and the dotted line in the top right diagram shows the theoretical power-law resulting from universal scaling theory.

3.2 Epidemics with asymptomatic transmission: Sub-critical phase from recursive contact tracing

We study the effectiveness of recursive contact tracing to combat a disease with a nonzero ratio of asymptomatic infections. We considered a simple toy SIR model with fixed connectivity in which infected nodes (I) can, with a transmission probability, infect neighboring susceptible nodes (S). Infected nodes become infectious for one time step after which they will be considered as removed (R), meaning for example recovered or deceased. Further, there is a probability that infected nodes are asymptomatic instead of symptomatic. We then consider that a fixed percentage of the population can be detected or contacted by an algorithm to reduce infections. We will assume that this is facilitated by these people using an app, although it could also be interpreted as cooperation with authorities or simply the probability of a person reporting their own infection or being reachable within reasonable time.

We then consider a recursive contact tracing algorithm in which the contacts of an infected person who is using the app will be quarantined for the time they might be infectious, as well as, depending on the recursion depth, their contacts and their contacts' contacts, and so on. We calculate the probability that an infectious person is quarantined, which leads to a critical point depending on the transmission probability, symptomatic rate, app adoption rate, degree distribution, and recursion depth, that separates a phase in which the infection will spread through the network and a phase in which it will quickly die out. Firstly, we find that the degree distribution has almost no influence on this critical point, so long as the basic reproduction number and average degree are the same.

Secondly, we calculate the critical values of the app usage rate depending on symptomatic rate and recursion depth, see figure 3.3. Here, it can be seen that, for a given degree distribution and transmissibility, there is a critical value of the symptomatic rate below which infection rates cannot be pushed below the critical point even with app adoption by the entire population; however, this critical symptomatic rate decreases with increasing recursion depth, and any disease with a nonzero symptomatic rate can be stopped using a high enough recursion depth. This shows the value of recursive contact tracing, as only tracing the contacts of infected people would be unable to stop diseases with low symptomatic rates.

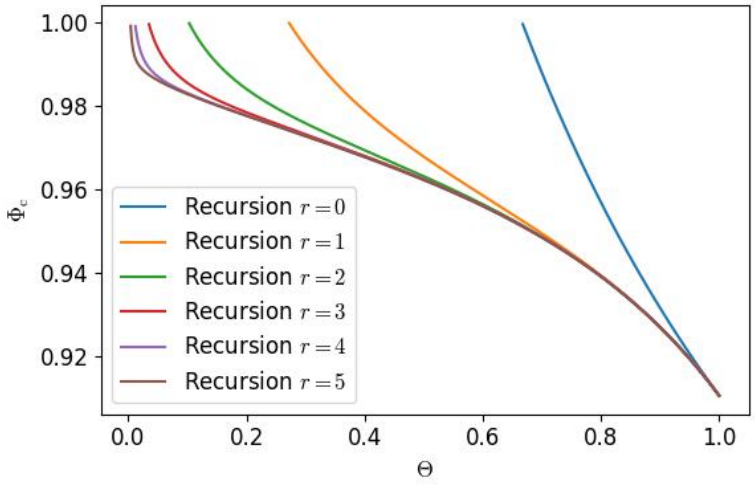


Figure 3.3: Critical app usage rate Φ_c as a function of the symptomatic rate Θ for different recursion depths r for a realistic example degree distribution and transmissibility.

Finally, we test our calculations using simulations, which show excellent agreement with our theoretical results. These simulations also show the fraction of quarantined individuals in the population, which will of course increase with recursion depth. We find that, for high symptomatic rates, the rate of unnecessary quarantining of healthy individuals can grow fast with increasing recursion depths while the critical app usage rate is almost unaffected. In other words, there needs to be a compromise in the chosen recursion depth between infection prevention effective-

ness and unnecessary restrictions on the population. Our research could be used to inform such a decision.

3.3 Robustness to noisy signal transmission delays in genetic networks

Since genetic networks need to function reliably in a noisy environment, we hypothesize that these networks are specifically evolved to have robust dynamics under noise. We test this hypothesis by comparing Boolean genetic network models, collected from the cellcollective.org database, to randomized variants of these networks with equal attractor lengths. Our criterion for robustness is the time a network follows the dynamic attractor it would have without noise in the presence of noise in signal transmission times. We find a significant advantage of the real networks over their randomized variants for most of our studied networks, not much of a difference for some, and an advantage of the randomized variant for one network, as shown in figure 3.4.

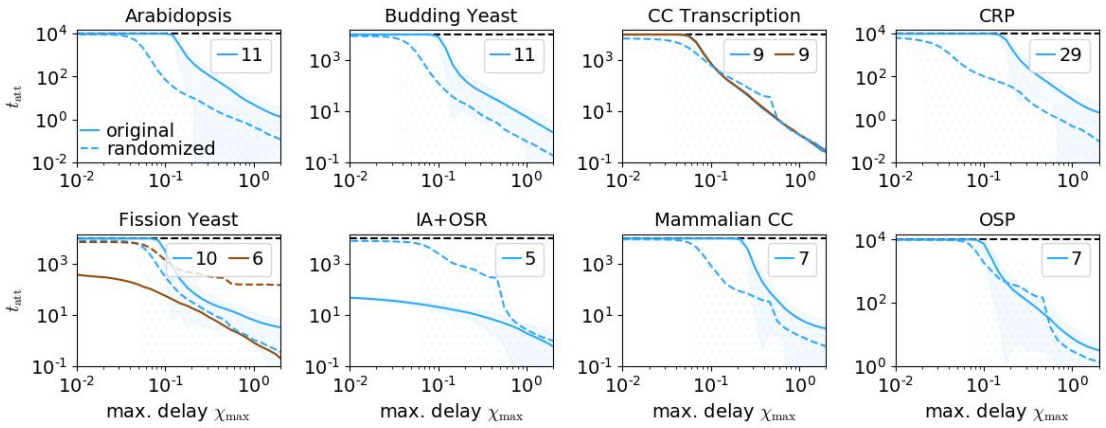


Figure 3.4: Average time t_{att} networks remain in their attractors under noise in signal transmission times, as a function of the maximum delay $\chi_{\max}$ of a signal due to noise. The noise delay $\chi \in [0, \chi_{\max}]$ is added onto a standard signal transmission time of one. Different colors denote different attractors, in order of frequency of appearance, and the dotted, black line shows the maximum simulation runtime. For an explanation of the network names, see 4. Solid lines show the original genetic network, with their standard deviation indicated by the shaded area, while dashed lines show the randomized variants, with their standard deviation indicated by the dotted area.

To further study the origin of this increased robustness, we observe the average Hamming distance of consecutive states in the attractor and find that this is often significantly larger in random networks than in the genetic networks. To eliminate this factor, we studied two more randomized variants of the genetic networks: one

that has the exact same attractor as the original network and one with a random attractor of equal length and similar average Hamming distances between consecutive attractor states. Here, we still find an advantage, albeit a slightly smaller one, for most of the genetic networks over the randomized ones, while the randomized networks with the same attractors as the original networks have a slight advantage over the randomized networks with merely similar Hamming distances. This indicates that the genetic networks' robustness is encoded in the shape of the attractor itself as well as the underlying network topology facilitating it.

3.4 Universal computing using localized activity in threshold networks

We want to create a universal computing scheme in a random, irregular, two-dimensional threshold network using collision-based computing. For this, we create Boolean gates that allow universal computing, in which two gliders, pockets of activity propagating through the network, collide with each other and produce an output signal. This output signal can be encoded in the form of the presence or absence of another glider or persistent activity or inactivity within a target region of the network. We develop a fitness-based rewiring algorithm in which gliders from two input nodes travel towards a target region in which they interact, which then produces the output. The input nodes produce gliders with different prime attractor lengths, which makes the output independent of the activation times of the input nodes, since the prime attractor lengths ensure that any phase difference between the two will occur at some point. A resulting example gate with two outputs, one for each input node, is shown in figure 3.5.

We demonstrate that a universal set of Boolean gates can be created using our algorithm and that these can easily be combined into a universal computer.

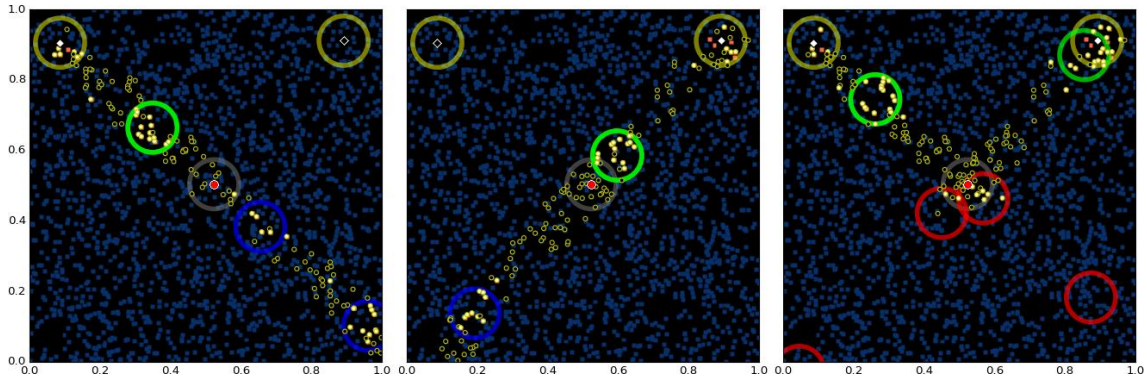


Figure 3.5: Snapshots of a gate where the output (bottom right) for the left node (A) is $A\neg B$, and the output (bottom left) of the right node (B) is $B\neg A$. Both of the outputs are in the form of gliders traveling past the target region. Rhombi denote the input nodes, the red circle denotes the target, and rings denote the intended position of gliders as well as the start (yellow) and target (gray) regions of the signal. Green rings denote gliders that are traveling towards the target point, blue rings denote gliders after the target point for a TRUE output, and red rings show the position a glider would be, had it continued its path, given a FALSE output signal. Nodes have a yellow outline if they are ever active during the simulation, and a yellow body if they are currently active. In other words, yellow nodes in blue rings and no yellow nodes in red rings indicate that the computation is working correctly.

4 Conclusion

In this work, we utilized the framework of network science to formulate complex real-life problems in the form of simple toy models to study their behavior. Our main goal in this formulation was to create minimal models, i.e., with as few parameters and rules as possible, that can still meaningfully recreate phenomena we observe in the real world. Our goal in all of this research is therefore never to make quantitative predictions, as this would require more complicated models, but to qualitatively understand mechanisms that create the effects we see in reality.

In the context of neural science, we studied the critical point that brains seem to tune themselves to using two-dimensional threshold network models. We found a new critical point at high average degrees with properties that can be found in neuronal networks and developed simple algorithms using only locally available information to tune threshold networks towards such a critical point at both low and high degrees. We hope that these findings can help increase our understanding of criticality in real neuronal networks. However, since we used toy models here, further research could be focused on implementing the provided algorithm in more realistic network models.

Next, we turned to the area of epidemiology in which we studied the possibility of stopping infectious diseases with asymptomatic infections using recursive contact tracing. We calculated critical values of compliance in a population necessary to push the spread of a disease into a subcritical regime depending on the disease's rate of asymptomatic infections, recursion depth of contact tracing, and network structure, which we verified using network simulations. These calculations show that it is possible to suppress any disease with an arbitrary asymptomatic rate given high enough compliance of the population if one uses recursive contact tracing, which cannot be achieved with simple one-step contact tracing. Further work could study our quarantining algorithms on more realistic models with real diseases' infection profiles, network structure that considers household structures and similar topological properties of real social networks, and possible delays, should the contact tracing not be facilitated by an app enabling instant tracing and communication.

In the field of epidemiology, we formulated a hypothesis that genetic networks should be evolved to be extraordinarily robust towards noisy environments. To test this, we studied the dynamic attractors of eight epigenetic networks from the cellcollective.org database under random signal transmission delays and found that most of them are significantly more robust under these conditions than randomized variants of each network. Our research shows that this robustness is both a result of the shape of these attractors themselves as well as the underlying network topology. Further research could study more genetic networks, as well as more realistic imple-

mentations of these networks than simple Boolean networks. Also, we believe that our research could be used to evaluate if a Boolean representation of a real genetic network is realistic, as a network that is not robust to noise is likely not realistic for a real organism.

Finally, we use random two-dimensional threshold networks to create a universal computer. This is facilitated by gates in which gliders of activity propagating through the network collide with each other and create an output that can be interpreted as a logical TRUE or FALSE. These gliders and gliders are automatically created by an algorithm altering the network topology. We show that, using this strategy, a universal set of gates can be created and describe how these gates could be combined to create a universal computer. While our algorithm is unlikely to function in the presence of noise, and therefore is likely not actionable in the real world, we hope that, as a proof of concept, it can spark ideas regarding the implementation of amorphous computing in the future.

Bibliography

Selected original works involving the author

Lorenz Baumgarten and Stefan Bornholdt. "Critical excitation-inhibition balance in dense neural networks". In: *Physical Review E* 100.1 (2019), p. 010301.

Stefan Landmann, Lorenz Baumgarten, and Stefan Bornholdt. "Self-organized criticality in neural networks from activity-based rewiring". In: *Physical Review E* 103.3 (2021), p. 032304.

Lorenz Baumgarten and Stefan Bornholdt. "Epidemics with asymptomatic transmission: Subcritical phase from recursive contact tracing". In: *Physical Review E* 104.5 (2021), p. 054310.

Lorenz Baumgarten and Stefan Bornholdt. "A toy model for brain criticality: self-organized excitation/inhibition ratio and the role of network clustering". In: *arXiv preprint arXiv:2202.03330* (2022).

Lorenz Baumgarten and Stefan Bornholdt. "Universal computation using localized limit-cycle attractors in neural networks". In: *International Journal of Unconventional Computing, accepted for publication, arXiv preprint arXiv:2112.05558* (2022).

Lorenz Baumgarten and Stefan Bornholdt. "Robustness to noisy signal transmission delays in genetic networks". To be submitted.

Other original works involving the author

Lorenz Baumgarten and Jan Kierfeld. "Buckling of thermally fluctuating spherical shells: Parameter renormalization and thermally activated barrier crossing". In: *Physical Review E* 97.5 (2018), p. 052801.

Lorenz Baumgarten and Jan Kierfeld. "Shallow shell theory of the buckling energy barrier: From the Pogorelov state to softening and imperfection sensitivity close to the buckling pressure". In: *Physical Review E* 99.2 (2019), p. 022803.

References

[1] Theodore B Achacoso and William S Yamamoto. *AY's Neuroanatomy of C. elegans for Computation*. CRC Press, 1991.

[2] Stuart A Kauffman. "Metabolic stability and epigenesis in randomly constructed genetic nets". In: *Journal of theoretical biology* 22.3 (1969), pp. 437–467.

[3] René Thomas. "Boolean formalization of genetic control circuits". In: *Journal of theoretical biology* 42.3 (1973), pp. 563–585.

[4] Andrei Broder et al. "Graph structure in the web". In: *Computer networks* 33.1-6 (2000), pp. 309–320.

[5] Mark EJ Newman. "The structure of scientific collaboration networks". In: *Proceedings of the national academy of sciences* 98.2 (2001), pp. 404–409.

[6] John Scott. "Social network analysis". In: *Sociology* 22.1 (1988), pp. 109–127.

[7] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994. DOI: 10.1017/CBO9780511815478.

[8] John M. Beggs and Dietmar Plenz. "Neuronal Avalanches in Neocortical Circuits". In: *Journal of Neuroscience* 23.35 (2003), pp. 11167–11177. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.23-35-11167.2003. eprint: https://www.jneurosci.org/content/23/35/11167.full.pdf. URL: https://www.jneurosci.org/content/23/35/11167.

[9] Alberto Mazzoni et al. "On the dynamics of the spontaneous activity in neuronal networks". In: *PloS one* 2.5 (2007).

[10] Elakkat D. Gireesh and Dietmar Plenz. "Neuronal avalanches organize as nested theta- and beta/gamma-oscillations during development of cortical layer 2/3". In: *Proceedings of the National Academy of Sciences* 105.21 (2008), pp. 7576–7581. ISSN: 0027-8424. DOI: 10.1073/pnas.0800537105. eprint: https://www.pnas.org/content/105/21/7576.full.pdf. URL: https://www.pnas.org/content/105/21/7576.

[11] V Pasquale et al. "Self-organization and neuronal avalanches in networks of dissociated cortical neurons". In: *Neuroscience* 153.4 (2008), pp. 1354–1369.

[12] Thomas Petermann et al. "Spontaneous cortical activity in awake monkeys composed of neuronal avalanches". In: *Proceedings of the National Academy of Sciences* 106.37 (2009), pp. 15921–15926.

[13] Christian Tetzlaff et al. "Self-organized criticality in developing neuronal networks". In: *PLoS computational biology* 6.12 (2010).

[14] Shan Yu et al. "Higher-order interactions characterized in cortical activity". In: *Journal of neuroscience* 31.48 (2011), pp. 17514–17526.

[15] Nir Friedman et al. “Universal critical dynamics in high resolution neuronal avalanche data”. In: *Physical review letters* 108.20 (2012), p. 208102.

[16] Enzo Tagliazucchi et al. “Criticality in large-scale brain fMRI dynamics unveiled by a novel point process analysis”. In: *Frontiers in physiology* 3 (2012), p. 15.

[17] Jiangbo Pu et al. “Developing neuronal networks: self-organized criticality predicts the future”. In: *Scientific reports* 3.1 (2013), pp. 1–6.

[18] Viola Priesemann et al. “Spike avalanches in vivo suggest a driven, slightly subcritical brain state”. In: *Frontiers in systems neuroscience* 8 (2014), p. 108.

[19] Gregory Scott et al. “Voltage imaging of waking mouse cortex reveals emergence of critical neuronal dynamics”. In: *Journal of Neuroscience* 34.50 (2014), pp. 16611–16620.

[20] Timothy Bellay et al. “Irregular spiking of pyramidal neurons organizes as scale-invariant neuronal avalanches in the awake state”. In: *Elife* 4 (2015), e07224.

[21] Paolo Massobrio, Valentina Pasquale, and Sergio Martinoia. “Self-organized criticality in cortical assemblies occurs in concurrent scale-free and small-world networks”. In: *Scientific reports* 5 (2015), p. 10578.

[22] Nicholas M Timme et al. “Criticality maximizes complexity in neural tissue”. In: *Frontiers in physiology* 7 (2016), p. 425.

[23] Yuichiro Yada et al. “Development of neural population activity toward self-organized criticality”. In: *Neuroscience* 343 (2017), pp. 55–65.

[24] Shan Yu et al. “Maintained avalanche dynamics during task-induced changes of neuronal activity in nonhuman primates”. In: *Elife* 6 (2017), e27119.

[25] Zhengyu Ma et al. “Critical dynamics are a homeostatic set point of cortical networks in vivo.” In: *bioRxiv* (2018), p. 503243.

[26] Adrián Ponce-Alvarez et al. “Whole-brain neuronal activity displays crackling noise dynamics”. In: *Neuron* 100.6 (2018), pp. 1446–1459.

[27] Mohammad Yaghoubi et al. “Neuronal avalanche dynamics indicates different universality classes in neuronal cultures”. In: *Scientific reports* 8.1 (2018), pp. 1–11.

[28] Zac Bowen et al. “Neuronal avalanches in input and associative layers of auditory cortex”. In: *Frontiers in systems neuroscience* 13 (2019), p. 45.

[29] Antonio J Fontenele et al. “Criticality between cortical states”. In: *Physical review letters* 122.20 (2019), p. 208101.

[30] Stephanie R Miller, Shan Yu, and Dietmar Plenz. “The scale-invariant, temporal profile of neuronal avalanches in relation to cortical γ–oscillations”. In: *Scientific reports* 9.1 (2019), pp. 1–14.

[31] Rodrigo P Rocha et al. "Homeostatic plasticity and emergence of functional networks in a whole-brain model at criticality". In: *Scientific reports* 8.1 (2018), pp. 1–15.

[32] Clayton Haldeman and John M Beggs. "Critical branching captures activity in living neural networks and maximizes the number of metastable states". In: *Physical review letters* 94.5 (2005), p. 058101.

[33] Woodrow L Shew et al. "Information capacity and transmission are maximized in balanced cortical networks with neuronal avalanches". In: *Journal of neuroscience* 31.1 (2011), pp. 55–63.

[34] Xiumin Li, Qing Chen, and Fangzheng Xue. "Biological modelling of a computational spiking neural network with neuronal avalanches". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 375.2096 (2017), p. 20160286.

[35] Woodrow L Shew and Dietmar Plenz. "The functional benefits of criticality in the cortex". In: *The neuroscientist* 19.1 (2013), pp. 88–100.

[36] David Hsu et al. "Simple spontaneously active Hebbian learning model: homeostasis of activity and connectivity, and consequences for learning and epileptogenesis". In: *Physical Review E* 76.4 (2007), p. 041909.

[37] Christian Meisel et al. "Failure of adaptive self-organized criticality during epileptic seizure attacks". In: *PLoS Comput Biol* 8.1 (2012), e1002312.

[38] Fangting Li et al. "The yeast cell-cycle network is robustly designed". In: *Proceedings of the National Academy of Sciences* 101.14 (2004), pp. 4781–4786.

[39] Stefan Braunewell and Stefan Bornholdt. "Superstability of the yeast cell-cycle dynamics: ensuring causality in the presence of biochemical stochasticity". In: *Journal of Theoretical Biology* 245.4 (2007), pp. 638–643.

[40] Bernard Derrida and Yves Pomeau. "Random networks of automata: a simple annealed approximation". In: *EPL (Europhysics Letters)* 1.2 (1986), p. 45.

[41] Karl E Kürten. "Critical phenomena in model neural networks". In: *Physics Letters A* 129.3 (1988), pp. 157–160.

[42] Thimo Rohlf and Stefan Bornholdt. "Criticality in random threshold networks: annealed approximation and beyond". In: *Physica A: Statistical Mechanics and its Applications* 310.1-2 (2002), pp. 245–259.

[43] Agnes Szejka, Tamara Mihaljev, and Barbara Drossel. "The phase diagram of random threshold networks". In: *New Journal of Physics* 10.6 (2008), p. 063009.

[44] Joao Pinheiro Neto et al. "Inhibition as a determinant of activity and criticality in dynamical networks". In: *arXiv preprint arXiv:1712.08816* (2017).

[45] Mark EJ Newman. "Spread of epidemic disease on networks". In: *Physical review E* 66.1 (2002), p. 016128.

[46] Albert-Laszlo Barabâsi et al. “Evolution of the social network of scientific collaborations”. In: *Physica A: Statistical mechanics and its applications* 311.3-4 (2002), pp. 590–614.

[47] Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’networks”. In: *nature* 393.6684 (1998), pp. 440–442.

[48] DJ Price de Solla. “Networks of scientific papers”. In: *Science* 149.3683 (1965), pp. 510–515.

[49] Albert-László Barabási and Réka Albert. “Emergence of scaling in random networks”. In: *science* 286.5439 (1999), pp. 509–512.

[50] Thilo Gross and Bernd Blasius. “Adaptive coevolutionary networks: a review”. In: *Journal of the Royal Society Interface* 5.20 (2008), pp. 259–271.

[51] Paul Erdős and Alfréd Rényi. “On the evolution of random graphs”. In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pp. 17–60.

[52] Stanley Milgram. “The small world problem”. In: *Psychology today* 2.1 (1967), pp. 60–67.

[53] Giorgio Fagiolo. “Clustering in complex directed networks”. In: *Physical Review E* 76.2 (2007), p. 026107.

[54] R. Duncan Luce and Albert D. Perry. “A method of matrix analysis of group structure”. In: *Psychometrika* 14.2 (June 1949), pp. 95–116. ISSN: 1860-0980. DOI: 10.1007/BF02289146. URL: https://doi.org/10.1007/BF02289146.

[55] Mark EJ Newman. “The structure and function of complex networks”. In: *SIAM review* 45.2 (2003), pp. 167–256.

[56] K. E. Stephan et al. “ Computational analysis of functional connectivity between areas of primate cerebral cortex. ” In: *Philos Trans R Soc Lond B Biol Sci.* 355.1393 (Jan. 2000), pp. 111–126.

[57] O. Sporns and J. D. Zwi. “ The small world of the cerebral cortex ”. In: *Neuroinformatics* 2.2 (June 2004), pp. 145–162.

[58] O Sporns, C. J. Honey, and R. Kötter. “ Identification and Classification of Hubs in Brain Networks ”. In: *PLoS One* 2.10 (Oct. 2007), e1049.

[59] D. Smith Bassett and E. Bullmore. “ Small-World Brain Networks ”. In: *The Neuroscientist* 12.6 (Dec. 2006), pp. 512–523.

[60] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. “Random graphs with arbitrary degree distributions and their applications”. In: *Physical review E* 64.2 (2001), p. 026118.

[61] Albert-László Barabási, Réka Albert, and Hawoong Jeong. “Scale-free characteristics of random networks: the topology of the world-wide web”. In: *Physica A: statistical mechanics and its applications* 281.1-4 (2000), pp. 69–77.

[62] Sidney Redner. "How popular is your paper? An empirical study of the citation distribution". In: *The European Physical Journal B-Condensed Matter and Complex Systems* 4.2 (1998), pp. 131–134.

[63] V.M. Eguíluz et al. " Scale-free brain functional networks ". In: *Phys. Rev. Lett.* 9.1 (Jan. 2005), p. 018102.

[64] S. Achard et al. " A Resilient, Low-Frequency, Small-World Human Brain Functional Network with Highly Connected Association Cortical Hubs ". In: *Neurosci.* 26.1 (Jan. 2006), pp. 63–72.

[65] Y. He, Z.J. Chen, and A.C. Evans. " Small-World Anatomical Networks in the Human Brain Revealed by Cortical Thickness from MRI ". In: *Cereb. Cortex* 17.10 (Jan. 2007), pp. 2407–2419.

[66] Y. Iturria-Medina et al. " Studying the human brain anatomical network via diffusion-weighted MRI and Graph Theory ". In: *Neuroimage* 40.3 (Apr. 2008), pp. 1064–1076.

[67] M.P. van den Heuvel et al. " Small-world and scale-free organization of voxel-based resting-state functional connectivity in the human brain ". In: *Neuroimage* 43.3 (Nov. 2008), pp. 528–539.

[68] G. Gong et al. " Mapping Anatomical Connectivity Patterns of Human Cerebral Cortex Using In Vivo Diffusion Tensor Imaging Tractography ". In: *Cereb. Cortex* 19.3 (Mar. 2009), pp. 524–536.

[69] S. Hayasaka and P.J. Laurienti. " Comparison of characteristics between region-and voxel-based network analyses in resting-state fMRI data ". In: *Neuroimage* 50.2 (Apr. 2010), pp. 499–508.

[70] L.R. Varshney et al. " Structural Properties of the Caenorhabditis elegans Neuronal Network ". In: *PLoS Comput. Biol.* 7.2 (Feb. 2011), e1001066.

[71] Stuart A Kauffman et al. *The origins of order: Self-organization and selection in evolution*. Oxford University Press, USA, 1993.

[72] Réka Albert and Hans G Othmer. "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster". In: *Journal of theoretical biology* 223.1 (2003), pp. 1–18.

[73] István Albert et al. "Boolean network simulations for life scientists". In: *Source code for biology and medicine* 3.1 (2008), pp. 1–8.

[74] Stefan Bornholdt. "Boolean network models of cellular regulation: prospects and limitations". In: *Journal of the Royal Society Interface* 5.suppl_1 (2008), S85–S94.

[75] Maria Davidich and Stefan Bornholdt. "The transition from differential equations to Boolean networks: a case study in simplifying a regulatory network model". In: *Journal of Theoretical Biology* 255.3 (2008), pp. 269–277.

[76] Guy Karlebach and Ron Shamir. "Modelling and analysis of gene regulatory networks". In: *Nature reviews Molecular cell biology* 9.10 (2008), pp. 770–780.

[77] Ilya Shmulevich and John D Aitchison. "Deterministic and stochastic models of genetic regulatory networks". In: *Methods in enzymology* 467 (2009), pp. 335–356.

[78] Matthias Rybarsch and Stefan Bornholdt. "Binary threshold networks as a natural null model for biological networks". In: *Physical Review E* 86.2 (2012), p. 026114.

[79] Rui-Sheng Wang, Assieh Saadatpour, and Reka Albert. "Boolean modeling in systems biology: an overview of methodology and applications". In: *Physical biology* 9.5 (2012), p. 055001.

[80] Pauli Rämö et al. "Measures for information propagation in Boolean networks". In: *Physica D: Nonlinear Phenomena* 227.1 (2007), pp. 100–104.

[81] Takuma Tanaka, Takeshi Kaneko, and Toshio Aoyagi. "Recurrent infomax generates cell assemblies, neuronal avalanches, and simple cell-like selectivity". In: *Neural computation* 21.4 (2009), pp. 1038–1067.

[82] Bruno Del Papa, Viola Priesemann, and Jochen Triesch. "Criticality meets learning: Criticality signatures in a self-organizing recurrent neural network". In: *PloS one* 12.5 (2017).

[83] François Jacob and Jacques Monod. "Genetic repression, allosteric inhibition, and cellular differentiation". In: *Cytodifferentiation and macromolecular synthesis*. Vol. 21. Academic Press New York, NY, 1963, pp. 30–64.

[84] Stefan Bornholdt and Stuart Kauffman. "Ensembles, dynamics, and cell types: Revisiting the statistical mechanics perspective on cellular regulation". In: *Journal of theoretical biology* 467 (2019), pp. 15–22.

[85] Maria I Davidich and Stefan Bornholdt. "Boolean network model predicts cell cycle sequence of fission yeast". In: *PloS one* 3.2 (2008), e1672.

[86] L De Arcangelis and D Stauffer. "Period distribution for Kauffman cellular automata". In: *Journal de Physique* 48.11 (1987), pp. 1881–1886.

[87] H Eugene Stanley et al. "Dynamics of spreading phenomena in two-dimensional Ising models". In: *Physical review letters* 59.20 (1987), p. 2326.

[88] J. W. Clark, K. E. Kürten, and J. Rafelski. "Access stability of cyclic modes in quasirandom networks of threshold neurons obeying a deterministic synchronous dynamics". In: *Computer Simulation in Brain Science*. Ed. by Rodney M. J.Editor Cotterill. Cambridge University Press, 1988, pp. 316–344. DOI: 10.1017/CBO9780511983467.022.

[89] K. E. Kürten. "Correspondence between neural threshold networks and Kauffman Boolean cellular automata". In: *Journal of Physics A: Mathematical and General* 21.11 (1988), p. L615.

[90] Ugo Bastolla and Giorgio Parisi. “Closing probabilities in the Kauffman model: An annealed computation”. In: *Physica D: Nonlinear Phenomena* 98.1 (1996), pp. 1–25.

[91] Theodore Edward Harris et al. *The theory of branching processes*. Vol. 6. Springer Berlin, 1963.

[92] Peter Jagers et al. *Branching processes with biological applications*. Wiley, 1975.

[93] Richard Otter. “The multiplicative process”. In: *The Annals of Mathematical Statistics* (1949), pp. 206–224.

[94] Stefano Zapperi, Kent Bækgaard Lauritsen, and H Eugene Stanley. “Self-organized branching processes: mean-field theory for avalanches”. In: *Physical review letters* 75.22 (1995), p. 4071.

[95] Hongdian Yang et al. “Maximal Variability of Phase Synchrony in Cortical Networks with Neuronal Avalanches”. In: *Journal of Neuroscience* 32.3 (2012), pp. 1061–1072. ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.2771-11.2012. eprint: https://www.jneurosci.org/content/32/3/1061.full.pdf. URL: https://www.jneurosci.org/content/32/3/1061.

[96] Oren Shriki et al. “Neuronal avalanches in the resting MEG of the human brain”. In: *Journal of Neuroscience* 33.16 (2013), pp. 7079–7090.

[97] Tiago L Ribeiro et al. “Spike avalanches exhibit universal dynamics across the sleep-wake cycle”. In: *PloS one* 5.11 (2010), e14129.

[98] Woodrow L Shew et al. “Adaptation to sensory input tunes visual cortex to criticality”. In: *Nature Physics* 11.8 (2015), pp. 659–663.

[99] B. Luque and R.V. Solé. “ Phase transitions in random networks: Simple analytic determination of critical points ”. In: *Phys. Rev. E* 55.1 (Jan. 1997), pp. 257–260.

[100] I. Shmulevich and S.A. Kauffman. “ Activities and Sensitivities in Boolean Network Models ”. In: *Phys. Rev. Lett.* 93.4 (July 2004), p. 048701.

[101] Jonathan Touboul and Alain Destexhe. “Power-law statistics and universal scaling in the absence of criticality”. In: *Physical Review E* 95.1 (2017), p. 012413.

[102] James P Sethna, Karin A Dahmen, and Christopher R Myers. “Crackling noise”. In: *Nature* 410.6825 (2001), pp. 242–250.

[103] Per Bak, Chao Tang, and Kurt Wiesenfeld. “Self-organized criticality: An explanation of the 1/f noise”. In: *Physical review letters* 59.4 (1987), p. 381.

[104] Per Bak, Kan Chen, and Chao Tang. “A forest-fire model and some thoughts on turbulence”. In: *Physics letters A* 147.5-6 (1990), pp. 297–300.

[105] Barbara Drossel and Franz Schwabl. “Self-organized critical forest-fire model”. In: *Physical review letters* 69.11 (1992), p. 1629.

[106] Per Bak and Chao Tang. "Earthquakes as a self-organized critical phenomenon". In: *Journal of Geophysical Research: Solid Earth* 94.B11 (1989), pp. 15635–15637.

[107] Per Bak and Kim Sneppen. "Punctuated equilibrium and criticality in a simple model of evolution". In: *Physical review letters* 71.24 (1993), p. 4083.

[108] Stefan Bornholdt and Thimo Rohlf. "Topological evolution of dynamical networks: Global criticality from local dynamics". In: *Physical Review Letters* 84.26 (2000), p. 6114.

[109] James J Binney et al. *The theory of critical phenomena: an introduction to the renormalization group*. Oxford University Press, 1992.

[110] Kenneth G Wilson and John Kogut. "The renormalization group and the ϵ expansion". In: *Physics reports* 12.2 (1974), pp. 75–199.

[111] Tawan TA Carvalho et al. "Subsampled directed-percolation models explain scaling relations experimentally observed in the brain". In: *Frontiers in neural circuits* 14 (2020).

[112] Miguel A Munoz et al. "Avalanche and spreading exponents in systems with absorbing states". In: *Physical Review E* 59.5 (1999), p. 6175.

[113] Peter Grassberger. "The Bak-Sneppen model for punctuated evolution". In: *physics Letters A* 200.3-4 (1995), pp. 277–282.

[114] Bernhard Mikeska. "Monte Carlo renormalization-group approach to the Bak-Sneppen model". In: *Physical Review E* 55.3 (1997), p. 3708.

[115] Herbert W Hethcote. "The mathematics of infectious diseases". In: *SIAM review* 42.4 (2000), pp. 599–653.

[116] Norman TJ Bailey et al. *The mathematical theory of infectious diseases and its applications*. 2nd. Charles Griffin & Company Ltd 5a Crendon Street, High Wycombe, Bucks HP13 6LE., 1975.

[117] Romualdo Pastor-Satorras and Alessandro Vespignani. "Epidemic spreading in scale-free networks". In: *Physical review letters* 86.14 (2001), p. 3200.

Original works

This chapter contains all original works involving the author selected for this thesis, as listed on page 45.

Critical excitation-inhibition balance in dense neural networks

Lorenz Baumgarten and Stefan Bornholdt. "Critical excitation-inhibition balance in dense neural networks". In: *Physical Review E* 100.1 (2019), p. 010301.

Rapid Communications

Critical excitation-inhibition balance in dense neural networks

Lorenz Baumgarten* and Stefan Bornholdt†
Institut für Theoretische Physik, Universität Bremen, 28759 Bremen, Germany

(Received 29 March 2019; published 10 July 2019)

The "edge of chaos" phase transition in artificial neural networks is of renewed interest in light of recent evidence for criticality in brain dynamics. Statistical mechanics traditionally studied this transition with connectivity k as the control parameter and an exactly balanced excitation-inhibition ratio. While critical connectivity has been found to be low in these model systems, typically around $k = 2$, which is unrealistic for natural neural systems, a recent study utilizing the excitation-inhibition ratio as the control parameter found a new, nearly degree independent, critical point when connectivity is large. However, the new phase transition is accompanied by an unnaturally high level of activity in the network. Here we study random neural networks with the additional properties of (i) a high clustering coefficient and (ii) neurons that are solely either excitatory or inhibitory, a prominent property of natural neurons. As a result, we observe an additional critical point for networks with large connectivity, regardless of degree distribution, which exhibits low activity levels that compare well with neuronal brain networks.

DOI: 10.1103/PhysRevE.100.010301

Between the ordered and chaotic regimes of threshold neural networks lies the "edge of chaos," a critical point where the length and size distributions of activity avalanches are governed by characteristic power laws. This dynamical phase transition has been thoroughly studied in random neural networks [1–4], nonsymmetric spin glasses [5], and random Boolean networks [6–10]. Traditionally, threshold neural networks have been studied with precisely balanced excitation and inhibition, usually by randomly assigning activating and inhibiting links with equal probabilities. In these networks, criticality occurs for small average degrees k [1]. However, when allowing the fraction of excitatory links F_+ as a second control parameter of the phase transition, it was recently discovered that there exist two critical lines in the k-F_+-plane: one almost parallel to the F_+ axis at low k and one almost independent of k at some $F_+ > 0.5$ [11]; see Fig. 1.

The relevance of this new critical point becomes apparent in the context of neural brain networks which exhibit a high average degree ($k \approx 10^4$ in human brains [12]) and a characteristic imbalance between excitation and inhibition (20–30% of neurons are inhibitory in monkey brains [13]). There is a large amount of evidence suggesting that the brain operates near a critical point, namely, avalanche sizes and durations governed by power laws [14–19], the possibility of tuning from a subcritical regime through the critical point to a supercritical regime [20], mathematical relations between critical exponents, and collapsible avalanche shapes [15,19,21]. Further, Fraiman *et al.* showed striking similarities between correlation networks extracted from brains and the Ising model at the critical point [22]. The interest in the role of criticality in the brain is illustrated by the large amount of research devoted to criticality in network models inspired by biological networks [23–30].

Unfortunately, the high-degree critical point of Fig. 1 exists in a high-activity regime which is unrealistic for brain networks. We find, however, an additional critical point that persists at low activities, at the left flank of the high sensitivity region, when including additional network properties characteristic of brain networks, thereby providing a more likely network model candidate for describing the processes behind brain criticality.

We use threshold networks consisting of N nodes connected by kN directed edges, whose node states are updated in parallel according to

$$\sigma_i(t+1) = \begin{cases} 1 & \text{if } \sum_{j=1}^{N} w_{ij}\sigma_j(t) > h \\ 0 & \text{if } \sum_{j=1}^{N} w_{ij}\sigma_j(t) \leqslant h, \end{cases} \tag{1}$$

where $\sigma_i(t)$ is the node i's state at time t and w_{ij} is the weight of the connection from node j to node i. The weights w_{ij} can be 0 if there is no connection between nodes i and j, or ± 1 otherwise. The weights of existing connections are chosen randomly with excitatory links $w_{ij} = +1$ chosen with probability F_+. Initial states of the nodes are chosen according to a random initial activity $A_0 = \frac{1}{N}\sum_i \sigma_i$.

A simple quantity that we use to measure criticality is the sensitivity λ [31,32]. Imagine switching one node's state in the current time step; then λ is defined as the average number of nodes whose states will then be different in the next time step from what they would have been otherwise. If sensitivity is smaller or larger than 1, perturbations will quickly die out or dominate the entire network, respectively. Hence, at $\lambda = 1$, the network is in a critical state.

First, in order to establish whether the vertical white line defined by $\lambda = 1$ seen in Fig. 1 indeed is a critical point, we measure the averages of multiple quantities of interest, as well as the average sensitivity for 10^3 time steps after letting the network relax from its initial condition within 2×10^3 time steps (tests show that increasing this time or waiting until an attractor is reached—where possible, attractors cannot be

*lbaumgarten@itp.uni-bremen.de
†bornholdt@itp.uni-bremen.de

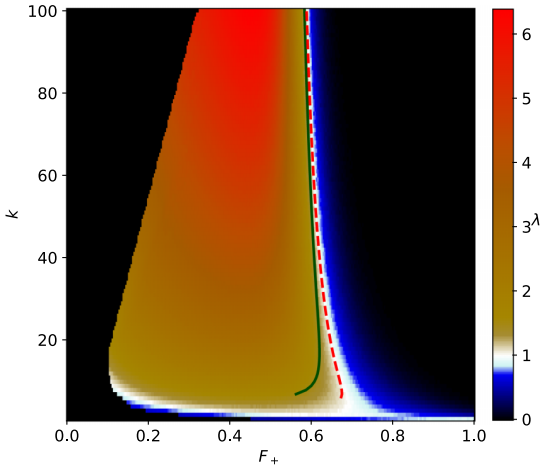


FIG. 1. Sensitivity as a function of fraction of excitatory links F_+ and connectivity k in random neural networks, similar to Fig. 1C in [11], for threshold $h = 0$ and $N = 10^3$ nodes. Lines compare Eq. (3) (green solid line) and the numerical solution of Eq. (2) (red dashed line) with the simulation results. Both lines approximate the simulation's critical line well for large k. Note that the left flank of the sensitive region of the simulation does not exhibit a (white) critical corridor, which is further discussed in the text.

found in a reasonable amount of time for $\lambda \gg 1$—does not change the results) for different values of F_+. Afterwards, we can plot the measured quantities as a function of sensitivity. The measured quantities are the network's activity A, the fraction of nodes which do not change their state within the 10^3 time steps N_S, and the average number of state changes per node and time step F/Nt. This measurement is shown in Fig. 2.

For $\lambda < 1$, essentially all nodes are static (i.e., remaining in one state, either active or inactive) and almost no flips happen, whereas for $\lambda > 1$, the number of static nodes drops and the number of flips increases, so $\lambda = 1$ is a boundary between order and chaos. Also note that the network's activity is very high at the critical point. It seems, therefore, that this critical point cannot underlie a mechanism that defines criticality in the brain, as almost all neurons constantly firing is not realistic.

Further, we measure avalanche sizes and durations at the critical point, as described in the Supplemental Material [33]; see Fig. 3. We observe power laws in both avalanche size and duration distributions.

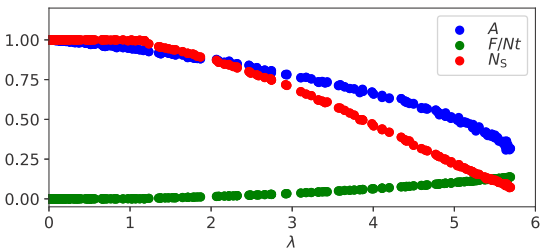


FIG. 2. Activity A, static node fraction N_S, and flips per node and time step F/Nt as a function of the sensitivity λ for $k = 80, N = 10^4$, and $h = 1$.

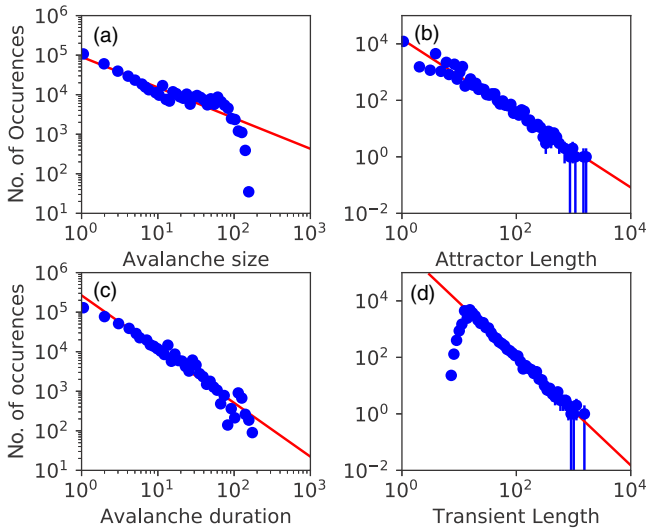


FIG. 3. Distributions of avalanches. (a) Sizes and (c) durations for networks with $F_+ \approx 0.6, k = 80, N = 10^4$, and $h = 1$. The slopes shown in red are (a) -0.8 and (c) -1.4. Also, (b) attractor and (d) transient length distributions for networks with $0.95 \leqslant \lambda \leqslant 1.05$, $F_+ \approx 0.6$, $k = 80$, $N = 4444$, and $h = 1$. The slopes shown in red are (b) -1.3 and (d) -1.9. Logarithmic binning is used for all four figures.

Finally, we measure the attractor and transient lengths, as well as the average sensitivity within the attractor for a number of different network realizations for fixed parameters. We only use parameter and attractor lengths of networks whose average sensitivity λ is within $1 - \delta \leqslant \lambda \leqslant 1 + \delta$ with $\delta = 0.05$. Attractor and transient length distributions are shown in Fig. 3. Both the attractor lengths as well as the right flank of the transient length distributions show clear power laws, as is to be expected for critical networks [34].

All of the above discussed properties lead us to conclude that this is indeed a critical point.

We use Derrida's annealed approximation [6], adopted for threshold networks [2], to estimate the critical F_+ as a function of k, and arrive at the equation

$$\frac{1}{k} = \binom{k}{\frac{k+h+1}{2}} F_+^{\frac{k+h+1}{2}} (1 - F_+)^{\frac{k-h-1}{2}} \frac{k+h+1}{2k}. \tag{2}$$

Under the assumption of large average degree $k \gg h, k \gg 1$, this can be simplified to

$$F_+ = \frac{1}{2}\left[1 + \left\{1 - \left(\frac{2\pi}{k}\right)^{\frac{k}{2}}\right\}^{\frac{1}{2}}\right]. \tag{3}$$

See the Supplemental Material [33] for details. Figure 1 shows a comparison of Eq. (3), as well as the numerical solution of Eq. (2), with our simulation results.

Let us now focus on the the left flank of the central high sensitivity region in Fig. 1. When lowering the value of F_+ from intermediate values towards 0, sensitivity λ seems to suddenly drop to 0 from values larger than 1. In the simulations, this is due to a sudden drop in persistent activity: All activity dies out before the average sensitivity crosses through one. Critical sensitivity here falls into the left (black) region

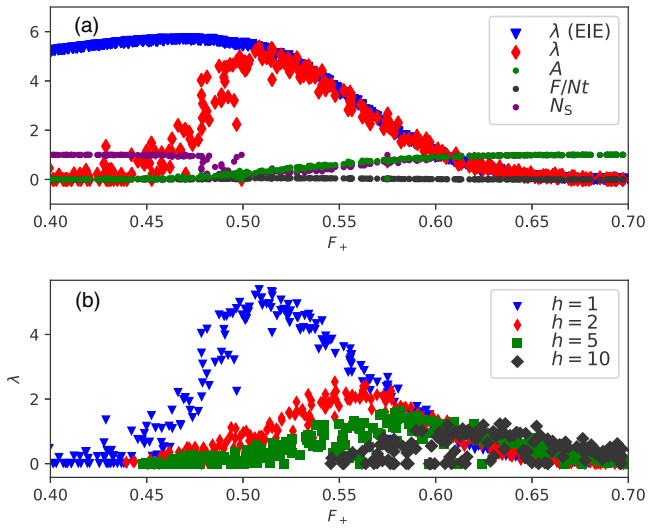


FIG. 4. (a) Activity A, static node fraction N_S, flips per node and time step F/Nt, and sensitivity λ at $h = 1$ and (b) sensitivity λ for different thresholds h as a function of F_+ for $k = 80$, $N = 10^4$, and WS-EIN networks with rewiring probability $\beta = 10^{-2}$ ($C \approx 0.72$). The sensitivity for an equivalent ER network ($C = 0.008$) is also shown in (a) for comparison.

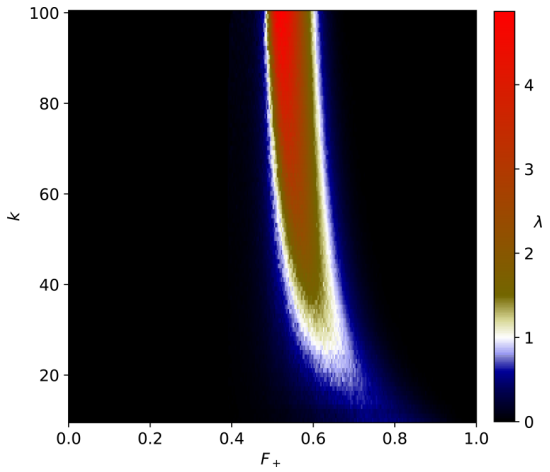


FIG. 5. Sensitivity as a function of fraction of excitatory links F_+ and connectivity k in clustered EIN (Dale) neural networks for threshold $h = 2$, clustering coefficient $C = 0.65$, and $N = 10^4$ nodes.

of entirely inactive networks, whose sensitivity is not shown (as only persisting activity is relevant and, therefore, plotted).

However, as a central observation of our study, we find that networks can be kept from abruptly dying out for low F_+ by introducing two properties to the network: increasing the networks' clustering coefficient C and requiring that nodes have either only excitatory or only inhibitory outgoing edges (Dale's principle). Both of these properties are prominent features of brain networks [35–39]. Note that these properties do not necessarily cause networks to show finite activity for values of F_+ in which the random network has zero activity, but instead that the activity goes continuously towards zero with lowering F_+ instead of abruptly dropping to zero.

We believe the mechanism underlying the left flank's survival to be as follows: If two excitatory nodes which are connected to each other are active, then for high clustering coefficients, they are likely to have shared neighbors and can therefore combine their efforts to also activate these neighbors more easily than in random networks and thereby create islands of surviving activity. The contribution of nodes being either excitatory or inhibitory is likely that if few random nodes are active within a region, this property significantly increases the variance of the relative number of activating signals in that region and therefore increases the probability of areas exhibiting high excitation by random chance.

We also see that the sensitivity in clustered graphs with nodes either fully excitatory or inhibitory closely follows the sensitivity of random graphs for high values of F_+, but then drops off for lower F_+; see Fig. 4. This is likely due to nodes in the center of activity islands receiving many more excitatory connections than necessary for activation. This both lowers the overall activity because these redundant excitatory signals essentially lower the network's total excitation and lowers the sensitivity because only nodes with an input sum near the excitation threshold contribute to it.

Networks with only a high clustering coefficient, without the second property of nodes having either only excitatory or only inhibitory outgoing edges, can also show surviving activity on the left flank for some initial configurations and for exceedingly high clustering coefficients and thresholds, but even then the left flank drops sharply towards zero. In the following, let us denote networks obeying Dale's principle [39], i.e., networks consisting of excitatory neurons and inhibitory neurons as *EIN networks*, as opposed to networks with excitatory-inhibitory edges which we will call *EIE networks*.

Since the network's activity does not abruptly die out on the left flank anymore for clustered EIN networks, a second critical point can be found here, as shown in Fig. 5 and Fig. 4(a). Plotting the sensitivity in the F_+-k plane in Fig. 5, we now see that the left flank indeed exhibits critical sensitivity $\lambda = 1$ (white color). Note that in contrast to the first critical point at the right flank of the sensitive region, this second critical point at the left flank exists in a low-activity state, making it more interesting for real-life applications, such as studying mechanisms underlying brain criticality.

To construct networks with different high clustering coefficients, here we use directed Watts-Strogatz (WS) networks [40,41]. The original WS model consists of a ring of N neurons with periodic boundary conditions in which every neuron is connected to its k nearest neighbors. Then, connections are randomly rewired with rewiring probability β. We use an essentially equivalent implementation without explicit rewiring from [41] in which the probability of a connection from a node i to a node j existing is

$$\begin{aligned} p_{ij} = {} & \beta p_0 + (1-\beta)\Theta[p_0 - D_{ij}/(N/2)] \\ & + \tfrac{1}{2}(1-\beta)\Theta[p_0 + D_{ij}/(N/2)] \\ & \times \Theta[p_0 - (D_{ij} - 0.5)/(N/2)], \end{aligned} \tag{4}$$

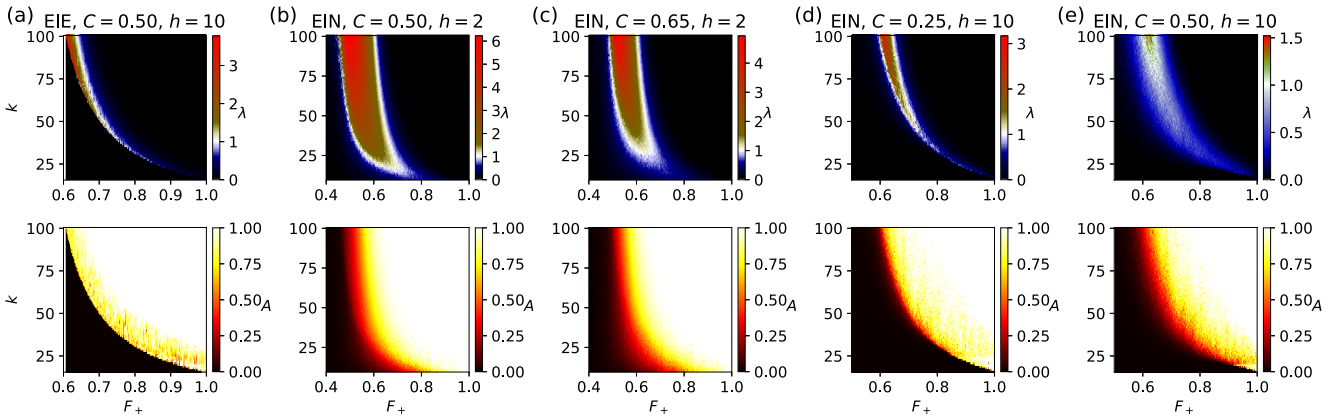


FIG. 6. Sensitivity λ and activity A for different network configurations. White denotes a critical sensitivity. (a) For EIE networks—except for very high clustering coefficients and thresholds—the left flank dies out before reaching the critical point. (b) Switching to an EIN network stabilizes the left flank; however, it still collapses for high average degrees k without a high clustering coefficient. Some white artifacts can be seen because the left flank does not die out within 2×10^3 time steps; it does, however, die out after a larger number of time steps, and therefore no second critical point exists here; see Supplemental Material [33] for more information. (c) A higher clustering coefficient $C = 0.65$ stabilizes the left flank even for higher average degrees k (this case is taken from Fig. 5). (d) With a higher threshold $h = 10$, even a lower clustering coefficient $C = 0.25$ can have a stable left flank. The distance between the critical points shrinks for higher thresholds and both critical points are also moved to higher F_+. (e) For EIN networks, a higher clustering coefficient ($C = 0.5$) lowers the network's average sensitivity, leading this configuration to only barely pass above $\lambda = 1$ between the critical points. From the shape of the left critical line from (b) to (d), it can also be seen that the left critical line is merely a continuation of the horizontal line from Fig. 1 folded upwards.

where $p_0 = k/(N-1)$ and D_{ij} is the distance between nodes i and j on the ring, i.e., $D_{ij} = \min(|i-j|, N-|i-j|)$. The third term has been added to enable uneven values of k. By manipulating the rewiring probability β, we can vary a network's clustering coefficient and average path length. The Watts-Strogatz model's strength is that when varying β, there is a region in which the clustering coefficient is nearly constant while the average path length changes drastically and a second region in which the clustering coefficient changes and the average path length is nearly constant, enabling us to isolate these two parameters' effects.

In our study of clustered EIN networks, we find that the second critical point comes into existence in the region in which the clustering coefficient changes, while it is unaffected by changes within the region in which the clustering coefficient is constant. Therefore, a high clustering coefficient is sufficient to enable the second critical point's existence.

The influence of thresholds and clustering coefficients, as well as the difference between EIE and EIN networks is shown in Figs. 4(b) and 6.

So far, our networks had degree distributions centered around an average value; however, random or Watts-Strogatz models rarely describe real-life networks. Scale-free or similar networks with a broad degree distribution are significantly more abundant in nature. In fact, for neuronal networks, cumulative degree distributions ranging from power laws [42–44] over exponentially truncated power laws [45–49] to exponential laws [50–53] have been found, with the observation that distributions following exponentially truncated power laws increasingly resemble true power laws for measurements on finer scales [45].

In analogy to the brain, we focus on EIN networks with a broad link distribution. For generating the topology, we require an algorithm that (1) can produce a scale-free graph in which low-degree nodes can exist, (2) can initialize large networks fast, (3) can produce networks with variable clustering coefficient, as we have already seen that this can have a large impact on criticality, and, if possible, (4) can also produce other degree distributions similar to scale-free graphs.

For this purpose, we adapt the algorithm described by Lo *et al.* [54], a particularly efficient implementation of preferential attachment [55], to fit our criteria.

In our algorithm, we start with a single node and iteratively add a connection between two nodes every two time steps, so that the sum of in and out degrees in the network increases by one per time step. The origins and targets of these added nodes are chosen by preferential attachment, meaning that the probability of a node being chosen is proportional to the sum of its in and out degree plus an offset δ, which ensures that the probability of previously unconnected nodes receiving connections is nonzero. Further, every m time steps, a new node is added to the network. One significant difference between our algorithm and other algorithms creating scale-free graphs is that the newly added edges need not connect to the newly added node, but can instead connect any two nodes in the system, allowing low-degree nodes to exist in the final network.

This process is repeated multiple times and the connections of every initialization are added together into one network until the desired average degree is reached. Finally, we add i random incoming and outgoing connections to every node, where i is the first integer with $i > h$, so that all nodes have the chance of being activated. For a detailed description of this algorithm, see the Supplemental Material [33].

The two parameters δ and m control whether the resulting degree distribution is scale free or an exponentially truncated power law and also the clustering coefficient. In general,

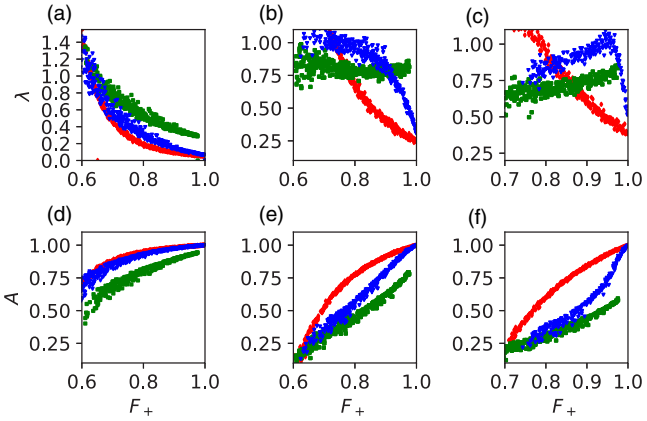


FIG. 7. (a)–(c) Sensitivity λ and (d)–(f) activity A as a function of F_+ for an exponentially truncated power law network with low clustering coefficient ($\delta = 40, m = 2$) (red) and a scale-free network with high clustering coefficient, where the largest node is excitatory ($\delta = 1, m = 10$) (green) or inhibitory (blue) at $k = 40$, $N = 10^4$, and (a), (d) $h = 1$, (b), (e) $h = 7$, (c), (f) $h = 10$. The sensitivity and activity for the highly clustered network were measured as the average within the network's attractor.

lower δ and higher m lead to scale-free distributions with high clustering, whereas high δ and low m lead to low clustering truncated power law distributions.

Studying the dynamics of EIN networks with such a topology, we find that for scale-free graphs, the right critical point still exists (see Fig. 7), and that the sensitivity λ splits into two paths on the right flank and is therefore no longer solely dependent on F_+. The two different paths are dependent on whether the network's largest node is excitatory or inhibitory (in our algorithm, there is a clear hierarchy between nodes, dictated by when they were introduced to the network, and therefore the first node is always clearly larger than the rest, so no multiple nodes are competing for the spot of largest node). Similarly to the existence of the left flank in WS networks, this split in the sensitivity is amplified by high clustering coefficients and thresholds.

Figure 7 also shows the existence of the left flank's second critical point for high clustering coefficients C and thresholds h; see Fig. 7. For low clustering coefficients, the left flank still dies out. High clustering coefficients and thresholds lower the sensitivity curve's slope, so that for certain parameters, the sensitivity, and therefore criticality, is almost constant over a wide area of F_+; see Fig. 7(b).

To summarize, in threshold neural networks, a phase transition between a chaotic and a quiescent regime has been found for highly clustered networks with exclusively excitatory-inhibitory nodes. This critical point exhibits a persisting, yet low level of average activity (which in unclustered networks would die out). Besides the requirement of a certain level of clustering, it is robust both for random as well as broad (scale-free) degree distributions.

This new critical point is of particular interest to neuroscience because it is relatively independent of the degree k and may, therefore, occur at the large average degree present in brains. Furthermore, the main prerequisites for this critical point's existence are present in the brain: a highly clustered architecture and nodes being either exclusively excitatory or inhibitory (Dale's principle).

It can only be speculated what role criticality may play in nature. It has been discussed that it could optimize a network's information processing capabilities. Yet also, dynamical phase transitions are a simple means that physics provides, allowing a complex system to tune to an intermediate activity regime with great ease.

Last, but not least, research has shown that the balance between excitation and inhibition in the brain, which needs to be a specific value for a network to be critical in our model, is vital for a functioning brain [56–60] and that disturbing this balance can negatively impact information processing [61]. Interestingly, the ratio of excitatory and inhibitory neurons in brain networks is observed to be almost constant throughout an organism's development, and feedback algorithms that regulate this ratio are currently discussed [62]. This supports our hypothesis that the critical point described in this Rapid Communication, resulting from the statistical mechanics of a dynamical phase transition, may provide a natural target value for mechanisms that regulate the excitation-inhibition balance in the brain.

[1] K. Kürten, Phys. Lett. A **129**, 157 (1988).
[2] S. Bornholdt and T. Rohlf, Physica A **310**, 245 (2002).
[3] A. Szejka, T. Mihaljev, and B. Drossel, New J. Phys. **10**, 063009 (2008).
[4] T. Rohlf, Phys. Rev. E **78**, 066118 (2008).
[5] B. Derrida, J. Phys. A **20**, L721 (1987).
[6] B. Derrida and Y. Pomeau, Europhys. Lett. **1**, 45 (1986).
[7] K. E. Kürten, J. Phys. A **21**, L615 (1988).
[8] M. Aldana, S. Coppersmith, and L. Kadanoff, in *Perspectives and Problems in Nolinear Science*, edited by E. Kaplan, J. Marsden, and K. Sreenivasan (Springer, New York, 2003).
[9] B. Drossel, in *Reviews of Nonlinear Dynamics and Complexity*, Vol. 1, edited by H. Schuster (Wiley-VCH, Weinheim, 2008).
[10] S. Bornholdt and S. Kauffman, J. Theor. Biol. **467**, 15 (2019).
[11] J. Neto, M. de Aguiar, J. Brum, and S. Bornholdt, arXiv:1712.08816.
[12] P. R. Huttenlocher, Brain Res. **163**, 195 (1979).
[13] S. Hendry, H. D. Schwark, E. Jones, and J. Yan, J. Neurosci. **7**, 1503 (1987).
[14] J. Beggs and D. Plenz, J. Neurosci. **23**, 11167 (2003).
[15] N. Friedman, S. Ito, B. A. W. Brinkman, M. Shimono, R. E. Lee DeVille, K. A. Dahmen, J. M. Beggs, and T. C. Butler, Phys. Rev. Lett. **108**, 208102 (2012).
[16] V. Priesemann, M. Wibral, M. Valderrama, R. Pröpper, M. Quyen, T. Geisel, J. Triesch, D. Nikolić, and M. Munk, Front. Syst. Neurosci. **8**, 108 (2014).
[17] N. Timme, N. Marshall, N. Bennett, M. Ripp, E. Lautzenhiser, and J. Beggs, Front. Physiol. **7**, 425 (2016).

[18] M. Yaghoubi, T. de Graaf, J. Orlandi, F. Girotto, M. Colicos, and J. Davidsen, Sci. Rep. **8**, 3417 (2018).
[19] A. J. Fontenele, N. A. P. de Vasconcelos, T. Feliciano, L. A. A. Aguiar, C. Soares-Cunha, B. Coimbra, L. Dalla Porta, S. Ribeiro, A. J. Rodrigues, N. Sousa, P. V. Carelli, and M. Copelli, Phys. Rev. Lett. **122**, 208101 (2019).
[20] C. Haldeman and J. M. Beggs, Phys. Rev. Lett. **94**, 058101 (2005).
[21] A. Shaukat and J. Thivierge, Front. Comput. Neurosci. **10**, 29 (2016).
[22] D. Fraiman, P. Balenzuela, J. Foss, and D. R. Chialvo, Phys. Rev. E **79**, 061922 (2009).
[23] T. Gross and B. Blasius, J. R. Soc. Interface **5**, 259 (2007).
[24] X. Li, Q. Chen, and F. Xue, Phil. Trans. R. Soc. A **375**, 20160286 (2017).
[25] W. Clawson, N. Wright, R. Wessel, and W. Shew, PLoS Comput. Biol. **13**, e1005574 (2017).
[26] L. Brochini, A. de Andrade Costa, M. Abadi, A. Roque, J. Stolfi, and O. Kinouchi, Sci. Rep. **6**, 35831 (2016).
[27] S. Gautam, T. Hoang, K. McClanahan, S. Grady, and W. Shew, PLoS Comput. Biol. **11**, e1004576 (2015).
[28] O. Shriki and D. Yellin, PLoS Comput. Biol. **12**, e1004698 (2016).
[29] B. V. Rodriguez, A. Avena-Koenigsberger, O. Sporns, A. Griffa, P. Hagmann, and H. Larralde, Sci. Rep. **7**, 13020 (2017).
[30] M. Ferraz, H. Melo-Silva, and A. Kihara, PLoS One **12**, e0184367 (2017).
[31] B. Luque and R. V. Solé, Phys. Rev. E **55**, 257 (1997).
[32] I. Shmulevich and S. A. Kauffman, Phys. Rev. Lett. **93**, 048701 (2004).
[33] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevE.100.010301 for details on the annealed approximation and simulation methods.
[34] A. Bhattacharjya and S. Liang, Phys. Rev. Lett. **77**, 1644 (1996).
[35] K. E. Stephan, C.-C. Hilgetag, G. A. P. C. Burns, M. A. O'Neill, M. P. Young, and R. Kötter, Philos. Trans. R. Soc. London B Biol. Sci. **355**, 111 (2000).
[36] O. Sporns and J. D. Zwi, Neuroinformat. **2**, 145 (2004).
[37] O. Sporns, C. J. Honey, and R. Kötter, PLoS One **2**, e1049 (2007).
[38] D. S. Bassett and E. Bullmore, Neuroscientist **12**, 512 (2006).
[39] H. Dale, Proc. R. Soc. Med. **28**, 319 (1935).
[40] D. Watts and S. Strogatz, Nature(London) **393**, 440 (1998).
[41] H. F. Song and X.-J. Wang, Phys. Rev. E **90**, 062801 (2014).
[42] L. Varshney, B. Chen, E. Paniagua, D. Hall, and D. Chklovskii, PLoS Comput. Biol. **7**, e1001066 (2011).
[43] M. van den Heuvel, C. Stam, M. Boersma, and H. H. Pol, Neuroimage **43**, 528 (2008).
[44] V. Eguíluz, D. Chialvo, G. Cecchi, M. Baliki, and A. Apkarian, Phys. Rev. Lett. **94**, 018102 (2005).
[45] S. Hayasaka and P. Laurienti, Neuroimage **50**, 499 (2010).
[46] Y. He, Z. Chen, and A. Evans, Cereb. Cortex **17**, 2407 (2007).
[47] Y. Iturria-Medina, R. Sotero, E. Canales-Rodríguez, Y. Alemán-Gómez, and L. Melia-Garcia, Neuroimage **40**, 1064 (2008).
[48] S. Achard, R. Salvador, B. Whitcher, J. Suckling, and E. Bullmore, Neurosci. **26**, 63 (2006).
[49] G. Gong, Y. He, L. Concha, C. Lebel, D. Gross, A. Evans, and C. Beaulieu, Cereb. Cortex **19**, 524 (2009).
[50] D. Modha and R. Singh, Proc. Natl. Acad. Sci. USA **107**, 13485 (2010).
[51] L. Amaral, A. Scala, M. Barthélémy, and H. Stanley, Proc. Natl. Acad. Sci. USA **97**, 11149 (2000).
[52] P. Hagmann, L. Cammoun, X. Gagandet, R. Meuli, C. Honey, V. Wedeen, and O. Sporns, PLoS Biol. **6**, e159 (2008).
[53] D. de Santos-Sierra, I. Sendiña-Nadal, I. Leyva, J. A. Almendral, S. Anava, A. Ayali, D. Papo, and S. Boccaletti, PloS One **9**, e85828 (2014).
[54] Y. C. Lo, C. T. Li, and S. D. Lin, *International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing, SocialCom-PASSAT 28* (IEEE, Piscataway, NJ, 2012), pp. 229–238.
[55] A. Barabási and R. Albert, Science **286**, 509 (1999).
[56] B. Haider, A. Duque, A. R. Hasenstaub, and D. A. McCormick, J. Neurosci. **26**, 4535 (2006).
[57] J. L. R. Rubenstein and M. M. Merzenich, Genes Brain Behav. **2**, 255 (2003).
[58] E. C. Davenport, B. R. Szulc, J. Drew, J. Taylor, T. Morgan, N. F. Higgs, G. López-Doménech, and J. T. Kittler, Cell Rep. **26**, 2037 (2019).
[59] M. W. Antoine, T. Langberg, P. Schnepel, and D. E. Feldman, Neuron **101**, 648 (2019).
[60] J. D. Murray and X.-J. Wang, in *Computational Psychiatry*, edited by A. Anticevic and J. D. Murray (Academic Press, Cambridge, MA, 2018), pp. 3–25.
[61] O. Yizhar, L. E. Fenno, M. Prigge, F. Schneider, T. J. Davidson, D. J. O'Shea, V. S. Sohal, I. Goshen, J. Finkelstein, J. T. Paz, K. Stehfest, R. Fudim, C. Ramakrishnan, J. R. Huguenard, P. Hegemann, and K. Deisseroth, Nature (London) **477**, 171 (2011).
[62] S. Sahara, Y. Yanagawa, D. D. M. O'Leary, and C. F. Stevens, J. Neurosci. **32**, 4755 (2012).

Supplemental Material to: Critical excitation/inhibition balance in dense neural networks

Lorenz Baumgarten* and Stefan Bornholdt†

Institut für Theoretische Physik, Universität Bremen, 28759 Bremen, Germany
(Dated: June 14, 2019)

I. ANNEALED APPROXIMATION

We assume that all nodes are active, which is close to true at the critical point, and that the degree distribution is narrow enough that it can be approximated by a single peak at k. Consider a single node with a fixed number k of incoming signals. The probability of any specific signal Z is

$$p(Z) = \binom{k}{k_+} F_+^{k_+} F_-^{k_-}, \tag{1}$$

where $F_- = 1 - F_+$ and k_+ and k_- are the numbers of incoming excitatory and inhibitory signals, respectively. We want to calculate the probability p_s that the node will change its state in the next timestep if one of the incoming signals is turned off. This can only happen if the sum of incoming signals is

$$k_+ - k_- = h \tag{2}$$
$$\text{or } k_+ - k_- = h + 1. \tag{3}$$

Note that for given k and h only one of these two cases can occur since k and h need to both be odd or both be even for the first case and have different parities for the second case.
In the first case, an inhibitory incoming signal needs to be turned off to effect a flip. Therefore, the fraction of connections whose disabling would produce a flip is

$$\frac{k_-}{k} = \frac{k-h}{2k}, \tag{4}$$

and for the second case, where an excitatory connection has to be turned off to effect a flip, the fraction is

$$\frac{k_+}{k} = \frac{k+h+1}{2k}. \tag{5}$$

The respective damage spreading probabilities are

$$\begin{aligned} p_\mathrm{s}^{(-)} &= \frac{\sum_{Z\in Z_{(-)}} p(Z)}{\sum_Z p(Z)} \frac{k-h}{2k} \\ &= \binom{k}{\frac{k+h}{2}} F_+^{\frac{k+h}{2}} F_-^{\frac{k-h}{2}} \frac{k-h}{2k}, \end{aligned} \tag{6}$$

$$\begin{aligned} p_\mathrm{s}^{(+)} &= \frac{\sum_{Z\in Z_{(+)}} p(Z)}{\sum_Z p(Z)} \frac{k+h+1}{2k} \\ &= \binom{k}{\frac{k+h+1}{2}} F_+^{\frac{k+h+1}{2}} F_-^{\frac{k-h-1}{2}} \frac{k+h+1}{2k}. \end{aligned} \tag{7}$$

Because we assume high activity, which requires $k_+ - k_- > h$, equation (7) is used in the main text. For $k \gg h$ and $k \gg 1$, the probabilities can be approximated by

$$p_\mathrm{s}^{(-)} \approx p_\mathrm{s}^{(+)} \approx p_\mathrm{s} \approx \frac{1}{2}\binom{k}{\frac{k}{2}} (F_+F_-)^{\frac{k}{2}}. \tag{8}$$

A sensitivity of $\lambda = 1$ is equivalent to a damage-spreading probability of

$$p_\mathrm{s} = \frac{1}{k}. \tag{9}$$

We use Stirling's approximation on the binomial coefficient in p_s to arrive at

$$\begin{aligned} &\frac{1}{k} = \frac{2^k}{k}\sqrt{\frac{k}{2\pi}} (F_+F_-)^{\frac{k}{2}} \\ \Leftrightarrow\ &0 = F_+^2 - F_+ + \frac{1}{4}\left(\frac{2\pi}{k}\right)^{\frac{1}{k}} \\ \Rightarrow\ &F_+ = \frac{1}{2}\left[1 \pm \left\{1 - \left(\frac{2\pi}{k}\right)^{\frac{1}{k}}\right\}^{\frac{1}{2}}\right]. \end{aligned} \tag{10}$$

Only the solution with $F_+ > 0.5$ is realistic, since the assumption of high activity does not hold for $F_+ < 0.5$.

II. METHODS

A. Measuring avalanche sizes and lengths

We initialize random networks with random activity and let their dynamics evolve until the network runs into an attractor. If the average sensitivity within the attractor is within a small margin of one ($1-\delta < \lambda < 1+\delta$ with

* lbaumgarten@itp.uni-bremen.de
† bornholdt@itp.uni-bremen.de

$\delta = 10^{-2}$), we change the state of one node and again simulate the network dynamics until the network either returns to its old attractor or reaches a new one. During this procedure, we count the number of nodes that had a different state than they would have had in the untouched initial attractor at the corresponding time, and the number of timesteps it takes to arrive at an attractor. This is repeated for all nodes in the network.

B. Existence of the second critical point

For large clustering coefficients and high thresholds, transient lengths can become exceedingly long on the left flank, and therefore the assertion in the main text that letting networks evolve for $2 \cdot 10^3$ timesteps before measuring is sufficient to probe the network's dynamics may not always be correct here. Some left flanks may seemingly exist, but will eventually die out after a long time. Although the measurements shown in the main text were, unless otherwise stated, not explicitly done after the transient period, because waiting until the network reaches an attractor is not feasible for large sensitivities, the existence of all critical points shown was verified by letting the network reach an attractor and measuring the average sensitivity within the attractor.
Further, we also tested whether this left flank was merely an aberration caused by the WS model's unrealistic ring structure by repeating our measurements for a two-dimensional network in which node position's were randomly chosen within a square area and node connections were established according to a decreasing exponential probability function of node distance (with periodic boundary conditions). By varying this probability function, different clustering coefficients could be achieved. We verified that the left flank, and therefore also the second critical point, do exist in these more realistic networks.

C. Scale-free and exponentially truncated scale-free graphs

For our preferential attachment, we do not differentiate between a node's in- and outdegree, i.e. the probability of a node being chosen via preferential attachment is proportional to the sum of a node's in- and outdegree, and the probability of a node being chosen as the origin or the target of a node are also equal. Therefore, for simplicity, we refer to the sum of in- and outdegree simply as the degree in this section, as if the network were undirected. We start with a single node and add one edge between two nodes, both the edge's origin and target chosen by preferential attachment, in every two timesteps, so that the network's total degree increases by one per timestep. After every $2m$ timesteps, we also add one node to the system. Ignore for now that it is not always possible to add edges to the system for small t. We do not, however, use the nodes' actual degrees to choose which nodes will gain an edge, but instead calculate expected degrees for the nodes. The expected degree at timestep t_j of a node v_k that was introduced to the network at time t_k is

$$\begin{aligned} &ExpDeg(t_j, v_k) \\ &\quad = ExpDeg(t_{j-1}, v_k) + \frac{ExpDeg(t_{j-1}, v_k)}{total(t_{j-1})} \\ &\quad = ExpDeg(t_{j-1}, v_k)(1 + 1/total(t_{j-1})) \\ &\quad = ExpDeg(t_k, v_k) \prod_{i=k}^{j-1} (1 + 1/total(t_i)), \end{aligned} \tag{11}$$

where $total(t_j)$ is what we call the system's total adjusted degree at time t_j. In our definition of a node's adjusted degree (and the expected degree), we include a bias δ that is added to a node's number of edges. The total adjusted degree is therefore

$$total(t_j) = t_j + \delta n_j, \tag{12}$$

where n_j is the number of nodes in the system at time t_j. This bias is necessary because all nodes start with zero edges and would therefore never receive any new edges for $\delta = 0$. Using the nodes' expected instead of their actual degrees enables us to parallelize our algorithm, significantly speeding up a network's initialization. Note that for the expected degree the issue of being unable to add any edges for small network sizes is irrelevant. We simply pretend that nodes had the degree they could have, had we actually added an edge in every two timesteps, which still simulates preferential attachment. This simplistic approach enables us to simplify products in the following calculations which would otherwise be computationally expensive.
More useful than every single node's degree, however, is the cumulative degree

$$\begin{aligned} &ECumDeg(t_j, v_k) \\ &\quad = \sum_{i=1}^{k} ExpDeg(t_j, v_i) \\ &\quad = \sum_{i=1}^{k} ExpDeg(t_k, v_i) \prod_{i=k}^{j-1} (1 + 1/total(t_i)) \\ &\quad = total(t_k) \prod_{i=k}^{j-1} (1 + 1/total(t_i)) \\ &\quad = total(t_k) \cdot \xi(t_k, t_j), \end{aligned} \tag{13}$$

as it allows us to do a fast binary search to find the node an edge needs to be connected to, without having to calculate and sum up expected degrees for a large number of nodes. The factor $\xi(t_k, t_j)$ can easily be calculated when the number of nodes currently in the network n is

constant

$$\begin{aligned}\xi(t_k, t_j) &= \prod_{i=k}^{j-1} \left(1 + \frac{1}{i + \delta n}\right) \\ &= \left(\frac{k + \delta n + 1}{k + \delta n}\right)\left(\frac{k + \delta n + 2}{k + \delta n + 1}\right) \cdots \left(\frac{j + \delta n}{j + \delta n - 1}\right) \\ &= \frac{j + \delta n}{k + \delta n}. \end{aligned} \tag{14}$$

To calculate $\xi(t_k, t_j)$ even if nodes are added to the system between t_k and t_j, we split the product into parts with constant n

$$\begin{aligned}\xi(t_k, t_j) =& \frac{k + 2m + \delta n_k}{k + \delta n_k} \times \frac{k + 4m + \delta(n_k + 1)}{k + 2m + \delta n_k} \\ &\times \cdots \times \frac{k + 2m(n_j - n_k) + \delta n_j}{k + 2m(n_j - n_k - 1) + \delta(n_j - 1)} \\ &\times \frac{j + \delta n_j}{k + 2m(n_j - n_k) + \delta n_j}, \end{aligned} \tag{15}$$

The product in the first two lines can be written as

$$\begin{aligned}&\prod_{i=0}^{n_j - n_k - 1} \frac{k + 2mi + 2m + \delta(n_k + i)}{k + 2mi + \delta(n_k + i)} \\ &= \prod_{i=0}^{n_j - n_k - 1} 1 + \frac{1}{c + i(1 + \delta/2m)} \\ &= \frac{\Gamma(\frac{c}{d})\Gamma(b + \frac{c}{d} + \frac{1}{d})}{\Gamma(\frac{c}{d} + \frac{1}{d})\Gamma(b + \frac{c}{d})}, \end{aligned} \tag{16}$$

with $b = n_j - n_k,\ c = (k + \delta n_k)/2m,\ d = 1 + \delta/2m.$

The cumulative degree is then

$$\begin{aligned}&ECumDeg(t_j, v_k) = total(t_k) \\ &\times \frac{\Gamma(\frac{c}{d})\Gamma(b + \frac{c}{d} + \frac{1}{d})}{\Gamma(\frac{c}{d} + \frac{1}{d})\Gamma(b + \frac{c}{d})} \times \frac{j + \delta n_j}{k + 2m(n_j - n_k) + \delta n_j}. \end{aligned} \tag{17}$$

At time t_j, a node can now be chosen via preferential attachment by choosing a random number η between zero and one, and finding the first v_k for which $ECumDeg(t_j, v_k)/total(t_j) \geq \eta$.

It is likely that edges between the largest nodes will be added multiple times during a network's initialization. As it would be computationally expensive to check whether an edge already exists in the system, we ignore additional edges and remove them after initialization so weights remain as ± 1. This, and also the inability to add an edge every two timesteps for low t, leads to the network's eventual average degree being unpredictable. Therefore, we repeat the initialization process, adding up all of the single initializations' edges, until $k \approx k^*$, where k is the network's average degree and k^* is the desired average degree. Finally, we add edges from i permutations of the network's nodes to the unpermutated nodes, with i being the smallest integer with $i > h$, to ensure that every node has a chance of being activated and participating in the network's dynamics.

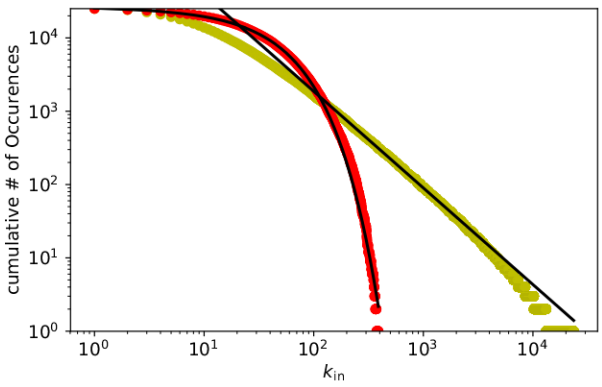


FIG. 1. Cumulative in-degree distributions for a scale-free graph with $\delta = 1$, $m = 10$, $k = 40$, $N = 2.5 \cdot 10^4$, $h = 0$, and $C \approx 0.52$ (yellow) and an exponentially truncated scale-free graph with $\delta = 40$, $m = 2$, $k = 40$, $N = 2.5 \cdot 10^4$, $h = 0$, and $C \approx 0.005$ (red). Solid black lines show power law and exponentially truncated power law fits, respectively. The power law's slope is -1.3.

Our algorithm has two parameters, δ and m, that enable us to tune the degree distribution as well as the clustering coefficient. Generally, lower δ and higher m lead to higher clustering coefficients and scale-free degree distributions, whereas higher δ and lower m lead to low clustering coefficients and exponentially truncated power law degree distributions, see FIG. 1.

Self-organized criticality in neural networks from activity-based rewiring

Stefan Landmann, Lorenz Baumgarten, and Stefan Bornholdt. "Self-organized criticality in neural networks from activity-based rewiring". In: *Physical Review E* 103.3 (2021), p. 032304.

Self-organized criticality in neural networks from activity-based rewiring

Stefan Landmann, Lorenz Baumgarten, and Stefan Bornholdt*
Institut für Theoretische Physik, Universität Bremen, Germany

(Received 18 September 2020; accepted 12 February 2021; published 3 March 2021)

Neural systems process information in a dynamical regime between silence and chaotic dynamics. This has lead to the *criticality hypothesis*, which suggests that neural systems reach such a state by self-organizing toward the critical point of a dynamical phase transition. Here, we study a minimal neural network model that exhibits self-organized criticality in the presence of stochastic noise using a rewiring rule which only utilizes local information. For network evolution, incoming links are added to a node or deleted, depending on the node's average activity. Based on this rewiring-rule only, the network evolves toward a critical state, showing typical power-law-distributed avalanche statistics. The observed exponents are in accord with criticality as predicted by dynamical scaling theory, as well as with the observed exponents of neural avalanches. The critical state of the model is reached autonomously without the need for parameter tuning, is independent of initial conditions, is robust under stochastic noise, and independent of details of the implementation as different variants of the model indicate. We argue that this supports the hypothesis that real neural systems may utilize such a mechanism to self-organize toward criticality, especially during early developmental stages.

DOI: 10.1103/PhysRevE.103.032304

I. INTRODUCTION

Neural systems, to efficiently process information, have to operate at an intermediate level of activity, avoiding both a chaotic regime as well as silence. It has long been speculated that neural systems may operate close to a dynamical phase transition that is naturally located between chaotic and ordered dynamics [1–4]. Indeed, recent experimental results support the criticality hypothesis, most prominently the so-called neuronal avalanches, specific neuronal patterns in the resting state of cortical tissue which are power-law distributed in their sizes and durations [5–9]. Studies suggesting that neural systems exhibit optimal computational properties at criticality [10–12] further support the criticality hypothesis.

However, which mechanisms could drive such complex systems toward a critical state? Ideally, criticality is reached by a decentralized, self-organized mechanism, an idea known as self-organized criticality (SOC) [13–15]. Models for self-organized criticality in neural networks were discussed even before experimental indications of neural criticality [5], including a self-organized critical adaptive network model [3,16], as well as an adaptation of the Olami-Feder-Christensen SOC model for earthquakes [17] in the context of neural networks [18].

The seminal paper of Beggs and Plenz [5] eventually inspired a multitude of self-organized critical neural network models, often with a particular focus on biological details in the self-organizing mechanisms. Some of these mechanisms are based on short-term synaptic plasticity [19], spike timing dependent plasticity [20], long-term plasticity [21], while others rely on Hebbian-like rules [22–24] or anti-Hebbian rules [25]. For recent reviews on criticality in neural systems see Refs. [26–31].

In this paper we revisit the earliest model, the self-organized critical adaptive network [3], in the wake of the observation of neural avalanches and ask two questions: Does this general model still self-organize to criticality when adapted to the particular properties of neural networks? How do its avalanche statistics compare to experimental data? Our aim remains to formulate the simplest possible model, namely, an autonomous dynamical system that generates avalanche statistics without external parameters and without any parameter tuning.

The original SOC network model [3] had been formulated as a spin system in the tradition of statistical physics, with binary nodes of values $\sigma \in \{-1, 1\}$, corresponding to inactive and active states respectively. To study avalanches in the critical state, a translation to Boolean state nodes $\sigma(t) \in \{0, 1\}$ is necessary, as has been formulated for modeling biological networks in Ref. [32]. For an adaptive neural network model with rewiring based on the correlation between neighboring nodes [16], we demonstrated earlier that in such a binary realization, avalanche statistics become accessible and exhibit self-organized criticality [33]. Nevertheless, the correlation-based rewiring of that model is not the simplest possible rule, and its algorithmic implementation falls short of a fully autonomous dynamical system: Its adaptation rule still uses data from different simulation runs to determine the synaptic change to be performed.

Therefore, we here reconsider the simpler activity-based rewiring and reformulate our model as a fully autonomous system with adaptation dynamics based on solely local information. It uses Boolean state nodes on a network without a predefined topology. The network topology changes by link adaptations (addition and removal of links) based on local

*bornholdt@itp.uni-bremen.de

information only, namely, the temporally averaged activity of single nodes. Neither information of the global state of the system nor information about neighboring nodes, e.g., activity correlations [33] or retrosynaptic signals [21], are needed. Last, it is well motivated by abundant evidence for homeostatic processes in neural plasticity.

II. THE MODEL

Let us now define our model in detail. Consider a directed graph with N nodes with binary states $\sigma(t) \in \{0, 1\}$ representing resting and firing nodes. Signals are transmitted between nodes i and j via activating or inhibiting links $c_{ij} \in \{-1, 1\}$. If there is no connection between i and j, then we set $c_{ij} = 0$. Besides the fast dynamical variables $\sigma(t)$ of the network, the connections c_{ij} form a second set of dynamical variables of the system which are evolving on a considerably slower timescale than the node states $\sigma(t)$. Let us define these two dynamical processes, activity dynamics and network evolution, separately.

A. Activity dynamics

The state $\sigma_i(t + \Delta t)$ of node i depends on the input

$$f_i(t) = \sum_{j=1}^{N} c_{ij}\,\sigma_j(t) \tag{1}$$

at some earlier time t. For simplicity of simulation we here choose a time step of $\Delta t = 1$ and perform parallel update such that this time step corresponds to one sweep where each node is updated exactly once. Please note that random sequential update as well as an autonomous update of each node according to a given internal timescale is possible as well and does not change our results. Having received the input $f_i(t)$, node i will be active at $t + 1$ with a probability

$$\text{Prob}[\sigma_i(t + 1) = 1] = \frac{1}{1 + \exp[-2\beta(f_i(t) - 0.5)]}. \tag{2}$$

Here, β is an inverse temperature, solely serving the purpose of quantifying the amount of noise in the model. For the low-temperature limit $\beta \to \infty$ the probability Eq. (2) becomes a step function which equals 0 for $f_i < 0.5$ and 1 for $f_i > 0.5$. This function broadens for decreasing β, also allowing for nodes being active once in a while without receiving any input. Such idling activity is observed in cortical tissue and will play a role in the evolutionary dynamics as defined in the following.

This model attempts to formulate the simplest rules for the activity dynamics possible, i.e., with the fewest states of the nodes and the fewest parameters. Thus the dynamics neither consider a refractory time nor a nonzero activation threshold. Nevertheless, as shown in Sec. V, the mechanism driving the network toward criticality works in very different biologically inspired implementations of the model. This suggests that despite being a coarse simplification of a real biological system, the model can represent basic mechanisms that can also be at work in real neuronal systems.

B. Network evolution

Following the natural timescale separation between fast neuron dynamics and slow change of their connectivity, we here implement changes of the network structure itself on a well-separated slow timescale. For every time step, each node is chosen with a small probability $\frac{\mu}{N} \ll 1$ and its connectivity is changed based on its average activity $A_i = \langle \sigma_i \rangle_W$ over the time window of the last W time steps according to the following rules:

(i) $A_i = 0$: add a new incoming link $c_{ij} = 1$ from another randomly chosen node j.

(ii) $A_i = 1$: add a new incoming link $c_{ij} = -1$ from another randomly chosen node j.

(iii) $A_i \notin \{0, 1\}$: remove one incoming link of node i.

Thus, inactive (i.e., nonswitching) nodes receive new links, while active (i.e., switching) nodes lose links. These rules prevent the system from reaching, both, an ordered phase where all nodes are permanently frozen, as well as a chaotic regime with abundant switching activity. In particular, the system is driven toward a dynamical phase transition between a globally ordered and a globally chaotic phase.

Note that rewiring is based on locally available information only. To simulate the way a single cell could keep a running average, we also implemented the average activity of a node as $A_i(t + 1) = \sigma(t + 1)(1 - \alpha) + A_i(t)\alpha$ as the basic principle, a biochemical average would be taken. Here, the parameter $\alpha \in [0, 1]$ determines the temporal memory of the nodes (instead of the averaging time window parameter W). Since the newly defined A_i can only approach but never attain 0 or 1, we have to reformulate the criteria which determine the type of rewiring to be performed. The condition for a node to receive an activating link is transformed from $A_i = 0$ to $A_i < \epsilon$ with $\epsilon \ll 1$, the other criteria are changed correspondingly. Then, we find that the model works accordingly.

For practical purposes, we perform the rewiring of only one randomly chosen node i after every $\frac{N}{\mu}$ sweeps, instead of selecting every node with a certain probability $\frac{\mu}{N}$ at each time step. Both implementations yield the same results. To minimize the number of model parameters, we quantify the separation between fast and slow timescales in the model with one parameter by setting $\frac{N}{\mu} = W$ and using W as the parameter.

The proposed rules for the network evolution are inspired by synaptic wiring and rewiring as observed in early developmental stages of neural populations or during the rewiring of dissociated cortical cultures [34]. In these systems, homeostatic plasticity mechanisms are at work, which lead to increased activity of overly inactive neurons and vice versa. In Ref. [35] it was found that the application of inhibitory neurotransmitters to pyramidal neurons in isolated cell cultures, and thus a decrease of activity, leads to an increased outgrowth of neurites. In contrast, if excitatory neurotransmitters are applied, a degeneration of dendritic structures is induced [35–37]. These observations were confirmed in experiments where electrical stimulation of neurons showed to inhibit dendritic outgrowth [38] and blocking of activity resulted in increased growth of dendrites [39,40]. Thus, if a neuron is overly inactive or active, it "grows and retracts its dendrites to find an optimal level of input..." [41], which is

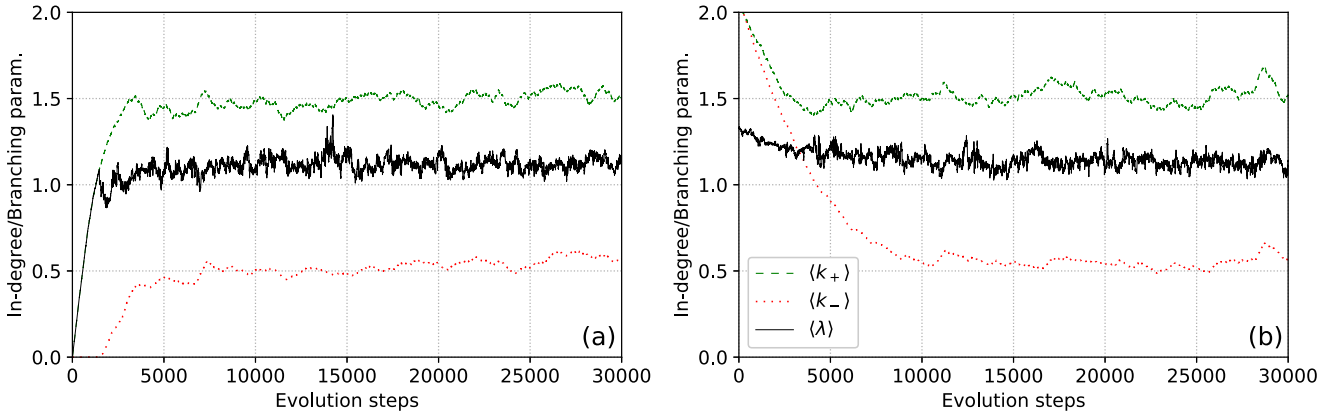


FIG. 1. (a) Time series of the average in-connectivities and branching parameter for $N = 1000$, $\beta = 10$, $W = 1000$, starting from a completely unconnected network. After a transient period, the average connectivities and the branching parameter become stationary. The branching parameter fluctuates around $\langle\lambda\rangle = 1.10 \pm 0.11$, indicating possible criticality. (b) Evolution starts with an average connectivity of $\langle k\rangle = 2$ for activating and inhibiting links. Even though having a very different initial configuration, the system evolves toward a similar steady state as found in panel (a).

mimicked by the proposed rewiring rules. Similar homeostatic adaption rules have been successfully used to model cortical rewiring after deafferentation [42]. In recent models, homeostatic regulation has been proposed as a key mechanism of self-organization and modulation of neural dynamics [43,44].

III. EVOLUTION OF THE NETWORK STRUCTURE

The evolution of the network starts with a specified initial configuration of links $\mathbf{c}(t = 0)$ and the state of all nodes set to $\boldsymbol{\sigma}(t = 0) = \mathbf{0}$. Doing so, all activity originates from small perturbations caused by stochastic noise. Applying the rewiring rules, the system then evolves toward a dynamical steady state with characteristic average numbers of activating and inhibiting links.

As a convenient observable of the dynamical state of the network, and an approximate indicator of a possible critical state of the network, we measure the branching parameter $\langle\lambda\rangle$ by calculating, for every node i, how many neighbor nodes λ on average change their state at time $t + 1$ if the state of i is changed at time t. Averaging λ over the network indicates the dynamical regime of the network, where $\langle\lambda\rangle = 1$ is often used as an indicator of criticality. Note that, by construction, $\langle\lambda\rangle$ depends on the connectivity matrix $c_{ij}(t)$ and on the state vector $\boldsymbol{\sigma}(t)$ and, therefore, has to be considered with some caution. For example, its critical value may differ from one when the evolved networks develop community structure or degree correlations between in- and out-links or between nodes [10]. Therefore, we will here use the branching parameter for a qualitative assessment of the network evolution, only, and analyze criticality with tools from dynamical scaling theory below.

Let us now turn to the evolutionary dynamics of the model, starting from a random network $\mathbf{c}(t = 0)$ with only the average connectivity specified at $t = 0$. Figure 1(a) shows the time series of the average number of incoming activating and inhibiting links per node $\langle k_+\rangle$ and $\langle k_-\rangle$ starting from a fully unconnected network. The figure also shows the temporal evolution of the branching parameter $\langle\lambda\rangle$. At the beginning of the network evolution, there are only a few links between the nodes, and noise-induced activity dies out fast. Therefore, the activity is very low, and only activating links are added. As a result, the branching parameter increases. When the value of $\langle k_+\rangle$ approaches one, the activity starts to propagate through the network and some nodes become permanently active. This causes the rewiring algorithm to insert inhibiting links. After some transient time, the average connectivities become stationary and fluctuate around a mean value. The branching parameter also becomes stationary and fluctuates around a value near one, indicating a possible critical behavior. The ratio of inhibiting links to activating links approximately attains $\langle k_-\rangle/\langle k_+\rangle \approx 0.3$ which is close to the ratio of inhibition/activation typically observed in real neural systems [45]. The connectivity in the stationary states exhibits Poisson-distributed degree distributions of incoming and outgoing links.

Figure 1(b) shows the evolution of the average connectivities with different initial conditions. Here, the initial average connectivities are chosen as $\langle k_+\rangle = \langle k_-\rangle = 2$. In contrast to the starting configuration in Fig. 1(a), the network is densely connected and the nodes change their states often. Since the nodes rarely stay in the same state during the averaging time W, links are preferentially deleted in the beginning. After a transient period, the system reaches a stationary steady state similar to the one already observed in Fig. 1(a), indicating independence from initial conditions.

This scenario is reminiscent of synaptic pruning during adolescence, where in some regions of the brain approximately 50% of synaptic connections are lost [46]. It is hypothesized that this process contributes to the observed increase in efficiency of the brain during adolescence [47]. In the proposed model, starting with the densely connected network shown in Fig. 1(b), the branching parameter is considerably larger than one. In this state, information transmission and processing are difficult since already small perturbations percolate through the entire network. The

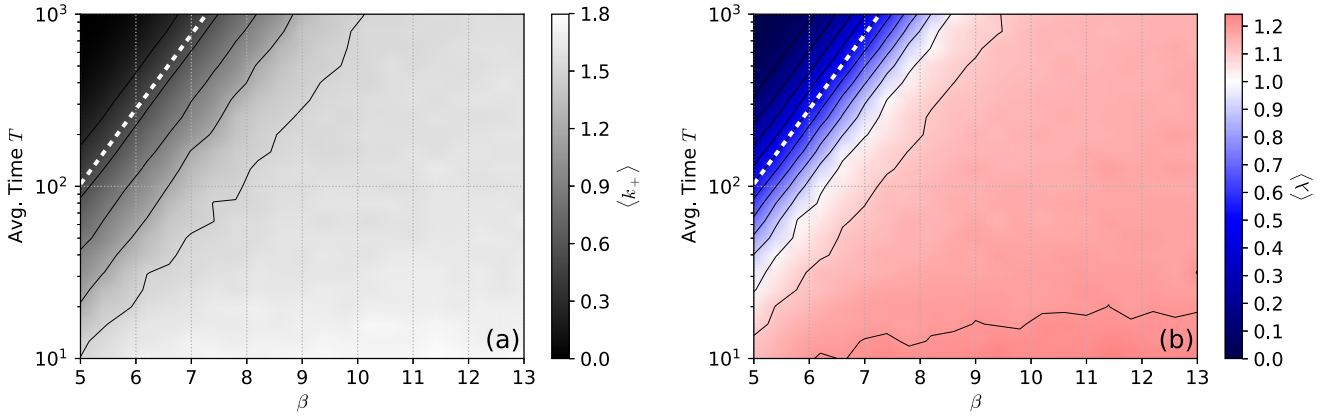


FIG. 2. (a) The average connectivity of activating incoming links $\langle k_+ \rangle$ for different values of the averaging length W and the inverse temperature β. The white dotted line is the upper bound of W given by Eq. (4). (b) The average branching parameter $\langle \lambda \rangle$ for different values of the averaging window length W and the inverse temperature β. The average branching parameter is close to a typical value near one over a broad range of (β, W). The data was obtained by averaging over 30 000 evolution steps, system size is $N = 1000$.

decrease in the number of links leads to a network with a branching parameter close to one, much better suited for information processing tasks.

To explore the parameter dependency of the model, let us now ask how the steady-state averages of the connectivities and the branching parameter depend on the system parameters (β, W, N). Figure 2(a) shows the average connectivity of activating incoming links over a broad range of parameter space. A prominent feature is the subcritical region (upper left corner) where the algorithm fails to create connected graphs and the average connectivity of incoming links is far below one. This is due to nodes being predominantly active by noise, instead of signal transmission. If a node i has no incoming links its probability to be turned on at least once by noise during the W time steps is given by

$$\text{Prob}(A_i > 0) = 1 - \left(1 - \frac{1}{1 + e^{\beta}}\right)^{W}. \tag{3}$$

Therefore, demanding that on average not more than half of the nodes should be turned on by noise during W steps gives an upper bound for the time window W:

$$W_{\max} = -\frac{\log 2}{\log\left(1 - \frac{1}{1+e^{\beta}}\right)}. \tag{4}$$

This boundary is shown as a white dashed line in Fig. 2(a), obviously being a good approximation for the boundary of the subcritical region. Most importantly, we see that if β is sufficiently large, i.e., if the noise is sufficiently small, then there always is a region in which connected networks emerge. Since $W_{\max}$ is independent of system size N, this also holds for large systems. Figure 2(b) shows the average branching parameter for the same range of (β, W) as Fig. 2(a). Note that $\langle \lambda \rangle$ is close to a value slightly larger than one, over a wide range of noise and averaging times. To explore whether this indicates criticality (with a critical branching parameter value larger than one for the evolved networks), let us now explore other criteria of criticality.

IV. CRITICALITY

An important feature of critical systems is scale-independent behavior, meaning that close to a phase transition similar patterns can be observed on all scales. Near criticality, correlations between distant parts of the system do not vanish and microscopic perturbations can cause influences on all scales. This also implies that power laws occur in many observables, as, e.g., in the size distribution of fluctuations.

A. Avalanches of perturbation spreading

Let us now investigate the statistics of avalanches of perturbations spreading on the networks. Note that the network evolution drives the system toward a state where activity never dies out. Therefore, we cannot consider avalanches of activity-spreading, as usually done in numerical experiments, with one perturbation at a time. The problem of persistent activity could be circumvented by introducing an activity threshold that defines the start and the end of avalanches as done in Ref. [48]. This procedure, nevertheless, is not reliable since the introduction of an activity threshold can generate power-law-like scaling from uncorrelated stochastic processes as was shown in Ref. [49]. Instead, showing that the size and duration of the fluctuations are power-law distributed is a more reliable procedure commonly used in statistical physics [50]. This method is related to the determination of the Boolean Lyapunov exponent, which was used, e.g., in Ref. [51] to examine the critical behavior of neural networks.

First, we let the system evolve until the branching parameter and the average connectivities reach steady average values. Then, noise is deactivated and a copy $\boldsymbol{\sigma}_c$ of the network is made. One node of this copy is chosen at random and its state is flipped: If it was active, then it is turned inactive and vice versa. By comparing the temporal evolution of the unperturbed system $\boldsymbol{\sigma}$ and the perturbed system $\boldsymbol{\sigma}_c$ one can examine the spreading of this perturbation. For quantifying the "difference" between the two copies it is convenient to use the Hamming distance of the state vectors $d_H(\boldsymbol{\sigma}, \boldsymbol{\sigma}_c)$ which is defined as the number of differing entries in $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_c$,

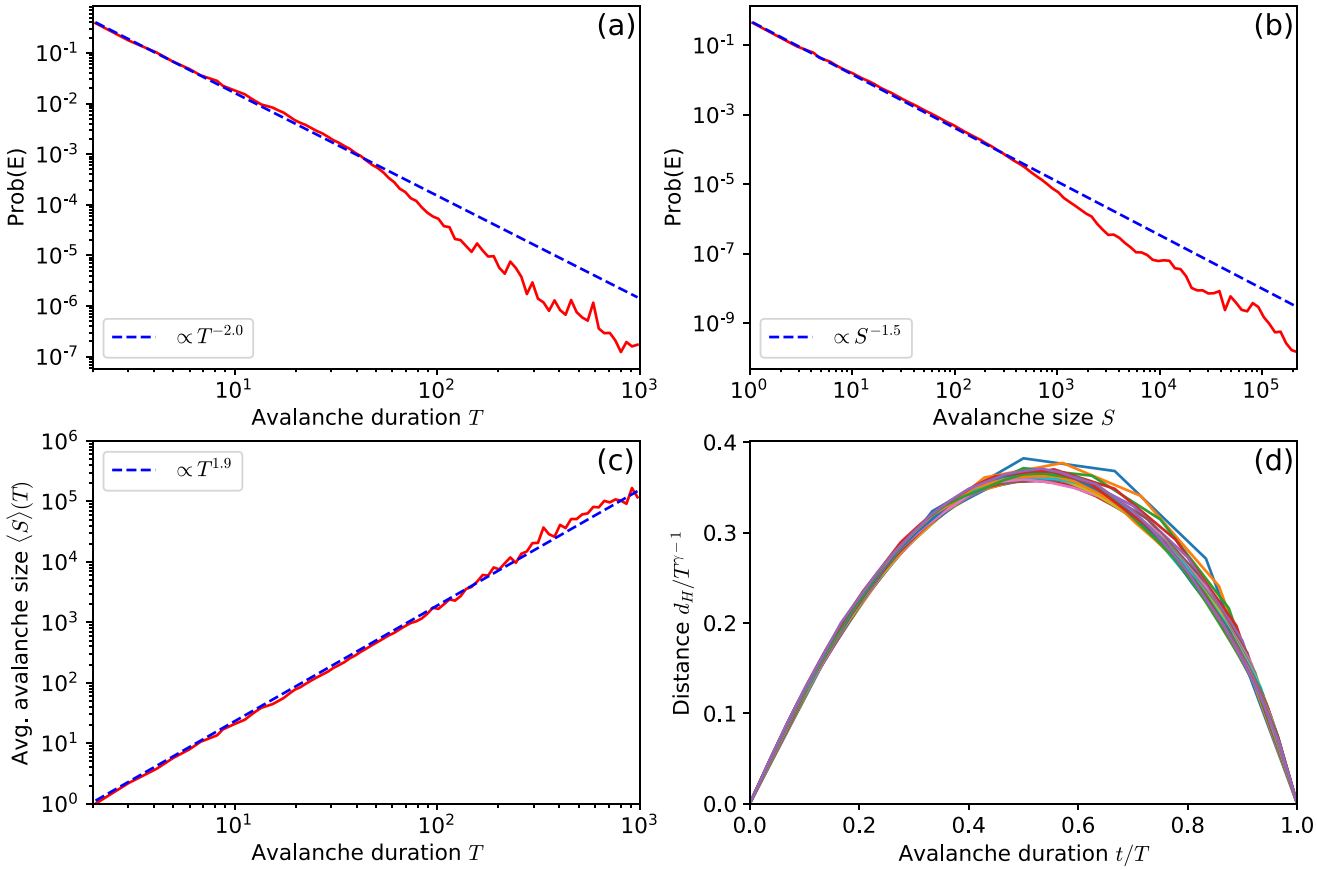


FIG. 3. Avalanche statistics and collapse of avalanche profiles. (a) Avalanche duration distribution. (b) Avalanche size distribution. (c) Average avalanche size over avalanche duration. (d) Collapse of avalanche shapes. The curves show the Hamming distance during avalanches of lengths between 5 and 30. $N = 2000$, $W = 1000$, $\beta = 10$, data from 6×10^6 avalanches.

i.e., the number of nodes which deviate from each other in their states. During the examination of one perturbation, the rewiring algorithm is not in action.

Performing simulations we found that in most cases $d_H(\boldsymbol{\sigma}, \boldsymbol{\sigma}_c) \to 0$ after some time, which means that the perturbed system falls back onto the attractor of the unperturbed system. For a system of, e.g., 2000 nodes with $\beta = 10$ and $W = 1000$, this was observed in more than 90% of all perturbations.

It is straightforward to define the avalanche duration T as the time between the start of the perturbation and the return of $\boldsymbol{\sigma}_c$ to the same attractor as $\boldsymbol{\sigma}$ and the avalanche size S as the cumulative sum of the Hamming distances between $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_c$ during the avalanche:

$$S = \sum_{t=0}^{T} d_H[\boldsymbol{\sigma}(t), \boldsymbol{\sigma}_c(t)]. \tag{5}$$

From universal scaling theory [52] it is expected that these observables exhibit power-law scaling at criticality:

$$\text{Prob}(S) \sim S^{-\tau}, \tag{6}$$

$$\text{Prob}(T) \sim T^{-\alpha}. \tag{7}$$

Furthermore, it should also hold that the relation between the average avalanche size and the avalanche duration shows power-law scaling

$$\langle S \rangle(T) \sim T^{-\gamma}, \tag{8}$$

with the exponents fulfilling the relation

$$\frac{\alpha - 1}{\tau - 1} = \gamma. \tag{9}$$

To further verify criticality it is possible to explicitly show the scale-freeness of the avalanche dynamics. This can be done by determining the average avalanche profiles (avalanche size over time) for different avalanche durations. Scaled properly, these shapes should collapse onto one universal curve if the system is critical.

B. Results

Figure 3 shows the distribution of avalanche sizes and durations as well as the collapse of avalanche profiles for avalanches of different durations. Exponents were fitted using the estimator for discrete integer variables described in Ref. [53].

Figure 3(a) shows that the avalanche duration scales with an exponent of $\alpha \approx 2.0332 \pm 0.0004$ up to the square root of the system size.

Figure 3(b) reveals a power-law scaling of the avalanche size with an exponent of $\tau \approx 1.5428 \pm 0.0002$. Both

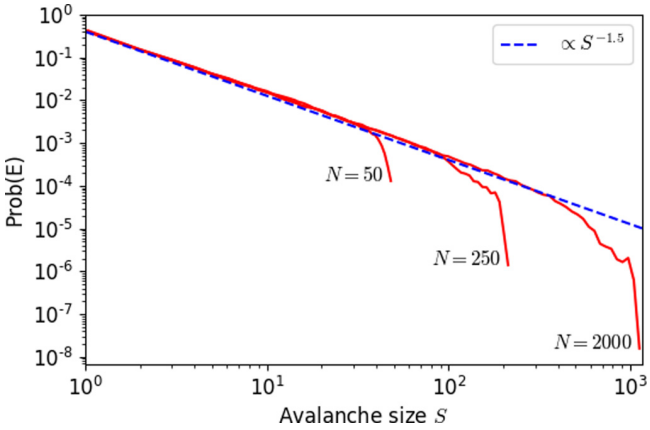


FIG. 4. Scaling of the avalanche size distribution with increasing system size N. Each distribution is obtained from 10^6 avalanches. During one avalanche each node can only contribute once to the avalanche size. Parameters: $\beta = 10, W = 1000$.

exponents α and τ are in line with experimental results [5,8]. Note that the right end of the avalanche duration and size distributions underestimates the true numbers of avalanches, as we here only count avalanches that return to the same attractor as in the unperturbed network. Larger avalanches more frequently end up in a different attractor when perturbed, resulting in a larger fraction of long avalanches not counted in the statistic.

Figure 3(c) shows that the relation between average avalanche size and avalanche duration also exhibits a power-law scaling up to the square root of system size with an exponent of $\gamma \approx 1.92 \pm 0.04$. These exponents fulfill the relation

$$\frac{\alpha - 1}{\tau - 1} = 1.9036 \pm 0.0003 \approx \gamma, \tag{10}$$

strongly suggesting that the system is critical.

Figure 3(d) shows the collapse of the activity curves onto one universal shape, as it was also found in experiments [8], reflecting the fractal structure of the avalanche dynamics.

A further verification of criticality can be found in Fig. 4, which shows the avalanche size distributions for different systems sizes N. Figure 4 uses a different definition of avalanche size than previously introduced. Here, instead of the sum of Hamming distances between perturbed and unperturbed networks, the avalanche size is the number of nodes that, at any time, have had a different state in the perturbed and the unperturbed network, where every node can only contribute once to the avalanche size. This means that the maximum avalanche size is N. With increasing system size the power-law-like regions of the distributions increase, showing that the cut-off is only a finite size effect.

V. OTHER VERSIONS OF THE MODEL

The main goal of this work is to present a minimal adaptive network model that exhibits self-organized critical behavior. At the same time, the model is supposed to be plausible, in the sense that only local information is used to approach the critical state. While we simulate this model on a von Neumann computer, a fully autonomous implementation is possible. To further demonstrate that our model represents a general mechanism and does not depend on particular features of the implementation, also variants of the model were tested.

A. Inhibiting nodes

We tested a variant that uses inhibiting nodes instead of inhibiting links (and excitatory nodes instead of excitatory links). In this modified model, nodes are connected by un-weighted links and the sign of the outgoing signal is determined by the nature of the node. Before starting the evolution of the network a fraction of all nodes is chosen to be inhibitory. Here we typically choose 20–30%, as it is often used as a rough approximation for real neural systems [45] (simulations show that in the frame of the model the exact number is not of importance). If an inhibitory node is active, it contributes a signal -1 to the inputs of all nodes to which it is connected via outgoing links, and vice versa for excitatory nodes. Further, the network evolution rules of our model are rewritten accordingly and now take the simple form:

(i) $A_i = 0$: add a new incoming link from another randomly chosen excitatory node j.

(ii) $A_i = 1$: add a new incoming link from another randomly chosen inhibitory node j.

(iii) $A_i \notin \{0, 1\}$: remove one incoming link of node i.

We find that the dynamics of this modified version closely resembles the dynamics of the original model.

B. Continuous link weights

Choosing discrete link weights $c_{ij} \in \{-1, 0, 1\}$ allows for a minimalistic description of the model and to formulate simple rules for the network evolution. However, to mimic the varying synaptic strengths of a real neural system, a version with continuous link weights has also been examined. We find that the following continuous rewiring rules lead to critical behavior, as well. In the same way as in the original model, after every W time steps, one node i is chosen at random. Depending on its average activity A_i its linkage is changed as described in the following:

(i) $A_i = 0$: randomly choose another node j. If $c_{ij} = 0$, then add a new incoming link $c_{ij} \in [0, \Delta]$. If $c_{ij} \neq 0$ multiply the link weight by a factor $[1 + \delta \,\mathrm{sign}\,(c_{ij})]$.

(ii) $A_i = 1$: randomly choose another node j. If $c_{ij} = 0$, then add a new incoming link $c_{ij} \in [-\Delta, 0]$. If $c_{ij} \neq 0$ multiply the link weight by a factor $[1 - \delta \,\mathrm{sign}\,(c_{ij})]$.

(iii) $A_i \notin \{0, 1\}$: randomly choose one incoming link of i. If $|c_{ij}| < 1$, then set $c_{ij} = 0$; otherwise, decrease the link weight by a factor $(1 - \delta)$.

Hereby, the additional parameters δ and Δ should be chosen such that $\delta \ll 1$ to keep incremental changes small, and $\Delta > 2$ for new links to have a dynamical effect in the face of the threshold update rule. Then the network robustly reaches a critical state.

VI. CONCLUSION AND OUTLOOK

In this article, we tried to sketch the simplest possible neural network model that self-organizes toward a critical state,

while reproducing detailed features of criticality observed in real biological neural systems.

Note that the model involves only three parameters, none of which is critical: The inverse temperature β determining the amount of noise in the model, the averaging time W defining the timescale separation between the fast neural dynamics and the slow homeostatic plasticity, and the system size N. None of these needs fine-tuning and they can be varied over a considerable range.

The homeostatic evolution of the network connectivity is based on the temporally averaged activity of single nodes only. Thus, neither information about the global state of the network, nor information about neighboring nodes is necessary for self-organized criticality in this neural network. The model is a variation of the earlier spin-based network SOC model [3], in an implementation with neurons with states zero and one, with a stochastic update rule, allowing for spontaneous activity, and with an evolution rule that specifies inhibitory and excitatory links separately.

Theoretical studies have demonstrated that neural networks can be tuned to criticality by properly adjusting the ratio of activating and inhibiting nodes/links [48]. This is in line with experimental results, which indicate that critical behavior arises in cortical networks with a balanced activation/inhibition ratio [11,54]. In the model studied here, we observe that the balance of inhibitory and excitatory links self-organizes to a steady state. It is possible that mechanisms of similar form help to keep the balance between activation and inhibition especially during early developmental stages of neural systems [34] where phases of rapid synaptic production [46] and synaptic pruning occur [47].

In contrast to the classical models of self-organized criticality, as, e.g., the sandpile model [13], the Bak-Sneppen model of evolution [15], or the forest fire model [14], the model we study here exhibits critical dynamics over a broad range of noise. Indeed, it even utilizes noise to sustain activity permanently. The origin of the noise resilience of this class of, what we could call "robust self-organized criticality" models, is the fact that the criticality of the system is stored in separate variables, in our case in the links between the nodes, rather than in the dynamical variables, the node states, themselves. Classical SOC models, on the other hand, are more vulnerable against noise as can be seen, for example, in the forest fire model, where criticality emerges as a fractal distribution of tree states that is easily disturbed. In our self-organized critical adaptive network model, in contrast, noise may vary over a broad range.

We have further explored the robustness of the rewiring mechanism in different versions of the model where, for example, inhibiting nodes instead of inhibiting links are implemented or continuous link weights are used. This illustrates that the observed self-organized critical characteristics arise as stable phenomena independent of even major features of the system, only depending on the structure of the rewiring algorithm. Together with the robustness against noise, these observations give strong support to the hypothesis that also real biological neural systems could take advantage of this simple and robust way to self-tune close to a phase transition.

Future work on minimal neural network models showing self-organized critical behavior could focus on how criticality influences learning, as it already has been touched on, e.g., in Refs. [21,24]. Further insights into this field could not only help our understanding of biological neural systems but also motivate new ways of constructing artificial neural networks optimally. The autonomous nature of the self-organized critical adaptive neural network should make it implementable with memristors or other forms of neuromorphic hardware.

ACKNOWLEDGMENTS

We thank G. G. Jensen and J. Zierenberg for discussions and comments on the manuscript.

[1] C. G. Langton, Computation at the edge of chaos: Phase transitions and emergent computation, Physica D: Nonlin. Phenom. **42**, 12 (1990).

[2] A. V. Herz and J. J. Hopfield, Earthquake Cycles and Neural Reverberations: Collective Oscillations in Systems with Pulse-Coupled Threshold Elements, Phys. Rev. Lett. **75**, 1222 (1995).

[3] S. Bornholdt and T. Rohlf, Topological Evolution of Dynamical Networks: Global Criticality from Local Dynamics, Phys. Rev. Lett. **84**, 6114 (2000).

[4] P. Bak and D. R. Chialvo, Adaptive learning by extremal dynamics and negative feedback, Phys. Rev. E **63**, 031912 (2001).

[5] J. M. Beggs and D. Plenz, Neuronal avalanches in neocortical circuits, J. Neurosci. **23**, 11167 (2003).

[6] J. M. Beggs and D. Plenz, Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures, J. Neurosci. **24**, 5216 (2004).

[7] T. Petermann, T. C. Thiagarajan, M. A. Lebedev, M. A. Nicolelis, D. R. Chialvo, and D. Plenz, Spontaneous cortical activity in awake monkeys composed of neuronal avalanches, Proc. Natl. Acad. Sci. U.S.A. **106**, 15921 (2009).

[8] N. Friedman, S. Ito, B. A. Brinkman, M. Shimono, R. L. DeVille, K. A. Dahmen, J. M. Beggs, and T. C. Butler, Universal Critical Dynamics in High Resolution Neuronal Avalanche Data, Phys. Rev. Lett. **108**, 208102 (2012).

[9] W. L. Shew, W. P. Clawson, J. Pobst, Y. Karimipanah, N. C. Wright, and R. Wessel, Adaptation to sensory input tunes visual cortex to criticality, Nat. Phys. **11**, 659 (2015).

[10] D. B. Larremore, W. L. Shew, and J. G. Restrepo, Predicting Criticality and Dynamic Range in Complex Networks: Effects of Topology, Phys. Rev. Lett. **106**, 058101 (2011).

[11] W. L. Shew, H. Yang, S. Yu, R. Roy, and D. Plenz, Information capacity and transmission are maximized in balanced cortical networks with neuronal avalanches, J. Neurosci. **31**, 55 (2011).

[12] W. L. Shew, H. Yang, T. Petermann, R. Roy, and D. Plenz, Neuronal avalanches imply maximum dynamic range in cortical networks at criticality, J. Neurosci. **29**, 15595 (2009).

[13] P. Bak, C. Tang, and K. Wiesenfeld, Self-organized criticality, Phys. Rev. A **38**, 364 (1988).

[14] B. Drossel and F. Schwabl, Self-Organized Critical Forest-Fire Model, Phys. Rev. Lett. **69**, 1629 (1992).

[15] P. Bak and K. Sneppen, Punctuated Equilibrium and Criticality in a Simple Model of Evolution, Phys. Rev. Lett. **71**, 4083 (1993).
[16] S. Bornholdt and T. Röhl, Self-organized critical neural networks, Phys. Rev. E **67**, 066118 (2003).
[17] Z. Olami, Hans Jacob S. Feder, and K. Christensen, Self-Organized Criticality in a Continuous, Nonconservative Cellular Automaton Modeling Earthquakes, Phys. Rev. Lett. **68**, 1244 (1992).
[18] C. W. Eurich, J. M. Herrmann, and U. A. Ernst, Finite-size effects of avalanche dynamics, Phys. Rev. E **66**, 066137 (2002).
[19] A. Levina, J. M. Herrmann, and T. Geisel, Dynamical synapses causing self-organized criticality in neural networks, Nat. Phys. **3**, 857 (2007).
[20] C. Meisel and T. Gross, Adaptive self-organization in a realistic neural network model, Phys. Rev. E **80**, 061917 (2009).
[21] V. Hernandez-Urbina and J. M. Herrmann, Self-organised criticality via retro-synaptic signals, Front. Phys. **4**, 54 (2017).
[22] L. de Arcangelis, C. Perrone-Capano, and H. J. Herrmann, Self-Organized Criticality Model for Brain Plasticity, Phys. Rev. Lett. **96**, 028107 (2006).
[23] G. L. Pellegrini, L. de Arcangelis, H. J. Herrmann, and C. Perrone-Capano, Activity-dependent neural network model on scale-free networks, Phys. Rev. E **76**, 016107 (2007).
[24] L. de Arcangelis and H. J. Herrmann, Learning as a phenomenon occurring in a critical state, Proc. Natl. Acad. Sci. U.S.A. **107**, 3977 (2010).
[25] M. O. Magnasco, O. Piro, and G. A. Cecchi, Self-Tuned Critical Anti-Hebbian Networks, Phys. Rev. Lett. **102**, 258102 (2009).
[26] W. L. Shew and D. Plenz, The functional benefits of criticality in the cortex, Neuroscientist **19**, 88 (2013).
[27] D. Marković and C. Gros, Power laws and self-organized criticality in theory and nature, Phys. Rep. **536**, 41 (2014).
[28] D. Plenz and E. Niebur, *Criticality in Neural Systems* (Wiley-Blackwell, Hoboken, NJ, 2014).
[29] J. Hesse and T. Gross, Self-organized criticality as a fundamental property of neural systems, Front. Syst. Neurosci. **8**, 166 (2014).
[30] V. Hernandez-Urbina and J. Michael Herrmann, Neuronal avalanches in complex networks, Cogent Phys. **3**, 1150408 (2016).
[31] R. Zeraati, V. Priesemann, and A. Levina, Self-organization toward criticality by synaptic plasticity, arXiv:2010.07888.
[32] M. Rybarsch and S. Bornholdt, Binary threshold networks as a natural null model for biological networks, Phys. Rev. E **86**, 026114 (2012).
[33] M. Rybarsch and S. Bornholdt, Avalanches in self-organized critical neural networks: A Minimal model for the neural SOC universality class, PloS One **9**, e93090 (2014).
[34] Y. Yada, T. Mita, A. Sanada, R. Yano, R. Kanzaki, D. J. Bakkum, A. Hierlemann, and H. Takahashi, Development of neural population activity toward self-organized criticality, Neuroscience **343**, 55 (2017).
[35] M. P. Mattson and S. Kater, Excitatory and inhibitory neurotransmitters in the generation and degeneration of hippocampal neuroarchitecture, Brain Res. **478**, 337 (1989).
[36] M. Mattson, P. Dou, and S. Kater, Outgrowth-regulating actions of glutamate in isolated hippocampal pyramidal neurons, J. Neurosci. **8**, 2087 (1988).
[37] P. Haydon, D. McCobb, and S. Kater, The regulation of neurite outgrowth, growth cone motility, and electrical synaptogenesis by serotonin, Dev. Neurobiol. **18**, 197 (1987).
[38] C. S. Cohan and S. B. Kater, Suppression of neurite elongation and growth cone mentality by electrical activity, Science **232**, 1638 (1986).
[39] R. D. Fields, E. A. Neale, and P. G. Nelson, Effects of patterned electrical activity on neurite outgrowth from mouse sensory neurons, J. Neurosci. **10**, 2950 (1990).
[40] F. Van Huizen and H. Romijn, Tetrodotoxin enhances initial neurite outgrowth from fetal rat cerebral cortex cells in vitro, Brain Res. **408**, 271 (1987).
[41] M. Fauth and C. Tetzlaff, Opposing effects of neuronal activity on structural plasticity, Front. Neuroanat. **10**, 75 (2016).
[42] M. Butz, A. Van Ooyen, and F. Wörgötter, A model for cortical rewiring following deafferentation and focal stroke, Front. Comput. Neurosci. **3**, 10 (2009).
[43] F. Y. Kalle Kossio, S. Goedeke, B. van den Akker, B. Ibarz, and R.-M. Memmesheimer, Growing Critical: Self-Organized Criticality in a Developing Neural System, Phys. Rev. Lett. **121**, 058301 (2018).
[44] J. Zierenberg, J. Wilting, and V. Priesemann, Homeostatic Plasticity and External Input Shape Neural Network Dynamics, Phys. Rev. X **8**, 031018 (2018).
[45] H. Markram, M. Toledo-Rodriguez, Y. Wang, A. Gupta, G. Silberberg, and C. Wu, Interneurons of the neocortical inhibitory system, Nat. Rev. Neurosci. **5**, 793 (2004).
[46] P. Rakic, J.-P. Bourgeois, and P. S. Goldman-Rakic, Synaptic development of the cerebral cortex: Implications for learning, memory, and mental illness, Prog. Brain Res. **102**, 227 (1994).
[47] L. P. Spear, Adolescent neurodevelopment, J. Adolescent Health **52**, S7 (2013).
[48] S.-S. Poil, R. Hardstone, H. D. Mansvelder, and K. Linkenkaer-Hansen, Critical-state dynamics of avalanches and oscillations jointly emerge from balanced excitation/inhibition in neuronal networks, J. Neurosci. **32**, 9817 (2012).
[49] J. Touboul and A. Destexhe, Can power-law scaling and neuronal avalanches arise from stochastic dynamics? PloS One **5**, e8982 (2010).
[50] H. E. Stanley, *Phase Transitions and Critical Phenomena* (Clarendon Press, Oxford, 1971).
[51] C. Haldeman and J. M. Beggs, Critical Branching Captures Activity in Living Neural Networks and Maximizes the Number of Metastable States, Phys. Rev. Lett. **94**, 058101 (2005).
[52] J. P. Sethna, K. A. Dahmen, and C. R. Myers, Crackling noise, Nature **410**, 242 (2001).
[53] A. Clauset, C. R. Shalizi, and M. E. Newman, Power-law distributions in empirical data, SIAM Rev. **51**, 661 (2009).
[54] F. Lombardi, H. Herrmann, C. Perrone-Capano, D. Plenz, and L. De Arcangelis, Balance Between Excitation and Inhibition Controls the Temporal Organization of Neuronal Avalanches, Phys. Rev. Lett. **108**, 228703 (2012).

A toy model for brain criticality: self-organized excitation/inhibition ratio and the role of network clustering

Lorenz Baumgarten and Stefan Bornholdt. “A toy model for brain criticality: self-organized excitation/inhibition ratio and the role of network clustering”. In: *arXiv preprint arXiv:2202.03330* (2022).

A toy model for brain criticality: self-organized excitation/inhibition ratio and the role of network clustering

Lorenz Baumgarten* and Stefan Bornholdt†

Institut für Theoretische Physik, Universität Bremen, 28759 Bremen, Germany
(Dated: February 8, 2022)

The critical brain hypothesis receives increasing support from recent experimental results. It postulates that the brain is at a critical point between an ordered and a chaotic regime, sometimes referred to as the "edge of chaos." Another central observation of neuroscience is the principle of excitation-inhibition balance: Certain brain networks exhibit a remarkably constant ratio between excitation and inhibition. When this balance is perturbed, the network shifts away from the critical point, as may for example happen during epileptic seizures. However, it is as of yet unclear what mechanisms balance the neural dynamics towards this excitation-inhibition ratio that ensures critical brain dynamics.

Here we introduce a simple yet biologically plausible toy model of a self-organized critical neural network with a self-organizing excitation to inhibition ratio.

The model only requires a neuron to have local information of its own recent activity and changes connections between neurons accordingly. We find that the network evolves to a state characterized by avalanche distributions following universal scaling laws typical of criticality, and to a specific excitation to inhibition ratio. The model connects the two questions of brain criticality and of a specific excitation/inhibition balance observed in the brain to a common origin or mechanism. From the perspective of the statistical mechanics of such networks, the model uses the excitation/inhibition ratio as control parameter of a phase transition, which enables criticality at arbitrary high connectivities. We find that network clustering plays a crucial role for this phase transition to occur.

Statistical Mechanics has been a long time companion to neuroscience. Decades ago, it played a central role in demonstrating how memory and computation can emerge from a network of neurons and thereby laid out the foundations of a theory of neural computation. Today, where signs of dynamical criticality emerge from neurophysiological data, statistical mechanics can, quite similarly, provide elements towards a theory of neural criticality. Statistical mechanics has been developed as a toolkit in physics for modeling interacting many particle systems by means of maximally reductionist models. Magnetic atoms that align to each other are represented as purely binary variables (with states up or down). For example, the iconic Ising Model [1, 2] almost looks like a toy model. Nevertheless, such models often make predictions about phase transitions of matter that match experimental observations with startling accuracy.

In the same reductionist approach, networks of neurons can be modeled by representing the neuron's activity as either on (1) or off (0), dropping most biological detail, in order to study dynamical mechanisms and phase transitions on the systemic level of neural networks. Such a simplified neural network model is surprisingly similar to disordered magnetic glasses, or Ising spin glasses, as has been pointed out by Hopfield in his seminal paper [3]. He formulated the idea of associative memory: storing memories of patterns as states in the energy landscape of a modified model of a magnetic spin glass. This paper initiated a field of statistical mechanics of neural networks and the theory of neural computation [4, 5]. Its success was based on its central idea that artificial neural networks based on symmetric synaptic links, which are nonsense from the biological viewpoint, enjoy full access to the tools of equilibrium statistical mechanics and spin glass physics [6–11]. It allowed to calculate the memory capacity of neural networks and to characterize the phase transition between order vs. chaos—or memory vs. forgetting—in great detail.

Explorations into the more realistic asymmetric neural networks turned out to be more difficult. Analytical results were mainly achieved in the sparsely asymmetric limit where asymmetry and loops do not fully destroy the energy landscape picture of spin glass physics [12–19]. In addition, numerical studies of random neural networks with fully asymmetric links exhibited interesting complex dynamics with an order-chaos phase transition [20–23]. A similar class of networks, random automata networks or random Boolean networks, originally motivated by the idea that gene regulation networks in living cells determine their cell type by means of a dynamical attractor [24, 25], added to this phenomenology. They exhibit a similar order-chaos phase transition and their dynamics is characterized by fixed points and periodic attractors and interesting properties near criticality [26–36]. A popular subset of random Boolean networks, the so-called random threshold networks, in fact map onto random neural networks with binary states and links [37–42]. The prominent dynamical feature of these networks is the transition between a chaotic regime at higher connectivities and a quiescent regime for lower connectivity,

* lbaumgarten@itp.uni-bremen.de
† bornholdt@itp.uni-bremen.de

divided by a critical point in the average connectivity, often at or around an average degree $K = 2$ for random Boolean networks and for neural networks with zero threshold.

Statistical mechanics has thus created a fundamental understanding of critical dynamics in networks. This has been of renewed interest for neuroscience since Beggs and Plenz [43] discovered power-law distributed activity avalanche profiles suggesting that the brain neural networks may also be poised at a critical point. Subsequent studies have produced increasingly convincing evidence for brain criticality, in the form of more critical neuronal avalanche size and duration distributions [44–65] with scale-invariant profiles [50, 57, 61, 65, 66]. One popular explanation why brains may be poised at a critical point is that criticality has been shown to optimize information processing tasks in certain model systems [57, 67–69]. We would like to point out here an alternative hypothesis, which might be the simplest after Occam's razor: The brain must function away from both chaos and quiescence, regardless of criticality. With phase transitions, statistical physics provides the opportunity to stabilize a system in an intermediate region via tuning to the critical point, where criticality itself is not the goal, but only the tool.

Many numerical models have been developed to describe criticality in neural networks, starting with simple critical branching models [67, 70], integrate-and-fire models [46, 56, 71–77], or models using other types of neurons [78] that try to reproduce the observed critical behavior via finely-tuned or realistic parameters.

There is a host of papers that go beyond replicating the critical behavior to presenting algorithms that lead the network to self-organized criticality (SOC), often using spike timing dependent plasticity [79–82], synaptic depression [83–88], Hebbian or anti-Hebbian learning [89–91], axonal and dendritic outgrowth [48, 92, 93] or combinations of these or other methods [94–99]—although many of these models still require manually fine-tuned parameters, see [100] for a review. Recent models also combine SOC with learning [101–103].

Some of these SOC papers acknowledge the importance of a balance between excitation and inhibition in a network for criticality [73, 74, 78, 89], as has also been observed experimentally [43]. A balance between excitation and inhibition in brains had already been theoretically assumed [104–108] and experimentally shown [109–111], see also [112] for a review, outside the context of criticality. The importance of the balance between excitation and inhibition can also be seen by the ratio of excitatory to inhibitory nodes being constant, roughly 4:1, among different cortical regions, species, and stages of development [113].

Self-organized criticality in adaptive Boolean and neural networks has already been established [39, 114–117] before Beggs and Plenz's seminal paper discovering signs of criticality in the brain [43]. Models bridging the gap between statistical mechanics and neuroscience have subsequently been developed [118], e.g., by combining statistical mechanics SOC models [39] with neural network adaptation mechanisms such as STDP [79]. These models self-organize to a connectivity $K = 2$ which is the critical value for networks with excitation/inhibition balance 1:1—the common value historically used in statistical mechanics models of random networks. As criticality in brains is not determined by the average degree (which in any case is much larger than $K = 2$), but instead by the balance between excitation and inhibition, it is an interesting question if these statistical mechanics SOC models also function at high connectivities and with excitation/inhibition balance as the control parameter. Both ingredients are present in some of the neuroscience models mentioned above; however, these contain considerable complexity in order to faithfully model real neural networks. Here, we want to provide a minimal, yet biologically plausible adaptive neural network model with as few parameters as possible that can nonetheless self-organize to a critical point at biologically relevant high average degrees and is also hopefully simple enough to allow it to be studied analytically using the methods of statistical mechanics.

Our model is a simple Markovian threshold network in which historically criticality has only been researched at the critical point at low connectivity $K = 2$ [119]. Based on our recent observation of the existence of K-independent critical points in such systems at high connectivities [120, 121], but dependent on the ratio of excitatory to inhibitory connections, we introduce an algorithm that tunes towards such a critical point using only locally available information. We show that this algorithm produces high-degree critical networks with specific excitation to inhibition ratios in a wide area of parameter space. Afterwards, we present an extension of the model which more closely resembles biological networks, using a constraint on the maximum number of incoming connections per node, as well as a refractory period after firing, and show that this extension also produces criticality. The property of our algorithm to be independent of implementation details points to a universality of the underlying mechanism, which suggests that such an algorithm could be used in a variety of networks of all levels of complexity.

ALGORITHM

We start with a collection of N neurons, whose states take Boolean values, randomly placed in a two-dimensional space with periodic boundary conditions. The probability of a neuron i being active at time step

$t+1$ is given by

$$P[\sigma_i(t+1)=1] = \frac{1}{1+\exp\left[-2\beta\left(f_i(t)-0.5\right)\right]}$$
$$\text{with } f_i = \sum_{j=1}^{N} c_{ij}(t)\sigma_j(t)\nu(t),$$

where $\sigma_i \in \{0,1\}$ is the state of neuron i, $f_i(t)$ is the neuron i's incoming signal, c is the adjacency matrix (with $c_{ij} \in \{0,1\}$), and $\nu_j = \pm 1$ determines whether a neuron j is excitatory or inhibitory. All neurons are updated synchronously. Initially, all neurons are unconnected and inactive, i.e., $c_{ij} = 0$ and $\sigma_i = 0$. The noise introduced via the inverse temperature β is therefore necessary for the network to escape this initial inactive state. Unless stated otherwise, we pick $\beta = 10$, which accomplishes this within reasonable time and does not affect the network dynamics much otherwise.

The algorithm tuning towards criticality adds and removes connections as follows:

- Every t_r time steps randomly select a neuron i.
- If the neuron i has been continuously active or continuously inactive during the last t_a time steps, it gains an incoming link from another neuron $j \neq i$ that is inhibitory or excitatory, respectively. A neuron j that does not have any outgoing connections can also be chosen, as such a neuron is effectively neither inhibitory nor excitatory yet.
  The neuron j is chosen as the nearest eligible neuron without a connection to i.
- If the neuron i has been neither exclusively active or inactive during the last t_a time steps, i.e., if it changed its state, it instead loses its longest incoming link.
- If a connection was created originating from a neuron j without any outgoing links, that neuron's identity is then set to inhibitory or excitatory depending on whether neuron i had been continuously active or inactive.

Unless stated otherwise, we choose $t_\mathrm{r} = 1$. As the algorithm starts with a connectionless network, the identities, i.e., excitatory/inhibitory, of all neurons are determined dynamically during rewiring. This algorithm will initially add excitatory links until the noise creates a stable nonzero activity, similar to real developing networks [122].

If connections are formed and removed randomly, similar to the model we studied in [123], instead of creating connections to the nearest eligible neuron and removing the longest connections as described above, the network will be tuned to the well-known critical point at average degree $K = 2$.

By keeping connections as short as possible, we can escape this $K = 2$ state and tune towards criticality at higher average degrees. This idea is inspired by our finding in [124] that at high degrees a network can be kept in a low-activity state with sensitivity $\lambda \approx 1$ if the network has a high clustering coefficient and can also be rationalized biologically by short connections between neurons being more common than long connections. The sensitivity λ is our first indicator for criticality. It is defined as the average number of neurons that will have a different state at time step $t+1$ if a neuron i's state is inverted in time step t, i.e., if i is active it is turned inactive and vice versa, than they would otherwise have had [125, 126]. If $\lambda > 1$, perturbations to the network will on average increase, or they will decrease for $\lambda < 1$. The border between these two regimes, $\lambda = 1$, is the critical point.

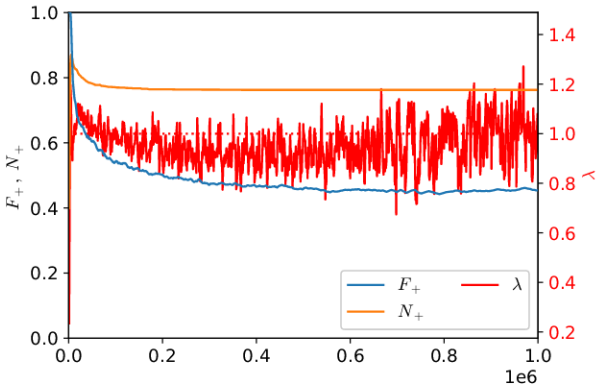


FIG. 1. A typical run of the algorithm for $N = 2000$ and $t_\mathrm{a} = 10^3$. The upper diagram shows the sensitivity λ (right axis), the fraction of excitatory connections F_+, and the fraction of excitatory nodes N_+ (left axis), as a function of t. The dotted red line shows the critical sensitivity $\lambda = 1$.

We find that the algorithm maintains a sensitivity λ near one while steadily increasing the average degree K. A typical run of the algorithm is shown in Figure 1. Increasing the considered time frame t_a or decreasing the inverse temperature β leads to a slower increase in K—or no increase at all if nodes are likely to have changed their states within the time window through noise alone—but still produces sensitivities near one if t_a and β are not too large or too small.

CRITICALITY

Figure 1 shows that the algorithm does not tune precisely to $\lambda = 1$, but since $\lambda = 1$ is merely an indicator of criticality and can be inaccurate for clustered networks, this need not discourage us. Additionally, Figure 2 shows that the sensitivity λ does not stray too far from one in a large parameter space.

To further test for criticality, we study activity avalanches in our networks. To measure avalanches, we

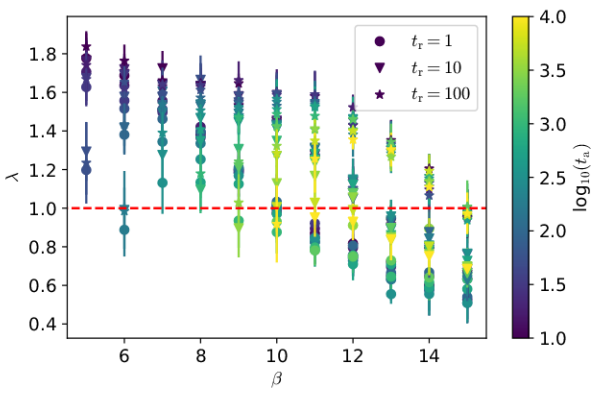


FIG. 2. Sensitivity λ for different values of the noise β, the window of states considered for rewiring t_a and the time between rewiring steps t_r ($N = 2000$, at $K = 45$). The red dotted line shows the critical value of $\lambda = 1$. Each point is the average of 10 simulations. Parameter combinations in which the network's average degree K did not increase within $10^6 \cdot t_r$ time steps before reaching $K = 45$ in any of the 10 simulations, i.e., if a network with $K = 45$ could not reliably be produced within reasonable time, are not shown.

first let the algorithm run until a target value of K is reached. Here, we pick $K = 45 \approx \sqrt{N}$ for $N = 2000$. Different choices of K and N do not affect the results much, as long as $K \ll N$ and t_r is changed accordingly. The alternative to stopping the algorithm at a fixed K would be to wait for it to arrive at a stationary point, which takes a long time and also only yields average degrees of the order of N. Once the target value of K is reached, we let the network's dynamics continue without any further rewiring and without noise. We then copy the network and flip a node i in the copy at time t_0. Next we let both networks run in parallel and measure the time it takes for them to arrive at the same state at a time $t' > t_0$. This is the avalanche duration $T = t' - t_0$. The sum of Hamming distances between the two networks during the avalanche is the avalanche size S. The measurement is repeated for every node i in a network and for many different networks.

It is of course possible that the manipulated copy ends up in a different attractor than the original, that it ends up in the same attractor but with a time shift compared to the original, or that the two networks only reach the same state after a large amount of time steps (we stop the measurement after 10^4 time steps). Since distance and duration are more difficult to define in these cases, we do not use them for our measurements. Our measurements show that this is a relatively rare case – depending on parameters at most for about 30% of avalanches.

At a critical point, the avalanche size S and avalanche duration T distributions, $P(S)$ and $P(T)$, as well as average avalanche size as a function of avalanche duration should follow power laws [127]

$$P(S) \propto S^{-\tau} \tag{1}$$
$$P(T) \propto T^{-\alpha} \tag{2}$$
$$\langle S \rangle (T) \propto T^{\gamma}, \tag{3}$$
$$\text{with } \frac{\alpha - 1}{\tau - 1} = \gamma. \tag{4}$$

Further, the avalanche profiles, i.e., Hamming distance to the unperturbed network as a function of time, should collapse onto each other if time is rescaled by the avalanche duration and the Hamming distance d_H is rescaled by $T^{\gamma-1}$. The three power-laws and the avalanche collapse are shown in Figures 3 and 4 for $K = 45$. We find approximate power-laws and a sufficient collapse of avalanches, indicating criticality. The values for τ, α, and γ indicated by the blue, dashed lines in Figure 3 are

$$\tau \approx 1.8767 \pm 0.0003$$
$$\alpha \approx 2.6916 \pm 0.0006$$
$$\gamma \approx 1.80 \pm 0.03,$$
$$\text{and } \frac{\alpha - 1}{\tau - 1} = 1.9296 \pm 0.0004 \approx \gamma.$$

These values have been fitted using the estimator for discrete integer variables described in [128], and the errors given are those resulting from this fitting method. We have verified that our algorithm also produces approximate power-laws following the scaling relation (4) and showing a data collapse for most of the parameter space shown in Figure 2. The exponents found vary between $\tau \approx 1.6$ and $\tau \approx 2.6$ as well as $\alpha \approx 2.2$ and $\alpha \approx 3.4$.

Lastly, we study the critical point's vicinity in the F_+ space.

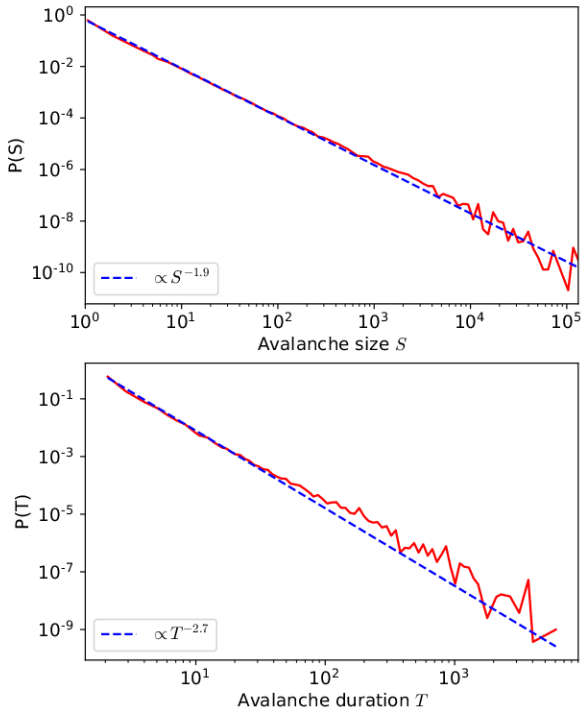


FIG. 3. Logarithmically binned avalanche size (upper diagram) and duration (lower diagram) distributions for $K = 45$, $N = 2000$, and $t_a = 10^3$. The dashed lines show a power-law fit.

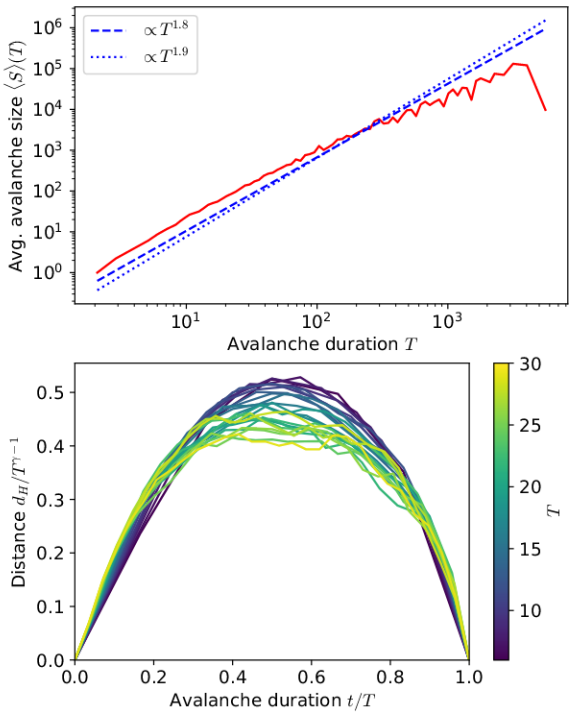


FIG. 4. Average avalanche size as a function of avalanche duration (upper diagram) and collapse of avalanche profiles for avalanches of duration 6–30 (lower diagram) for the avalanches shown in Figure 3. The dashed line shows a power-law fit, and the dotted line shows a power-law with the theoretical value of γ given by equation (4).

To observe the effect of the fraction of excitatory connections F_+ on criticality, we use the following procedure:

- Evolve a network up to an initial average degree K_{ini} using the previously described algorithm
- Perturb the network by either increasing or decreasing F_+. In order to do this, repeat the following steps until the desired value of F_+ is reached:
  - Pick a random neuron
  - If F_+ shall be increased/decreased and the neuron has an incoming inhibitory/excitatory link:
    - Remove the farthest incoming inhibitory/excitatory link
    - Form a new incoming excitatory/inhibitory link from the nearest eligible neuron.
  - Otherwise, do nothing.
- Resume the previous tuning algorithm until a final average degree K_{fin} is reached.

The algorithm quickly returns the network to a critical state, regardless of F_+ at K_{ini}, and we can observe the sensitivity and frozen components on the way the algorithm takes from the perturbed state to the critical state, as shown in Figure 5. Note that the perturbation changes F_+ but conserves all neurons' in-degrees.

Figure 5 illustrates the functionality of our algorithm. For F_+ below the critical point, the frozen off component is larger than the frozen on component, meaning that more nodes are permanently off than on, causing the algorithm to create more excitatory links and thereby increasing F_+. The opposite can be seen for F_+ above the critical point. The frozen on component is still larger than the frozen off component in the region of F_+ the algorithm tunes to. From this, one might expect that F_+ would be further decreased here; however, since due to the low total frozen component, many connections are also being removed, if on average more inhibitory connections are being removed than excitatory ones—due to inhibitory connections on average being longer because of the higher out-degree of inhibitory nodes—, F_+ must

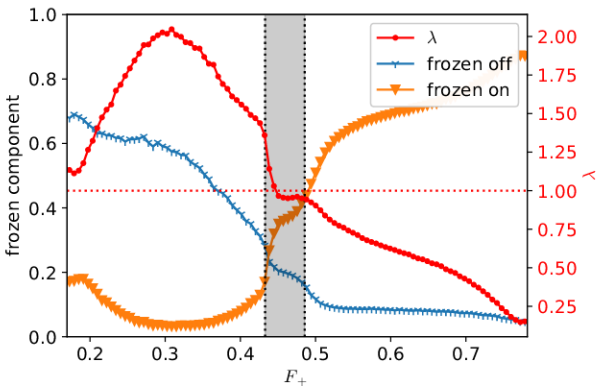


FIG. 5. Average frozen on/off component and sensitivity λ (right axis) for networks moving from a state perturbed in the F_+ space to a critical state ($N = 2000$, $K = 45$, $t_a = 1000$) as a function of the ratio of excitatory links F_+. Shown are the averages of 100 networks being perturbed to either side, and the red dotted line shows the critical value of λ. The frozen on/off component is the fraction of nodes at any point in time that was exclusively on/off during the last t_a time steps. F_+ was perturbed by ± 0.3 at $K_{\text{ini}} = 25$ and the simulation was ended at $K_{\text{fin}} = 45$. During the simulation, F_+ moves from the outer boundaries of the diagram towards the grey area, which shows the area between maximum and minimum values of F_+ that networks reached at K_{fin}, i.e., the critical area of F_+ the algorithm tunes towards. The exact value of F_+ that is reached is slightly dependent on initial conditions and F_+ also fluctuates slightly even after reaching the critical point.

not necessarily decrease within the grey region.
We also see that λ increases to values significantly above one for F_+ below the critical point and significantly below one above the critical point, meaning that it is possible to tune through the critical point, which is another indicator of criticality.
Our simulations also show that critical avalanche profiles can still be achieved if the rewiring rule removing connections of flickering nodes is omitted; however, we then lose the ability to tune through the critical point, and the network's ability to return to the critical point after perturbation is diminished because the only way for the network to change its F_+ is by adding new links. If we for example constrain the maximum number of links a neuron has, as we will discuss in the following section, returning to a critical state after a perturbation at a saturated degree would therefore be impossible for the network without the ability to remove links.

BIOLOGICALLY MOTIVATED MODEL EXTENSIONS

So far we have kept our model as minimal as possible in order to study the pure mechanisms of self-organized criticality and excitation/inhibiton regulation. We now want to demonstrate that the model can be easily expanded to more closely align with properties of realistic neural networks—namely the physiological limits on connectivity and the refractory nature of real neurons—without losing its ability to self-organize to a critical state.

Let us start with considering constraints on the resulting average degree K. As mentioned before, in earlier, non-spatial variations of our model, the algorithm would tune to the critical point $K = 2$ [123]; however, the algorithm presented here does not tune K, but instead the ratio of excitation to inhibition via F_+ and N_+. Therefore, to produce criticality, the connection-removing and connection-producing rules of our algorithm need not balance out, and thus K rises. To keep the average degree from increasing almost indefinitely, we add an additional rule to the model: A node's number of incoming connections cannot exceed a limit K_{max}. Such a rule can easily be motivated biologically. Firstly, a brain has reason to be parsimonious with its resources and therefore limit synapses if possible. Secondly, in a biological network, a neuron can simply not have an infinite amount of connections due to spatial restrictions.

The second natural expansion of our model, a refractory period of neuronal activity, has two beneficial side effects. In the base model, the average activity of the network is pushed towards 50 %, as connections are added to push nodes away from being permanently active or inactive. This is of course unrealistic for brains, as neurons—as long as they are not part of a spiking avalanche—tend to be inactive apart from occasional spiking due to background noise. This is also reflected in our initial definition of avalanches. Our avalanches are not avalanches of activity as is common, but instead of distance to an unperturbed comparison network. Both of these points are ameliorated by introducing a refractory period t_{ref} to our model as follows:

Nodes cannot be active for t_{ref} time steps after being active for one time step. This change is inspired by biology, and nodes can now either be considered as single, primitive spiking neurons or as clusters of neurons which "tire out" and need to recover after spike trains.

We choose a refractory period of $t_{\text{ref}} = 2$ as our observations have shown that $t_{\text{ref}} = 1$ simply produces clusters of two alternatingly blinking parts enabling sustained activity. Any refractory period $t_{\text{ref}} > 1$ does not produce this effect, and therefore we choose the lowest possible value for simplicity's sake. Our algorithm then produces a network whose default state in the absense of activating noise is inactive—although small clusters of sustained activity can still occur, but these are not the norm and often collapse under noise.

The refractory period also requires an adjustment of our rewiring rules as nodes cannot be permanently active anymore: Instead, nodes gain an incoming inhibitory connection if they have ever been active during the last t_{a} time steps and connections are never removed. As previously discussed, this diminishes the model's ability to return to criticality after perturbation, but it is sufficient

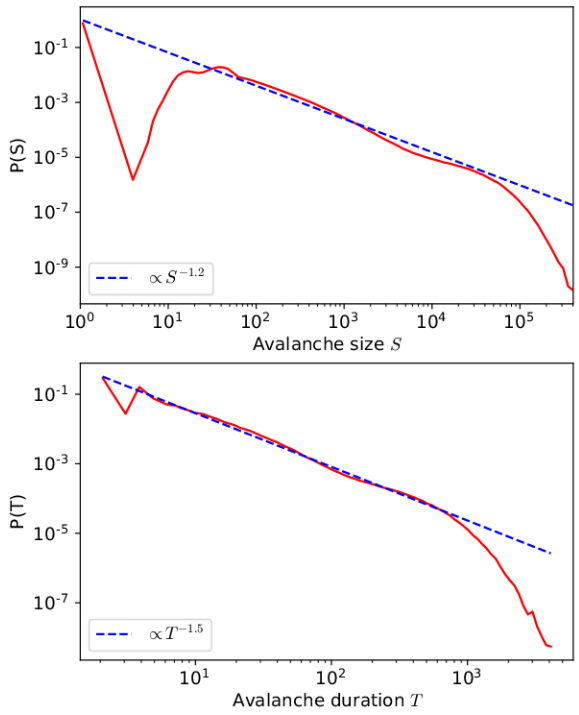


FIG. 6. Logarithmically binned avalanche size (upper diagram) and duration (lower diagram) distributions of the model with refractory periods for $K = 45$, $N = 2000$, $t_a = 10^2$, and $t_r = 10^2$. The dashed lines show a power-law fit.

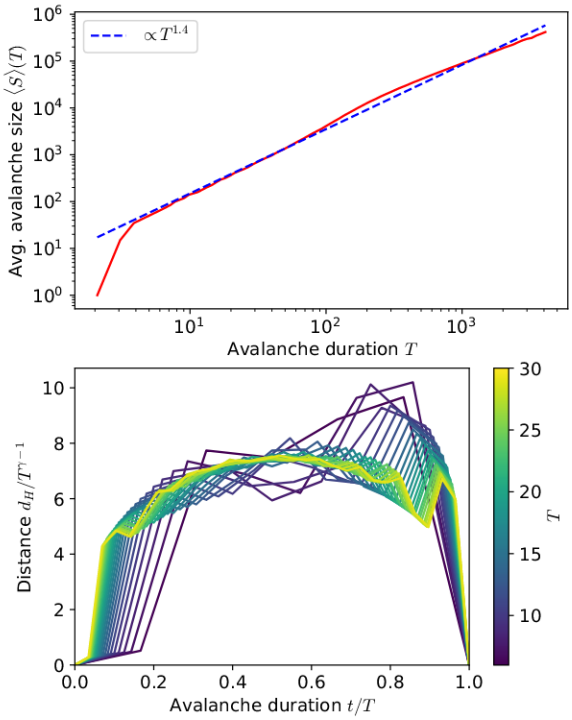


FIG. 7. Average avalanche size as a function of avalanche duration (upper diagram) and collapse of avalanche profiles for avalanches of duration 6–30 (lower diagram) for the avalanches shown in Figure 3. The dashed line shows a power-law fit.

to show that criticality can be achieved with the model extensions we want to present here. Of course, the rule for removing connections could still be implemented by setting average activity thresholds, but we were unwilling to add more parameters to our model.

Additionally, the sensitivity is no longer an acceptable indicator of criticality in this case because in the inactive state, the sensitivity is simply the average number of excitatory connections per node which is significantly larger than one. Only once an avalanche starts, does the refractory period prevent nodes from being active consecutively, and it therefore effectively lowers the sensitivity during an avalanche. This again leads to criticality as shown via power-laws and avalanche collapses in Figures 6 and 7. When testing other combinations of t_r, t_a, and β, we found that these parameters still need not be fine-tuned for this model to self-organize to a critical point. The model will reach a critical, high-degree state as long as all of these parameters are sufficiently large.

Since this network operates near a completely inactive state, we can now use a simpler, more intuitive definition of avalanches than before. We start with the network being completely inactive and then activate one node. The avalanche is then simulated until the network returns to the completely inactive state and its size is measured as the number of nodes being activated, where a node can also be counted multiple times if it has been active multiple times during the avalanche. Therefore, the avalanches here are true activity avalanches and no longer require a comparison to an unperturbed network.

In Figure 7, we can see that the first time step after a node has been activated usually causes a higher activity than subsequent time steps because an excitatory node will, in an inactive network, activate all of its neighbors, but after the first time step, an active node's neighbors may be in their refractory period, thereby reducing the number of nodes that can be activated. This also causes the initial dip in the avalanche size and duration distributions seen in Figure 6. The last time step before the network returns to inactivity also often shows a high activity. This is because, when a large number of nodes is activated, it is likely that a proportionately large number of inhibiting nodes is activated. Since inhibiting nodes have on average significantly more outgoing connections

than excitatory nodes, although more excitatory nodes may be activated, a node in proximity to this activation is likely connected to many of the active inhibitory nodes but only a few of the active excitatory nodes and may therefore not activate. This leads avalanches to often end after a time step with high activity. These high activity steps and subsequent dropoffs—albeit not to complete inactivity—also occur during avalanches but are averaged out in the avalanche profile.

CONCLUSION

In this paper we develop a simple self-organized critical neural network model to tune towards the newly discovered critical point at high average degrees discussed in [120, 121]. This critical point, unlike the previously known critical point at average degree $K = 2$, is nearly independent of the average degree and instead depends on the balance between excitation and inhibition. We have thus developed a simple algorithm that produces criticality in neural networks at high average degrees, using only local information and very few parameters. The algorithm differs from early physics papers, which studied the aforementioned critical point at a constant excitation/inhibition ratio 1:1 [26, 39, 41, 119], and also from neuroscience models, which—due to their closeness to biological reality—are more complex than our model.

The core idea behind our model is to move the network away from quiescence by adding excitatory/inhibitory connections to permanently inactive/active neurons, and to move the network away from chaos by removing connections from neurons switching their states. The addition of two-dimensional space and creating connections to the nearest neurons generates network clustering which ultimately allows the model to adjust to the critical point with high average degree.

As a result, the model exhibits power-law shaped distributions of activity avalanche sizes and durations, which obey universal scaling relationships and can be collapsed onto each other, the criteria for criticality required by dynamical scaling theory. The model allows to tune through the critical point, from a supercritical to a subcritical regime, by varying the ratio of excitatory to inhibitory connections in the network. We have also confirmed that the algorithm produces the observed scaling for a large part of parameter space. Variants of the model that more closely resemble properties of biological networks can be easily built, as we demonstrated with an extended version that constrains the maximum number of connections of a node, and by introducing a refractory period as a simple representation of firing neurons. This extension also resulted in networks in a critical state, further indicating that the precise implementation of our model is irrelevant for the emergence of criticality.

There already exists a host of neuroscience papers modeling self-organized criticality, however, the commonly complex nature and/or high number of parameters, see for example [80, 95, 99, 103], of the models and the breadth of possible implementations used—such as synaptic depression [82, 83, 85, 87, 99], Hebbian or anti-Hebbian learning [89–91], STDP [79, 81, 95, 103], or axonal outgrowth [48, 92, 93]—makes it difficult to gauge which properties of the models are essential for self-organized criticality, whereas the model presented here has been trimmed down to its minimal possible version. For example, nearly all neuroscience models of self-organized criticality utilize integrate-and-fire neurons or other parameter-heavy biologically realistic neuron behaviors [48, 79, 81–86, 92, 95–98, 129], but these, as well as their exact implementation, seem to merely be biological flavor not needed for criticality, as shown by our model. Many of these models opt for less realistic network structures than the one we used, such as fully-connected [82, 83, 86, 89, 96, 97, 99] or Watts-Strogatz networks [81, 129, 130], and some models omit inhibition [83, 87, 92, 130] or require very specific parameters to attain self-organized criticality [81–83, 95, 98]. The previously existing models closest to the one presented here are outgrowth models [48, 92, 93] in which neurons increase or decrease their interaction ranges depending on their activity level, but these also require, in addition to multiple parameters describing the neuron behavior, manually setting a parameter, namely a neuron target activity level, for the network to self-organize to criticality.

In contrast, the minimal nature of our model allows us to isolate and observe the underlying mechanism regulating criticality. As a main observation, we find that our model requires clustering to achieve the critical state at high connectivities K, and that, while a ratio of excitatory to inhibitory nodes consolidates during growth, its exact value is not central to criticality; rather, criticality is achieved by fine-tuning the connectivity between nodes, as has been observed experimentally [131]. Further, the exact algorithm parameters—namely how often rewiring operations are performed, governed by t_r, the time window considered for the rewiring rules, t_a, the noise level β, and the number of neurons N—and the neuron activity implementation, i.e., being inactive for a refractory period after activation or not, may vary details of the resulting network's behavior, such as the critical exponents, but do not impede criticality itself, so long as the parameters are within a rather large area of the parameter space. Lastly, to the best of our knowledge, no other self-organized criticality model dynamically produces the network's ratio of excitatory to inhibitory nodes.

In addition to pinpointing the mechanisms that enable self-organized criticality, due to the model's minimal nature, the assumptions we do make about connectivity and clustering are realistic for real neural networks within the frame of our modeling approach. Since we further showed that the model can be easily extended to include more biologically relevant implementation details, such as a refractory period after firing, we hope that it can form a

useful link between underlying mechanism and more detailed models of brain criticality. An interesting question is how the known biological and biochemical processes in the brain could implement or interpret the mechanisms studied here. We further hope that the simplicity of our model may also encourage analytical follow-up studies in statistical mechanics and dynamical systems theory [132] of self-organized critical neural networks.

[1] W. Lenz, Z. Physik **21**, 613 (1920).
[2] E. Ising, Z. Phys. **31**, 253 (1925).
[3] J. J. Hopfield, Proceedings of the national academy of sciences **79**, 2554 (1982).
[4] D. J. Amit, *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press, 1989).
[5] J. Hertz, A. Krogh, and R. G. Palmer, "Introduction to the theory of neural computation," (1991).
[6] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Physical Review A **32**, 1007 (1985).
[7] W. Kinzel, in *Complex Systems — Operational Approaches in Neurobiology, Physics, and Computers*, edited by H. Haken (Springer Berlin Heidelberg, Berlin, Heidelberg, 1985) pp. 107–115.
[8] E. Gardner, EPL (Europhysics Letters) **4**, 481 (1987).
[9] M. Mézard, G. Parisi, and M. A. Virasoro, *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*, Vol. 9 (World Scientific Publishing Company, 1987).
[10] W. Kinzel, Physica Scripta **1989**, 144 (1989).
[11] A. Zippelius, Physica A: Statistical Mechanics and its Applications **194**, 471 (1993).
[12] B. Derrida, E. Gardner, and A. Zippelius, EPL (Europhysics Letters) **4**, 167 (1987).
[13] E. Gardner, Journal of Physics A: Mathematical and General **19**, L1047 (1986).
[14] E. Gardner, B. Derrida, and P. Mottishaw, Journal de physique **48**, 741 (1987).
[15] E. Gardner, Journal of physics A: Mathematical and general **21**, 257 (1988).
[16] E. Gardner and B. Derrida, Journal of Physics A: Mathematical and general **21**, 271 (1988).
[17] R. Kree, D. Widmaier, and A. Zippelius, Journal of Physics A: Mathematical and General **21**, L1181 (1988).
[18] A. Coolen, in *Handbook of biological physics*, Vol. 4 (Elsevier, 2001) pp. 553–618.
[19] A. Coolen, in *Handbook of biological physics*, Vol. 4 (Elsevier, 2001) pp. 619–684.
[20] H. Sompolinsky, A. Crisanti, and H.-J. Sommers, Physical review letters **61**, 259 (1988).
[21] B. Cessac, EPL (Europhysics Letters) **26**, 577 (1994).
[22] B. Doyon, B. Cessac, M. Quoy, and M. Samuelides, Acta biotheoretica **42**, 215 (1994).
[23] M. Stern, H. Sompolinsky, and L. Abbott, Physical Review E **90**, 062710 (2014).
[24] S. A. Kauffman, Journal of theoretical biology **22**, 437 (1969).
[25] S. A. Kauffman *et al.*, *The origins of order: Self-organization and selection in evolution* (Oxford University Press, USA, 1993).
[26] B. Derrida and Y. Pomeau, EPL (Europhysics Letters) **1**, 45 (1986).
[27] B. Derrida and G. Weisbuch, Journal de physique **47**, 1297 (1986).
[28] B. Derrida and D. Stauffer, EPL (Europhysics Letters) **2**, 739 (1986).
[29] G. Weisbuch and D. Stauffer, Journal de physique **48**, 11 (1987).
[30] J. E. S. Socolar and S. A. Kauffman, Phys. Rev. Lett. **90**, 068702 (2003).
[31] M. Aldana, S. Coppersmith, and L. P. Kadanoff, Perspectives and Problems in Nolinear Science , 23 (2003).
[32] V. Kaufman, T. Mihaljev, and B. Drossel, Physical Review E **72**, 046124 (2005).
[33] B. Drossel, Physical Review E **72**, 016110 (2005).
[34] B. Drossel, T. Mihaljev, and F. Greil, Physical review letters **94**, 088701 (2005).
[35] T. Mihaljev and B. Drossel, Physical Review E **74**, 046101 (2006).
[36] B. Drossel, Reviews of nonlinear dynamics and complexity 1 , 69 (2008).
[37] K. Kurten, Journal of Physics A: Mathematical and General **21**, L615 (1988).
[38] K. E. Kürten, Physics Letters A **129**, 157 (1988).
[39] T. Rohlf and S. Bornholdt, Physica A: Statistical Mechanics and its Applications **310**, 245 (2002).
[40] T. Rohlf, Physical Review E **78**, 066118 (2008).
[41] A. Szejka, T. Mihaljev, and B. Drossel, New Journal of Physics **10**, 063009 (2008).
[42] R.-S. Wang and R. Albert, Physical Review E **87**, 012810 (2013).
[43] J. M. Beggs and D. Plenz, Journal of Neuroscience **23**, 11167 (2003), https://www.jneurosci.org/content/23/35/11167.full.pdf.
[44] A. Mazzoni, F. D. Broccard, E. Garcia-Perez, P. Bonifazi, M. E. Ruaro, and V. Torre, PloS one **2** (2007).
[45] E. D. Gireesh and D. Plenz, Proceedings of the National Academy of Sciences **105**, 7576 (2008), https://www.pnas.org/content/105/21/7576.full.pdf.
[46] V. Pasquale, P. Massobrio, L. Bologna, M. Chiappalone, and S. Martinoia, Neuroscience **153**, 1354 (2008).
[47] T. Petermann, T. C. Thiagarajan, M. A. Lebedev, M. A. Nicolelis, D. R. Chialvo, and D. Plenz, Proceedings of the National Academy of Sciences **106**, 15921 (2009).
[48] C. Tetzlaff, S. Okujeni, U. Egert, F. Wörgötter, and M. Butz, PLoS computational biology **6** (2010).
[49] S. Yu, H. Yang, H. Nakahara, G. S. Santos, D. Nikolić, and D. Plenz, Journal of neuroscience **31**, 17514 (2011).
[50] N. Friedman, S. Ito, B. A. Brinkman, M. Shimono, R. L. DeVille, K. A. Dahmen, J. M. Beggs, and T. C. Butler, Physical review letters **108**, 208102 (2012).
[51] E. Tagliazucchi, P. Balenzuela, D. Fraiman, and D. R. Chialvo, Frontiers in physiology **3**, 15 (2012).
[52] J. Pu, H. Gong, X. Li, and Q. Luo, Scientific reports **3**, 1 (2013).
[53] V. Priesemann, M. Wibral, M. Valderrama, R. Pröpper, M. Le Van Quyen, T. Geisel, J. Triesch, D. Nikolić, and M. H. Munk, Frontiers in systems neuroscience **8**, 108 (2014).

[54] G. Scott, E. D. Fagerholm, H. Mutoh, R. Leech, D. J. Sharp, W. L. Shew, and T. Knöpfel, Journal of Neuroscience **34**, 16611 (2014).

[55] T. Bellay, A. Klaus, S. Seshadri, and D. Plenz, Elife **4**, e07224 (2015).

[56] P. Massobrio, V. Pasquale, and S. Martinoia, Scientific reports **5**, 10578 (2015).

[57] N. M. Timme, N. J. Marshall, N. Bennett, M. Ripp, E. Lautzenhiser, and J. M. Beggs, Frontiers in physiology **7**, 425 (2016).

[58] Y. Yada, T. Mita, A. Sanada, R. Yano, R. Kanzaki, D. J. Bakkum, A. Hierlemann, and H. Takahashi, Neuroscience **343**, 55 (2017).

[59] S. Yu, T. L. Ribeiro, C. Meisel, S. Chou, A. Mitz, R. Saunders, and D. Plenz, Elife **6**, e27119 (2017).

[60] Z. Ma, G. G. Turrigiano, R. Wessel, and K. B. Hengen, bioRxiv , 503243 (2018).

[61] A. Ponce-Alvarez, A. Jouary, M. Privat, G. Deco, and G. Sumbre, Neuron **100**, 1446 (2018).

[62] M. Yaghoubi, T. de Graaf, J. G. Orlandi, F. Girotto, M. A. Colicos, and J. Davidsen, Scientific reports **8**, 1 (2018).

[63] Z. Bowen, D. Winkowski, S. Seshadri, D. Plenz, and P. O. Kanold, Frontiers in systems neuroscience **13**, 45 (2019).

[64] A. J. Fontenele, N. A. de Vasconcelos, T. Feliciano, L. A. Aguiar, C. Soares-Cunha, B. Coimbra, L. Dalla Porta, S. Ribeiro, A. J. Rodrigues, N. Sousa, *et al.*, Physical review letters **122**, 208101 (2019).

[65] S. R. Miller, S. Yu, and D. Plenz, Scientific reports **9**, 1 (2019).

[66] A. Shaukat and J.-P. Thivierge, Frontiers in computational neuroscience **10**, 29 (2016).

[67] C. Haldeman and J. M. Beggs, Physical review letters **94**, 058101 (2005).

[68] W. L. Shew, H. Yang, S. Yu, R. Roy, and D. Plenz, Journal of neuroscience **31**, 55 (2011).

[69] X. Li, Q. Chen, and F. Xue, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **375**, 20160286 (2017).

[70] R. V. Williams-García, M. Moore, J. M. Beggs, and G. Ortiz, Physical Review E **90**, 062714 (2014).

[71] C. W. Eurich, J. M. Herrmann, and U. A. Ernst, Physical review E **66**, 066137 (2002).

[72] A. Levina, U. Ernst, and J. M. Herrmann, Neurocomputing **70**, 1877 (2007).

[73] S.-J. Wang, C. Hilgetag, and C. Zhou, Frontiers in computational neuroscience **5**, 30 (2011).

[74] S.-S. Poil, R. Hardstone, H. D. Mansvelder, and K. Linkenkaer-Hansen, Journal of Neuroscience **32**, 9817 (2012).

[75] L. Dalla Porta and M. Copelli, PLoS computational biology **15**, e1006924 (2019).

[76] A. Haimovici, E. Tagliazucchi, P. Balenzuela, and D. R. Chialvo, Physical review letters **110**, 178101 (2013).

[77] R. P. Rocha, L. Koçillari, S. Suweis, M. Corbetta, and A. Maritan, Scientific reports **8**, 1 (2018).

[78] M. Benayoun, J. D. Cowan, W. van Drongelen, and E. Wallace, PLoS computational biology **6** (2010).

[79] C. Meisel and T. Gross, Physical Review E **80**, 061917 (2009).

[80] M. Rubinov, O. Sporns, J.-P. Thivierge, and M. Breakspear, PLoS computational biology **7** (2011).

[81] F. P. P. Teixeira and M. Shanahan, in *2014 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2014) pp. 2383–2390.

[82] M. Khoshkhou and A. Montakhab, Frontiers in systems neuroscience **13**, 73 (2019).

[83] A. Levina, J. M. Herrmann, and T. Geisel, Nature physics **3**, 857 (2007).

[84] A. Levina, J. M. Herrmann, and T. Geisel, Physical review letters **102**, 118110 (2009).

[85] D. Millman, S. Mihalas, A. Kirkwood, and E. Niebur, Nature physics **6**, 801 (2010).

[86] W. L. Shew, W. P. Clawson, J. Pobst, Y. Karimipanah, N. C. Wright, and R. Wessel, Nature Physics **11**, 659 (2015).

[87] J. G. F. Campos, A. de Andrade Costa, M. Copelli, and O. Kinouchi, Physical Review E **95**, 042303 (2017).

[88] J. Zierenberg, J. Wilting, and V. Priesemann, Phys. Rev. X **8**, 031018 (2018).

[89] M. O. Magnasco, O. Piro, and G. A. Cecchi, Physical review letters **102**, 258102 (2009).

[90] L. de Arcangelis, C. Perrone-Capano, and H. J. Herrmann, Physical review letters **96**, 028107 (2006).

[91] G. L. Pellegrini, L. de Arcangelis, H. J. Herrmann, and C. Perrone-Capano, Physical Review E **76**, 016107 (2007).

[92] L. Abbott and R. Rohrkemper, Progress in brain research **165**, 13 (2007).

[93] F. Y. K. Kossio, S. Goedeke, B. van den Akker, B. Ibarz, and R.-M. Memmesheimer, Physical review letters **121**, 058301 (2018).

[94] M. Uhlig, A. Levina, T. Geisel, and M. Herrmann, Frontiers in computational neuroscience **7**, 87 (2013).

[95] N. Stepp, D. Plenz, and N. Srinivasa, PLoS computational biology **11** (2015).

[96] L. Brochini, A. de Andrade Costa, M. Abadi, A. C. Roque, J. Stolfi, and O. Kinouchi, Scientific reports **6**, 1 (2016).

[97] A. A. Costa, L. Brochini, and O. Kinouchi, Entropy **19**, 399 (2017).

[98] V. Hernandez-Urbina and J. M. Herrmann, Frontiers in Physics **4**, 54 (2017).

[99] M. Girardi-Schappo, L. Brochini, A. A. Costa, T. T. Carvalho, and O. Kinouchi, Physical Review Research **2**, 012042 (2020).

[100] R. Zeraati, V. Priesemann, and A. Levina, Frontiers in Physics **9** (2021), 10.3389/fphy.2021.619661.

[101] S. Scarpetta, I. Apicella, L. Minati, and A. de Candia, Physical Review E **97**, 062305 (2018).

[102] L. M. van Kessenich, M. Luković, L. De Arcangelis, and H. J. Herrmann, Physical Review E **97**, 032312 (2018).

[103] B. Del Papa, V. Priesemann, and J. Triesch, PloS one **12** (2017).

[104] M. N. Shadlen and W. T. Newsome, Current opinion in neurobiology **4**, 569 (1994).

[105] C. Van Vreeswijk and H. Sompolinsky, Science **274**, 1724 (1996).

[106] D. J. Amit and N. Brunel, Cerebral cortex (New York, NY: 1991) **7**, 237 (1997).

[107] M. N. Shadlen and W. T. Newsome, Journal of neuroscience **18**, 3870 (1998).

[108] N. Brunel, Journal of computational neuroscience **8**, 183 (2000).

[109] Y. Shu, A. Hasenstaub, and D. A. McCormick, Nature **423**, 288 (2003).

[110] M. Wehr and A. M. Zador, Nature **426**, 442 (2003).

[111] J. S. Isaacson and M. Scanziani, Neuron **72**, 231 (2011).
[112] S. Denève and C. K. Machens, Nature neuroscience **19**, 375 (2016).
[113] S. Sahara, Y. Yanagawa, D. D. M. O'Leary, and C. F. Stevens, Journal of Neuroscience **32**, 4755 (2012), https://www.jneurosci.org/content/32/14/4755.full.pdf.
[114] S. Bornholdt and T. Rohlf, Physical Review Letters **84**, 6114 (2000).
[115] S. Bornholdt and T. Röhl, Physical Review E **67**, 066118 (2003).
[116] T. Gross and B. Blasius, Journal of the Royal Society Interface **5**, 259 (2008).
[117] T. Gross and H. Sayama, *Adaptive Networks: Theory, Models and Applications* (Springer Science & Business Media, 2009).
[118] F. Droste, A.-L. Do, and T. Gross, Journal of The Royal Society Interface **10**, 20120558 (2013).
[119] K. E. Kürten, Physics Letters A **129**, 157 (1988).
[120] J. P. Neto, M. A. de Aguiar, J. A. Brum, and S. Bornholdt, arXiv preprint arXiv:1712.08816 (2017).
[121] L. Baumgarten and S. Bornholdt, Physical Review E **100**, 010301 (2019).
[122] B. Jiang, Z. J. Huang, B. Morales, and A. Kirkwood, Brain Research Reviews **50**, 126 (2005).
[123] S. Landmann, L. Baumgarten, and S. Bornholdt, Physical Review E **103**, 032304 (2021).
[124] L. Baumgarten and S. Bornholdt, Phys. Rev. E **100**, 010301 (2019).
[125] B. Luque and R. Solé, Phys. Rev. E **55**, 257 (1997).
[126] I. Shmulevich and S. Kauffman, Phys. Rev. Lett. **93**, 048701 (2004).
[127] J. P. Sethna, K. A. Dahmen, and C. R. Myers, Nature **410**, 242 (2001).
[128] A. Clauset, C. R. Shalizi, and M. E. Newman, SIAM Review **51**, 661 (2009).
[129] M. Lin and T. Chen, Physical Review E **71**, 016133 (2005).
[130] R. Pazzini, O. Kinouchi, and A. A. Costa, Phys. Rev. E **104**, 014137 (2021).
[131] N. Sukenik, O. Vinogradov, E. Weinreb, M. Segal, A. Levina, and E. Moses, Proceedings of the National Academy of Sciences **118** (2021).
[132] T. Gross, Frontiers in Neural Circuits **15**, 7 (2021).

Epidemics with asymptomatic transmission: Subcritical phase from recursive contact tracing

Lorenz Baumgarten and Stefan Bornholdt. "Epidemics with asymptomatic transmission: Subcritical phase from recursive contact tracing". In: *Physical Review E* 104.5 (2021), p. 054310.

Epidemics with asymptomatic transmission: Subcritical phase from recursive contact tracing

Lorenz Baumgarten* and Stefan Bornholdt†
Institut für Theoretische Physik, Universität Bremen, 28759 Bremen, Germany

(Received 27 August 2020; revised 7 September 2021; accepted 16 November 2021; published 29 November 2021)

The challenges presented by the COVID-19 epidemic have created a renewed interest in the development of new methods to combat infectious diseases, and it has shown the importance of preparedness for possible future diseases. A prominent property of the SARS-CoV-2 transmission is the significant fraction of asymptomatic transmission. This may influence the effectiveness of the standard contact tracing procedure for quarantining potentially infected individuals. However, the effects of asymptomatic transmission on the epidemic threshold of epidemic spreading on networks have rarely been studied explicitly. Here we study the critical percolation transition for an arbitrary disease with a nonzero asymptomatic rate in a simple epidemic network model in the presence of a recursive contact tracing algorithm for instant quarantining. We find that, above a certain fraction of asymptomatic transmission, standard contact tracing loses its ability to suppress spreading below the epidemic threshold. However, we also find that recursive contact tracing opens a possibility to contain epidemics with a large fraction of asymptomatic or presymptomatic transmission. In particular, we calculate the required fraction of network nodes participating in the contact tracing for networks with arbitrary degree distributions and for varying recursion depths and discuss the influence of recursion depth and asymptomatic rate on the epidemic percolation phase transition. We anticipate recursive contact tracing to provide a basis for digital, app-based contact tracing tools that extend the efficiency of contact tracing to diseases with a large fraction of asymptomatic transmission.

DOI: 10.1103/PhysRevE.104.054310

I. INTRODUCTION

The methods used to fight the spread of the contemporary COVID-19 epidemic in its initial phase have largely been the same as 100 years ago during the Spanish flu [1,2]. In particular, contact tracing has been used as a standard procedure that is well understood, both analytically and in network modeling approaches [3–8]. Some early papers even already considered the concept of recursive contact tracing, i.e., not only tracing direct contacts but also contacts of contacts and so on [9,10].

However, the arrival of the SARS-CoV-2 epidemic, with its high asymptomatic transmission rate and the possibility of presymptomatic infections, presents new challenges that need addressing [11–14]. As such, a renewed interest in recursive contact tracing [15–21], as well as in digital contact tracing solutions [22–34] that could enable instantaneous recursive contact tracing, has emerged in an effort to surpass the methods of 100 years ago.

In this article, we introduce a simple model that considers an epidemic as a percolation problem, as is common in network epidemiology theory [35–44], in combination with a recursive contact tracing algorithm operating on the model. Throughout this paper, we assume this algorithm to be facilitated by a digital contact tracing app which enables contact tracing and quarantining to happen effectively instantly; however, similar results could be achieved using recursive manual contact tracing, provided that the time necessary to trace contacts is small compared to the time between a person being infected and being infectious themselves, and that the recursion depth is sufficiently small. Note that the model operates in the theoretical limit of an ideal world without reporting, communication, or quarantining delays and without noncompliance with quarantining instructions, and we aim not to make quantitative predictions but to create a base model to further theoretical understanding. We thus do not explicitly model the SARS-CoV-2 virus, but an arbitrary virus with finite asymptomatic rate. We will study the efficacy of recursive contact tracing and characterize the influence of a possible future disease's asymptomatic transmission rate on the model's critical transition. Our model allows for arbitrary instantaneous recursion depths, as has been done only in [16], and our results, to the best of our knowledge, are the first to discuss the relationship of recursion depth and asymptomatic infection rate with regard to the critical transition.

We find a critical value in the fraction of nodes participating in the contact tracing (corresponding to tracing app usage) which depends on the asymptomatic transmission rate of the disease. Further we find a critical (maximum allowed) asymptomatic transmission rate as a function of the algorithm's recursion depth. We show that any disease with arbitrary basic reproduction number and finite asymptomatic rate can be stopped by a sufficiently large recursion depth. Finally, we validate our calculations using simulations on infection trees and networks with different degree distributions, as degree distribution can have a significant impact on an epidemic [38,40,42,44–48]. Let us now start by defining the model.

*lbaumgarten@itp.uni-bremen.de
†bornholdt@itp.uni-bremen.de

II. THEORY

We consider an SIR (susceptible, infected, removed) model with N nodes and an arbitrary degree distribution $p(k)$ in which a proportion Φ of nodes take part in contact tracing ("use a contact tracing app"). Nodes in the network are infected with a virus with symptomatic rate Θ and basic reproduction number R_0. It is known that in such a network, if we fix R_0, the disease has a transmissibility

$$T = R_0 \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle} \tag{1}$$

[40]. Carriers of the disease will be able to infect their susceptible neighbors with probability T one time step after being infected themselves and be immune and noncontagious afterwards.

If an infectious agent is symptomatic and uses the contact tracing app, this will trigger an alarm on the app and warn neighboring nodes of the chance of being infected, sending them into quarantine for their one infectious time step and removing them from quarantine afterwards so they effectively skip the infectious state and jump directly to the recovered stage. An infectious, symptomatic node will, however, have the chance to infect its neighboring nodes before triggering an alarm, which can be interpreted as a presymptomatic period or a testing delay.

We can consider higher degrees of recursivity r for the app, meaning how many time steps in the past the app will consider to guess who might currently be infected. For $r = 0$, only the node's direct neighbors are sent into quarantine. For $r = 1$ in addition to those nodes that are quarantined for $r = 0$, any node with a distance of exactly three to the symptomatic node is quarantined, for $r = 2$ any node with a distance of five is quarantined, and so on. This is illustrated in Fig. 1. The algorithm disregards any possible immunities due to nodes having already been infected previously, but it does consider breaks in the infection chain that are caused by the app's own quarantining algorithm, i.e., if a node was quarantined at time t, the app does not consider this node a possible infection spreader at that time step. We disregard possible immunities, although they are present in the underlying infection model because, for a new disease, the exact nature of immunity due to previous infection would likely not be immediately known, and it would thus be prudent to err on the side of caution and not assume immunity. Also note that it is nontrivial to determine which nodes' immunities would be known to the app (certainly the ones of nodes triggering alarms, but for other nodes it is unclear).

Given a vector $\vec{S}$ of symptomatically infected nodes at time t_0,

$$S_i = \begin{cases} 1 & \text{if node } i \text{ is symptomatically infected} \\ 0 & \text{otherwise} \end{cases},$$

the vector of nodes $\vec{U}$ using the app, the vectors $\vec{Q}(t)$ of quarantined nodes and $\vec{P}(t)$ of not quarantined nodes at time steps $t \leqslant t_0$, and the adjacency matrix A, the vector of quarantined nodes at $t = t_0 + 1$ can be calculated by

$$\vec{Q}(t_0+1) = \underbrace{\{A \cdot [\vec{S} \cdot \vec{U} \cdot \vec{P}(t_0)]\} \cdot \vec{P}(t_0) \cdot \vec{U}}_{r=0} + \underbrace{[A \cdot (A \cdot \{[A \cdot (\vec{S} \cdot \vec{U})] \cdot \vec{P}(t_0-1) \cdot \vec{P}(t_0-2) \cdot \vec{U}\}) \cdot \vec{P}(t_0-1) \cdot \vec{P}(t_0) \cdot \vec{U}] \cdot \vec{P}(t_0) \cdot \vec{U}}_{r=1} + \underbrace{\ldots}_{r>1}.$$

Multiplications with $\vec{P}(\cdot)$ ensure that a considered node in the backtracking chain was quarantined neither at its supposed time of infection nor at the time it could have infected its neighbors, and multiplications with $\vec{U}$ ensure that all nodes in the backtracking chain use the app. We now calculate the probability that an infected node is correctly put into quarantine by our algorithm. For this, we assume an infinitely large network, with a finite number of nodes being infected. In a network in which a finite fraction of nodes is infected, it is of course possible that a node will be in contact with multiple infected nodes in a single time step. With our assumption of the fraction of infected nodes being infinitely small, barring any nontrivial network structure, the chance of a node being in contact with more than one infected node also becomes infinitely small. We assume that the clustering in the network is negligible so that we can consider the infection chain effectively as a tree.

For $r = 0$, both the infected node and the infecting node must be part of the network and the infecting node needs to be symptomatic. Therefore, a first approximation of the probability $P_q^{r=0}$ of an infected node i being correctly put into quarantine is simply

$$P_q^{r=0}(\Phi, \Theta) = \Phi^2 \Theta. \tag{2}$$

However, node i has to have been infected by a different node j. For this infecting node j to have been infectious in the previous time step, it cannot have been quarantined in that time step. There are two possible reasons why node j would not have been quarantined despite being infected. Either it is not using the app, which happens with a probability $1 - \Phi$, or it is using the app, which has a probability of Φ, but the algorithm did not quarantine it in the time step in which it was infectious, which happens, for a node using the app, with a probability $1 - \frac{P_q^r}{\Phi}$. Therefore, the node j's probability of using the app, with the observation that it has not been quarantined despite being infectious, is

$$\Phi' = \frac{\Phi\left(1 - \frac{P_q^r}{\Phi}\right)}{\Phi\left(1 - \frac{P_q^r}{\Phi}\right) + (1 - \Phi)} = \frac{\Phi - P_q^r}{1 - P_q^r} \leqslant \Phi,$$

as the amount of nodes using the app with the ability to infect other nodes is reduced by a factor $(1 - P_q^{r=0})$, resulting in the

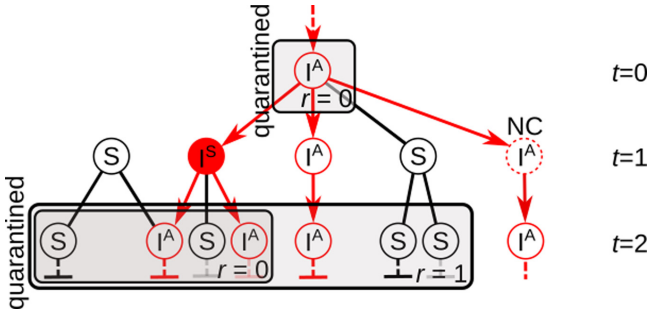


FIG. 1. Illustration of the quarantining algorithm with an infection spreading from top to bottom. Nodes with a black outline are not infected (susceptible = S), nodes with a red outline are infected (I), either symptomatically (filled nodes, I^S) or asymptomatically (unfilled nodes, I^A), and nodes with a dashed outline are not using the contact tracing app (noncompliant, NC). Red arrows indicate the spread of the infection, while black lines indicate noninfectious connections between nodes. The time t indicated on the right-hand side marks the time at which infected nodes are infectious—or, in the case of the uninfected nodes, the latest time at which the app would consider them to be possibly infectious. While nodes could reappear in later time steps, e.g., the node in the $t = 0$ row could also be shown in the $t = 2$ row as it is connected to (most of) the nodes in the $t = 1$ row, we show nodes only once for visual clarity. At time $t = 1$ a symptomatic node triggers an alarm on the app. For recursion depth $r = 0$, only its nearest neighbors are quarantined. These quarantined nodes cannot infect any other nodes, as indicated by the blocked outgoing connections. For $r = 1$, the app considers every nearest neighbor of the symptomatic node as a possible origin of the symptomatic node's infection and therefore quarantines all nodes that the infection could have spread to within two time steps from these nearest neighbors. This results in every node with a distance of exactly three to the symptomatic node being quarantined, so long as the connection is not interrupted by a node not using the app or by a node that was in quarantine itself at its time of infection or in the time step after infection, as shown on the right-hand side. This can, of course, also include nodes which have not yet actually been in contact with any infected nodes, as shown by the leftmost nodes in the $t = 1$ and $t = 2$ rows. Note that, although the infection chain is shown in a treelike structure for visual clarity, these nodes can be part of a network of arbitrary structure, so that two nodes might be connected via multiple different paths and therefore also have multiple possible distances to each other.

numerator, which is normalized by the total fraction of nodes that are not being quarantined, which is the denominator; and therefore

$$P_q^{r=0} = \Phi\Phi'\Theta. \tag{3}$$

For higher degrees of recursion, the chance of being quarantined is increased:

$$P_q^{r>0} = P_q^{r=0} + \underbrace{\left(1 - P_q^{r=0}\right)\Phi'' P_1}_{\text{r=1}} + \underbrace{\cdots}_{r>1} \tag{4}$$

$$= \Phi\Phi'[P_0 + (1 - P_0)\Phi''\{P_1 + (1 - P_1)\Phi''(\cdots)\}] \tag{5}$$

$$\text{with } P_0 = \Theta. \tag{6}$$

Here, in every part of the sum, the chance of a node having already been quarantined due to a lower recursion level is excluded via $(1 - P_i)$, and a factor Φ'' is added for the chance of the next upstream node using the app. The factor Φ'' represents the chance of a node using the app if the next downstream node has not been quarantined, and needs to be used for nodes that are two or more levels above the currently regarded node in the infection tree. The chance of such a node using the app regardless of the behavior of its downstream nodes is Φ'. The chance of a downstream node, which is using the app, of a node that is also using the app not being quarantined is approximately $(1 - \frac{P_q^r}{\Phi\Phi'})$. Since we assume both infecting node and infected node to be using the app, the factor $\Phi\Phi'$ is removed from P_q^r. This approximation disregards that the upstream node not being quarantined also influences the chance of its downstream node being quarantined. Then the chance of an upstream node using the app, given that its downstream node is using the app and has not been quarantined is

$$\Phi'' = \frac{\Phi'\left(1 - \frac{P_q^r}{\Phi\Phi'}\right)}{\Phi'\left(1 - \frac{P_q^r}{\Phi\Phi'}\right) + (1 - \Phi')} \tag{7}$$

$$= \frac{\Phi\Phi' - P_q^r}{\Phi - P_q^r}. \tag{8}$$

Next, we need to calculate the chance P_i of a node being quarantined due to the ith recursion step, given that its r nearest upstream nodes are using the app. For simplicity's sake, we start with P_1. Here a leaf node i is quarantined due to the first recursion step if any of the downstream nodes of i's second degree upstream node, which we call j, have been infected, use the app, and are symptomatic. The chance of one node fulfilling these conditions is $\Phi'\Theta T$. Since just one node needs to cause an alarm on the app, the chance of being quarantined is

$$P_1 = 1 - (1 - \Phi\Theta T)^n, \tag{9}$$

where n is the average number of j's downstream nodes minus one. We subtract one, since one of j's downstream nodes is i's direct upstream node and would already have caused i to be quarantined in the zeroth recursion step, if it were symptomatic. Since the chance of a node of degree k being infected is proportional to $kp(k)$ [46], the average number of downstream nodes minus one is

$$n = \frac{\sum_{k=2}^{\infty} k(k-2)p(k)}{\sum_{k=2}^{\infty} kp(k)}, \tag{10}$$

where we subtract two from k because of the one downstream node that is not considered and j's upstream node. Therefore,

$$P_1 = 1 - (1 - \Theta\Phi T)^{\frac{\sum_{k=2}^{\infty} k(k-2)p(k)}{\sum_{k=2}^{\infty} kp(k)}} \tag{11}$$

$$= P_1(x)|_{x=2} = 1 - (1 - \Theta\Phi T)^{\frac{\sum_{k=x}^{\infty} k(k-x)p(k)}{\sum_{k=x}^{\infty} kp(k)}}\Big|_{x=2}. \tag{12}$$

We indicate how many connections are removed when calculating n via the variable x.

For the second recursion step, at least one of the downstream nodes of j's upstream node, which we call l, must fulfill the condition of P_1, meaning that at least one of their downstream nodes must be infected, using the app, and

symptomatic. This chance is given by

$$P_2 = 1 - [1 - P_1(1)\tilde{\Phi}]^{\frac{\sum_{k=2}^{\infty} k(k-2)p(k)}{\sum_{k=2}^{\infty} kp(k)}} \tag{13}$$

$$= P_2(x)|_{x=2} = 1 - [1 - P_1(1)\tilde{\Phi}]^{\frac{\sum_{k=x}^{\infty} k(k-x)p(k)}{\sum_{k=x}^{\infty} kp(k)}}\Big|_{x=2} \tag{14}$$

$$\text{with } \tilde{\Phi} = \frac{\Phi T(1-\Theta)}{\Phi T(1-\Theta) + (1-\Phi T)}. \tag{15}$$

Here, in $P_1(x)$, we do not discount one of each node's downstream nodes, since these nodes are not upstream nodes of node i, and therefore all of their downstream nodes need to be considered. Thus, we use $P_1(1)$ instead of $P_1(2)$. Also, we use $\tilde{\Phi}$, because nodes that are using the app and symptomatically infected would have already caused a quarantine in a previous time step and can therefore not be part of the considered tree. Similarly, the equation for following recursion steps is

$$P_i(x) = 1 - [1 - P_{i-1}(1)\tilde{\Phi}]^{\frac{\sum_{k=x}^{\infty} k(k-x)p(k)}{\sum_{k=x}^{\infty} kp(k)}}. \tag{16}$$

Summarizing these calculations, the chance of a leaf node being quarantined with a recursion degree of r is

$$P_q^r \approx \Phi\Phi' \sum_{i=0}^{r} \left(\left\{ \prod_{j=0}^{i-1} [1 - P_j(2)]\Phi'' \right\} P_i(2) \right) \tag{17}$$

$$\text{with } P_i(x) = \begin{cases} \Theta & \text{if } i = 0 \\ 1 - (1 - P_0(1)\Phi T)^{n(x)} & \text{if } i = 1 \\ 1 - (1 - P_{i-1}(1)\tilde{\Phi})^{n(x)} & \text{otherwise} \end{cases} \tag{18}$$

$$\text{and } n(x) = \frac{\sum_{k=x}^{\infty} k(k-x)p(k)}{\sum_{k=x}^{\infty} kp(k)}. \tag{19}$$

Note that (17) is a self-consistent equation, since Φ' and Φ'' contain P_q^r.

III. THEORETICAL RESULTS

It is easy to see that the upper limit of P_q^r is

$$P_q^r \leqslant \Phi\Phi' < \Phi \text{ if } \Phi < 1, \tag{20}$$

so contact tracing by recursive backtracking is strictly worse than vaccinating a fraction Φ of the population. Since such a vaccination strategy is already insufficient to stop an epidemic on an infinitely large scale-free network with a degree distribution $p(k) \propto k^{-\gamma}$ with $\gamma \leqslant 3$ [46], recursive backtracking can also not stop such an epidemic for $\Phi < 1$.

However, there is still something that can be learned from taking a closer look at scale-free networks. For $\gamma \leqslant 3$, the sum $\sum_{k=2}^{k} k^2 p(k)$ in the exponent of the P_i's diverges, therefore $P_1 \to 1$ (if $\Phi\Theta T > 0$), and P_q^r becomes

$$P_q^r = \Phi\Phi'[\Theta + (1-\Theta)\Phi'']. \tag{21}$$

We can see that all infected nodes that can be caught by the algorithm will already be detected in the first recursion step.

Luckily, real-world networks are not infinitely large, so the sum mentioned previously will not diverge, so recursive backtracking will be able to stop epidemics for $\Phi < 1$. For such networks, we expect the observation made for infinitely large scale-free networks to be still be relevant, i.e., the closer a real-world network is to an infinitely large scale-free network, the less will the epidemic threshold Φ_c be affected by recursion depths past $r = 1$.

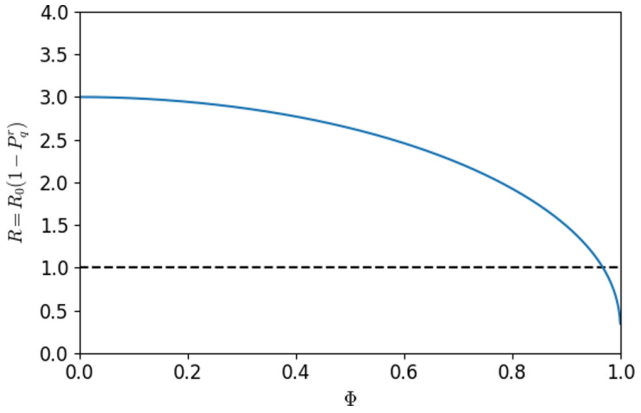


FIG. 2. Reduction of the reproduction number R as a function of the app-usage rate Φ for a Barabási-Albert (BA) network with a cutoff $\kappa = 1000$, recursion depth $r = 1$, $R_0 = 3$, and $\Theta = 0.5$. The dashed line shows the critical value $R_0(1 - P_q^r) = 1$.

In Fig. 2 we show the reduction of the reproduction number $R = R_0(1 - P_q^r)$ as a function of Φ for a Barabási-Albert (BA) network with average degree $\langle k \rangle = 4$ and a cutoff at $\kappa = 1000$ and recursion depth $r = 1$. We also tested this for a simple Erdős-Rényi (ER) network with average degree $\langle k \rangle = 4$, a scale-free network with exponential cutoff $p(k) \propto k^{-2} \exp(\frac{k}{94.2})$ that produces an epidemic threshold comparable to that of urban networks for SARS [49] and higher recursion depths. Since all of the resulting graphs are nearly indistinguishable [except that $R(\Phi = 1) \to 0$ for $r \to \infty$ while $R(\Phi = 1) \neq 0$ for $r = 1$], we chose to show only the BA network. We can also calculate the critical value Φ_c as a function of the symptomatic rate Θ, as is shown in Fig 3. There is a large visible difference between the classic contract tracing method with $r = 0$ and recursive contact tracing, even for relatively large values of Θ. While for $r > 0$ the

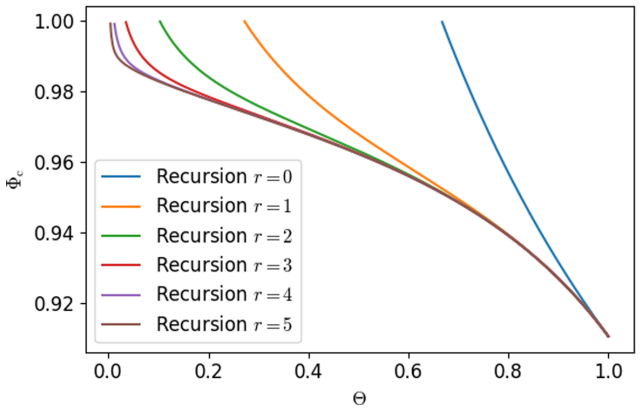


FIG. 3. Critical value Φ_c as a function of the symptomatic rate Θ for different recursion depths r with $R_0 = 3$. Since the ER distribution and the scale-free distribution with an exponential cutoff again yield almost the same results, we plot Φ_c only for the Barabási-Albert distribution with average degree $\langle k \rangle = 4$ and cutoff $\kappa = 1000$.

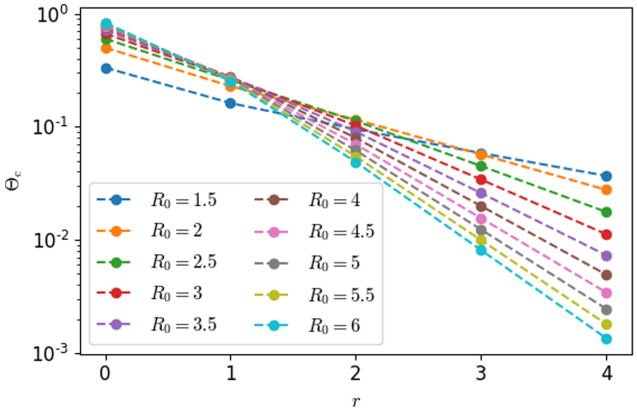


FIG. 4. Critical symptomatic rate Θ_c below which an epidemic cannot be stopped even for $\Phi = 1$ as a function of the recursion depth r for different basic reproduction numbers R_0 using a Barabási-Albert distribution with average degree $\langle k \rangle = 4$ and cutoff $\kappa = 1000$. For large recursion depths, the critical value $\Theta_c \to 0$ for all basic reproduction numbers, whereas for $r = 1$ there is a maximum $\Theta_c^{\max} \approx 0.28$ at $R_0 \approx 3.6$.

recursion depth has little influence on Φ_c for large values of the symptomatic rate Θ, we see that there is a critical value Θ_c, depending on the recursion depth, below which, even with $\Phi = 1$, an epidemic cannot be stopped. This critical value is approximately halved when going from the classical method $r = 0$ to $r = 1$, meaning that recursive contact tracing is an effective method to combat diseases with high asymptomatic rates which would not have been able to be stopped by previous contact tracing methods.

The critical value Θ_c is shown in Fig. 4 as a function of the recursion depth for different values of R_0. The critical value Θ_c exponentially decreases with r, with $\Theta_c \to 0$ for $r \to \infty$. Therefore, any disease with a symptomatic rate $\Theta > 0$ and arbitrarily large basic reproduction number R_0 can be stopped via recursive contact tracing, given a sufficiently large recursion depth and app usage rate.

IV. SIMULATIONS

To test the accuracy of our calculations in Sec. II, we simulate infection trees with recursive backtracking. The simulation starts with a single infected node, and each time step for each infected, unquarantined leaf node $k - 1$ downstream nodes are added, with k proportional to $kp(k)$. These new leaf nodes are infected with probability T and symptomatic with probability Θ. Then, according to the rules described in Sec. II, infected leaf nodes may be quarantined, causing them to not receive any downstream nodes. We let these dynamics run for 100 time steps or until there were 10 000 new infected leaf nodes added in a time step, at which point we consider the epidemic out of control. In Fig. 5 we show the fraction of trees in which the epidemic is not stopped within 100 time steps, the fraction of quarantined nodes, and the average reproduction number R for trees using an ER degree distribution

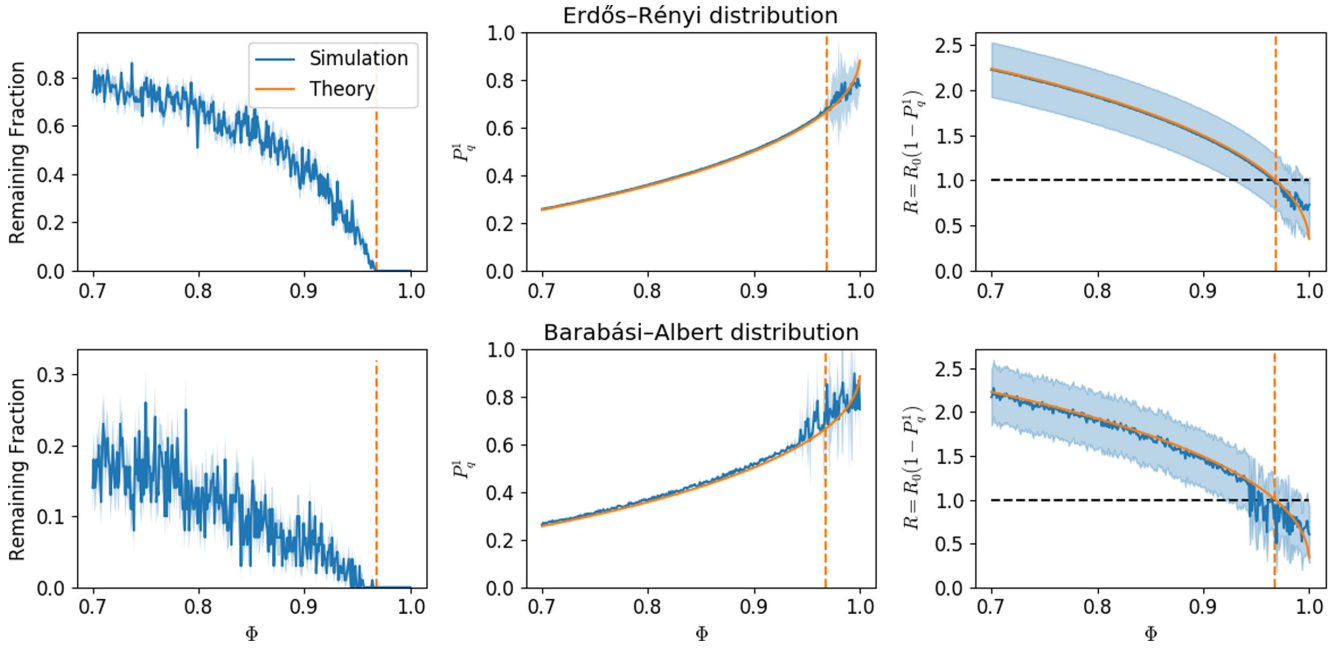


FIG. 5. Fraction of trees in which the epidemic survives 100 time steps (left column), probability of an infected node being quarantined P_q^1 (center column), and reproduction number R (right column) for trees built with an ER degree distribution (upper row) or a BA degree distribution with cutoff $\kappa = 1000$ (lower row), with $r = 1$, $R_0 = 3$, and $\Theta = 0.5$ where the shaded areas show the standard deviation. Blue lines show the averages of 100 trees per data point, unbroken orange lines show the theoretical results for P_q^1 and R, and dashed orange lines show the theoretical critical value Φ_c. The dashed black lines in the reproduction number diagrams show the critical value of R. Note that the measurement for the reproduction number R and the quarantined fraction P_q^1 are skewed near or past the critical point, because the measurements here are dominated by just the beginning of the tree where the quarantining algorithm does not have enough history yet to quarantine nodes.

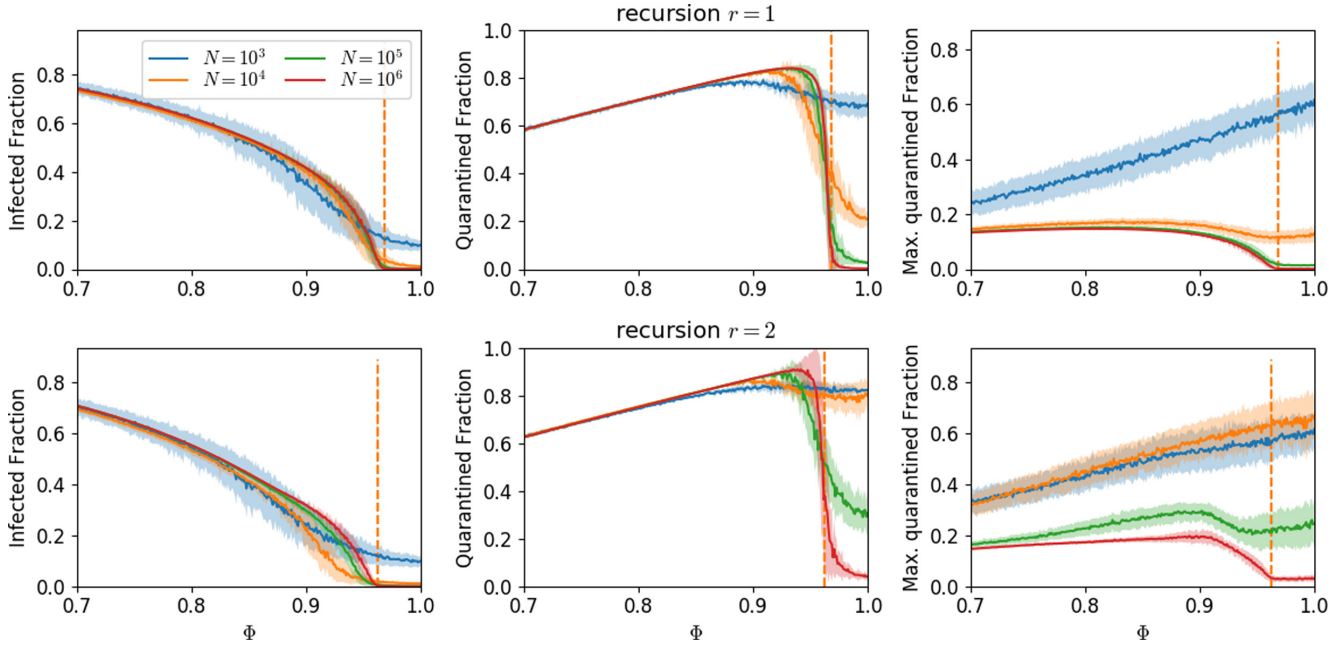


FIG. 6. Fraction of infected nodes (left), fraction of nodes that have ever been quarantined (center), and maximum number of nodes that have been quarantined at one time (right) for ER networks with recursion depth $r = 1$ (top row) and $r = 2$ (bottom row) as a function of the app-usage rate Φ. Different color graphs show networks of different sizes N, and orange dashed lines show the theoretical critical value Φ_c. All data points are the average of 100 simulation runs, and shaded areas show the standard deviation.

or a BA degree distribution with a cutoff $\kappa = 1000$. We see a very good agreement between our calculation and simulations for recursions $r = 1$; see Fig. 5. We have also verified that our calculations and simulations agree very well for larger recursion depths.

Next, we move away from the tree structure and use networks instead. In these networks, we start with ten initially infected nodes, which are chosen with a probability proportional to $kp(k)$, and we let the dynamics run until no new nodes are infected within a time step. Figure 6 shows the fraction of infected nodes, the fraction of nodes that have ever been quarantined, and the maximum fraction of nodes that has been quarantined at one point in time for ER networks with different recursion depths.

For the network size $N \to \infty$, we see that the fractions of infected and quarantined nodes drop to zero at the theoretical critical value Φ_c. For higher recursion depths and relatively small networks, the infected fraction is already kept quite low below the theoretical critical value because a large fraction of nodes is being quarantined and therefore the assumption we made in Sec. II that nodes are not coincidentally swept up in unrelated infection trees does not hold anymore; however, this lower infected fraction comes at the cost of wrongly quarantining a relatively large fraction of nodes. Also, this effect is mitigated for larger network sizes N.

For BA networks, especially for large networks, the infection dies out quickly even for low values of Φ, because the infection dynamics are dominated by the strongly connected hub nodes, which, after some time, will be in the recovered state, and therefore the effective degree distribution for the infection is quickly cut off for larger k. Additionally, in a BA network the first few nodes which are added to the network and later are likely to grow into the strongest connected nodes are likely to connect to each other and have common neighbors, meaning that the assumption we made in Sec. II of low clustering does not hold, which reduces the number of susceptible nodes adjacent to an infected large spreader i because its neighbors are likely to have already been infected by i's own infecting node. Both these effects lower the basic reproduction number R_0 below the theoretical value given by Eq. (1).

V. CONCLUSION

Considering the problem of epidemic spreading of an infectious disease with a finite asymptomatic transmission rate, such as the current epidemics caused by the SARS-CoV-2, we have introduced a combined infection model of nodes taking susceptible, infected, or recovered states with a recursive contact tracing algorithm for quarantining, equivalent to an app used by a network's nodes to stop a pandemic in our model. The contact tracing algorithm changes the percolation phase transition of the epidemic spreading model, and we here studied the interplay of these two processes in a minimal statistical mechanics model.

We have calculated the odds of an infected node being quarantined by the contact tracing algorithm, as well as the resulting theoretical critical values for the app usage rate above which an infection does not percolate through the network, and the minimum symptomatic rate beneath which a disease cannot be stopped, depending on the algorithm's recursion depth, the disease's basic reproduction number, and the contact network's underlying degree distribution.

We found that the critical app adoption rate and critical symptomatic rate are both significantly lower for an algorithm using recursive contact tracing, even with a low recursion

depth, than for the classically employed, nonrecursive method of direct contact tracing. In fact, any disease with a finite symptomatic rate and arbitrary basic reproduction number can be stopped if the app usage rate and recursion depth are large enough, meaning that recursive contact tracing can be an effective method for controlling diseases with large asymptomatic transmission rates which could not have been stopped with previous contact tracing methods.

Our critical app adoption rate of over 95% may seem unusually high at first glance compared to some other results [5,23,27,29], with other estimates generally lying between 56% and 95% [50]. However, this is simply caused by our model's harsh assumptions, such as a very high basic reproduction number $R_0 = 3$, a relatively high asymptomatic rate of 50%. Furthermore, keep in mind that we here study an idealized statistical mechanics model without further infection prevention measures, such as random testing or social distancing, apart from contact tracing, not distinguishing between the infectivity of symptomatic and asymptomatic disease carriers (symptomatic carriers are often assumed to self-quarantine and therefore infect fewer people), and a lack of manual contact tracing even for symptomatic infected individuals who are not using the app. Our results are comparable to those of other models making harsh assumptions [16,17,26]. We stress once again, however, that our model's goal is not to make quantitative predictions, but to provide a theoretical basis for understanding the limits of recursive contact tracing and further work.

Further, we found that, while higher recursion depths can stop diseases with a high asymptomatic rate, for low asymptomatic rates, recursion depths higher than one show very little improvement in the critical app usage rate while falsely quarantining more uninfected nodes, implying that for such diseases recursion depths larger than one are mostly not useful.

Also, the contact network's degree distribution was shown to have little impact on these critical values, so recursive contact tracing is not only viable for Erdős-Rényi graphs, as tested in previous studies, but also for more realistic scale-free-like networks, i.e., scale-free networks with a cutoff.

We have ensured the accuracy of our theoretical calculations using simulations on infection trees and networks with different degree distributions. We found very good agreement between our calculations and simulations for any degree distribution on infection trees and for Erdős-Rényi networks. For Barabási-Albert networks, the simulation's critical values lie below the calculated ones because quarantining the most connected nodes quickly changes the network's degree distribution and because the effect of clustering, as highly connected nodes in Barabási-Albert networks are likely to be connected to each other, was not considered in the calculations.

The calculations presented here are viable for a simple model, but we believe that the qualitative conclusions should be applicable to the real world as well. Future research should expand this simple model to be more realistic and possibly fit the infection profiles of real diseases, as well as consider the effect of clustering on the model's critical values.

The presented model could easily be extended to more closely model real-life processes, for example, by introducing parameters for presymptomatic durations, delays in testing or communication, using different reproduction numbers for symptomatic and asymptomatic individuals, or studying real-life networks that model household structures. Further, in the real world, an asymptomatic node who is considered by the algorithm to have potentially been infected could be tested and then be used as a new index case for further contact tracing.

Also, as the exact nature of immunity due to previous infection would not be immediately known for any new disease, we erred on the side of caution and assumed the possibility of reinfection when considering who should be quarantined, although the underlying infection model does not allow this. Should the existence of such immunities be known, one could instead remove previously infected individuals from consideration when determining possible infection chains and thereby lower the false positive rate of quarantining. Conversely, the model could also be modified to take into consideration the possibility of recovered or vaccinated agents still possibly becoming infectious disease carriers, despite being immune themselves.

Finally, while digital contact tracing has been the underlying case for our model, it could also be extended to simulate and explore the theoretical limits of recursive manual contact tracing, with the inherent difficulties and unavoidable delays therein.

[1] A. F. Franchini, F. Auxilia, P. M. Galimberti, M. A. Piga, S. Castaldi, and A. Porro, Covid 19 and Spanish flu pandemics: All it changes, nothing changes, Acta Biomed. **91**, 245 (2020).

[2] D. C. Wheelock, What can we learn from the Spanish Flu Pandemic of 1918-19 for COVID-19? Federal Reserve Bank of St. Louis Economic Synopses **30**, 1 (2020).

[3] M. Kretzschmar, Y. T. van Duynhoven, and A. J. Severijnen, Modeling prevention strategies for gonorrhea and chlamydia using stochastic network simulations, Am. J. Epidem. **144**, 306 (1996).

[4] M. Eichner, Case isolation and contact tracing can prevent the spread of smallpox, Am. J. Epidem. **158**, 118 (2003).

[5] K. T. Eames and M. J. Keeling, Contact tracing and disease control, Proc. R. Soc. London B **270**, 2565 (2003).

[6] C. Fraser, S. Riley, R. M. Anderson, and N. M. Ferguson, Factors that make an infectious disease outbreak controllable, Proc. Natl. Acad. Sci. USA **101**, 6146 (2004).

[7] I. Z. Kiss, D. M. Green, and R. R. Kao, Disease contact tracing in random and clustered networks, Proc. R. Soc. B **272**, 1407 (2005).

[8] J. Müller and M. Kretzschmar, Contact tracing—Old models and new challenges, Infect. Disease Model. **6**, 222 (2021).

[9] J. Müller, M. Kretzschmar, and K. Dietz, Contact tracing in stochastic and deterministic epidemic models, Math. Biosci. **164**, 39 (2000).

[10] D. Klinkenberg, C. Fraser, and H. Heesterbeek, The effectiveness of contact tracing in emerging epidemics, PloS ONE **1**, e12 (2006).

[11] X. Yu and R. Yang, Covid-19 transmission through asymptomatic carriers is a challenge to containment, Influenza and Other Respiratory Viruses **14**, 474 (2020).

[12] L. Pribylová and V. Hajnova, SEIAR model with asymptomatic cohort and consequences to efficiency of quarantine government measures in COVID-19 epidemic, arXiv:2004.02601 (2020).

[13] S. Khailaie, T. Mitra, A. Bandyopadhyay, M. Schips, P. Mascheroni, P. Vanella, B. Lange, S. C. Binder, and M. Meyer-Hermann, Development of the reproduction number from coronavirus SARS-CoV-2 case data in germany and implications for political measures, BMC medicine **19**, 1 (2021).

[14] S. M. Moghadas, M. C. Fitzpatrick, P. Sah, A. Pandey, A. Shoukat, B. H. Singer, and A. P. Galvani, The implications of silent transmission for the control of Covid-19 outbreaks, Proc. Natl. Acad. Sci. USA **117**, 17513 (2020).

[15] A. Okolie and J. Müller, Exact and approximate formulas for contact tracing on random trees, Math. Biosci. **321**, 108320 (2020).

[16] V. B. Bulchandani, S. Shivam, S. Moudgalya, and S. Sondhi, Digital herd immunity and Covid-19, Phys. Biol. **18**, 045004 (2021).

[17] A. Lambert, A mathematical assessment of the efficiency of quarantining and contact tracing in curbing the COVID-19 epidemic, Mathematical Modelling of Natural Phenomena **16**, 53 (2021).

[18] M. Barlow, A branching process with contact tracing, arXiv:2007.16182.

[19] A. Endo, Q. J. Leclerc, G. M. Knight, G. F. Medley, K. E. Atkins, S. Funk, A. J. Kucharski, Implication of backward contact tracing in the presence of overdispersed transmission in COVID-19 outbreak, [version 3; peer review 2 approved], Wellcome Open Research, 10.12688/wellcomeopenres.16344.3 (2021).

[20] S. Kojaku, L. Hébert-Dufresne, E. Mones, S. Lehmann, and Y.-Y. Ahn, The effectiveness of backward contact tracing in networks, Nat. Phys. **17**, 652 (2021).

[21] S. Shivam, V. B. Bulchandani, and S. Sondhi, Recursive contact tracing in Reed-Frost epidemic models, Phys. Biol. **18**, 065001 (2021).

[22] M. Faggian, M. Urbani, and L. Zanotto, Proximity: A recipe to break the outbreak, arXiv:2003.10222.

[23] J. Hellewell, S. Abbott, A. Gimma, N. I. Bosse, C. I. Jarvis, T. W. Russell, J. D. Munday, A. J. Kucharski, W. J. Edmunds, F. Sun *et al.*, Feasibility of controlling COVID-19 outbreaks by isolation of cases and contacts, The Lancet Global Health **8**, e488 (2020).

[24] R. Hinch, W. Probert, A. Nurtay, M. Kendall, C. Wymant, M. Hall, and C. Fraser, Effective configurations of a digital contact tracing app: A report to NHSX, https://github.com/BDI-pathogens/covid-19_instant_tracing/blob/master/Report.

[25] H. Kim and A. Paul, Automated contact tracing: A game of big numbers in the time of COVID-19, Journal of the Royal Society Interface **18**, 20200954 (2021).

[26] Y. Xia and G. Lee, How to return to normalcy: Fast and comprehensive contact tracing of Covid-19 through proximity sensing using mobile devices, arXiv:2004.12576.

[27] L. Ferretti, C. Wymant, M. Kendall, L. Zhao, A. Nurtay, L. Abeler-Dörner, M. Parker, D. Bonsall, and C. Fraser, Quantifying SARS-Cov-2 transmission suggests epidemic control with digital contact tracing, Science **368** eabb6936 (2020).

[28] S. McLachlan, P. Lucas, K. Dube, G. S. McLachlan, G. A. Hitman, M. Osman, and N. Fenton, The fundamental limitations of COVID-19 contact tracing methods and how to resolve them with a Bayesian network approach, London, UK, http://dx.doi.org/10.13140/RG.2.2.27042.66243 (2020).

[29] E. Hernández-Orallo, P. Manzoni, C. T. Calafate, and J.-C. Cano, Evaluating how smartphone contact tracing technology can reduce the spread of infectious diseases: The case of COVID-19, IEEE Access **8**, 99083 (2020).

[30] B. Prasse and P. V. Mieghem, Mobile smartphone tracing can detect almost all SARS-CoV-2 infections, arXiv:2006.14285 (2020).

[31] Y.-C. Ho, Y.-H. Chen, S.-H. Hung, C.-H. Huang, P. Po, C.-H. Chan, D.-K. Yang, Y.-C. Tu, T.-L. Liu, and C.-T. Fang, Social distancing 2.0 with privacy-preserving contact tracing to avoid a second wave of Covid-19, arXiv:2006.16611.

[32] G. Cencetti, G. Santin, A. Longa, E. Pigani, A. Barrat, C. Cattuto, S. Lehmann, M. Salathé, and B. Lepri, Digital proximity tracing on empirical contact networks for pandemic control, Nat. Commun. **12**, 1 (2021).

[33] G. Bianconi, H. Sun, G. Rapisardi, and A. Arenas, A message-passing approach to epidemic tracing and mitigation with apps, Phys. Rev. Research **3**, L012014 (2021).

[34] A. Barrat, C. Cattuto, M. Kivelä, S. Lehmann, and J. Saramäki, Effect of manual and digital contact tracing on Covid-19 outbreaks: A study on empirical contact data, J. R. Soc., Interface **18**, 20201000 (2020).

[35] P. Grassberger, On the critical behavior of the general epidemic process and dynamical percolation, Math. Biosci. **63**, 157 (1983).

[36] J. L. Cardy and P. Grassberger, Epidemic models and percolation, J. Phys. A: Math. Gen. **18**, L267 (1985).

[37] M. E. J. Newman and D. J. Watts, Scaling and percolation in the small-world network model, in *The Structure and Dynamics of Networks*, edited by M. E. J. Newman, A.-L. Barabasi, and D. J. Watts (Princeton University Press, 2011) pp. 310–320.

[38] C. Moore and M. E. J. Newman, Epidemics and percolation in small-world networks, Phys. Rev. E **61**, 5678 (2000).

[39] R. Pastor-Satorras and A. Vespignani, Epidemic dynamics and endemic states in complex networks, Phys. Rev. E **63**, 066117 (2001).

[40] M. E. J. Newman, Spread of epidemic disease on networks, Phys. Rev. E **66**, 016128 (2002).

[41] C. P. Warren, L. M. Sander, and I. M. Sokolov, Geography in a scale-free network model, Phys. Rev. E **66**, 056105 (2002).

[42] M. J. Keeling and K. T. Eames, Networks and epidemic models, J. R. Soc., Interface **2**, 295 (2005).

[43] L. Meyers, Contact network epidemiology: Bond percolation applied to infectious disease prediction and control, Bull. Am. Math. Soc. **44**, 63 (2007).

[44] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, Epidemic processes in complex networks, Rev. Mod. Phys. **87**, 925 (2015).

[45] R. Pastor-Satorras and A. Vespignani, Epidemic Spreading in Scale-Free Networks, Phys. Rev. Lett. **86**, 3200 (2001).

[46] N. Madar, T. Kalisky, R. Cohen, D. Ben-avraham, and S. Havlin, Immunization and epidemic dynamics in complex networks, Eur. Phys. J. B **38**, 269 (2004).

[47] J. O. Lloyd-Smith, S. J. Schreiber, P. E. Kopp, and W. M. Getz, Superspreading and the effect of individual variation on disease emergence, Nature (London) **438**, 355 (2005).

[48] C. Castellano and R. Pastor-Satorras, Thresholds for Epidemic Spreading in Networks, Phys. Rev. Lett. **105**, 218701 (2010).

[49] L. A. Meyers, B. Pourbohloul, M. E. Newman, D. M. Skowronski, and R. C. Brunham, Network theory and SARS: Predicting outbreak diversity, J. Theor. Biol. **232**, 71 (2005).

[50] I. Braithwaite, T. Callender, M. Bullock, and R. W. Aldridge, Automated and partially-automated contact tracing: A rapid systematic review to inform the control of COVID-19, The Lancet Digital Health **2**, e607 (2020).

Robustness to noisy signal transmission delays in genetic networks

Lorenz Baumgarten and Stefan Bornholdt. "Robustness to noisy signal transmission delays in genetic networks". To be submitted.

Robustness to noisy signal transmission delays in genetic networks

Lorenz Baumgarten* and Stefan Bornholdt†

Institut für Theoretische Physik, Universität Bremen, 28759 Bremen, Germany
(Dated: February 23, 2022)

It has been shown that many real life gene regulatory networks can be described by simple, deterministic Boolean networks models. However, in the real world, these networks need to operate in a highly noisy environment. It stands to reason that a genetic network needs to be specifically designed to be able to reliably fulfill its function despite this noise. Here, we test this hypothesis by extending a number of Boolean network models found on the cellcollective.org database to include stochastic transmission time delays of signals and comparing these networks' cyclic attractors' stability under noise to comparable random networks. We find that gene regulatory networks show a remarkable stability when compared to their randomized variants.

I. INTRODUCTION

Boolean network models have been a popular tool for modeling genetic networks since their inception [1–11], as they provide an attractive option of simulating complex genetic networks without needing to concern oneself with biological details. Real world networks, however, invariably will have to contend with noise while performing their functions [8, 12–14].

It is therefore no wonder that there exists a host of research concerning itself with the robustness of genetic networks against noise [15–26] and the relationship between the simplified Boolean models and their real world equivalents [27–30].

Due to the observed robustness in these papers, it is a reasonable assumption that most if not all genetic networks have evolved towards robustness against noise.

Here, we want to test this hypothesis by studying the robustness of cyclic attractors of networks found in the online genetic network database cellcollective.org [31, 32] in the form of Boolean network models. We extract all networks of the database whose papers sufficiently describe the network's cyclic attractors. We then test the robustness of these networks towards noise in the signal transmission time as done in [19]. As a comparison to these networks, we first use comparable, i.e., same number of genes, similar number of connections, same attractor length, etc., random networks and observe a substantially larger robustness for most of the genetic networks.

To determine whether this robustness is caused by the network's topology or the shape of its attractor itself, we also compare the genetic networks' robustness to randomized networks with cyclic attractors that more closely resemble or are equivalent to the original networks' attractors. Here, we also find a higher, albeit less so than for the earlier randomizations, robustness of the genetic networks. These findings suggest that the state sequence of genetic networks' cyclic attractors as well as their underlying topology are evolved to be robust against noise.

* lbaumgarten@itp.uni-bremen.de
† bornholdt@itp.uni-bremen.de

II. METHODS

A. Model

The models given in the cellcollective.org database are networks with Boolean update rules

$$\sigma_i(t) = f_i(\{\sigma_j(t - t_d)\})$$

representing real life genetic networks. Here, $\sigma_i(t) \in \{0, 1\}$ is the Boolean state of gene i at time t, the function $f_i \in \{0, 1\}$ determines a gene's response to a given input, i.e., its next state depending on the current states of all genes, and $t_d = 1$ is the signal transmission time. To test a network's stability under noise, as done in [19] with techniques introduced in [33], we modify this model by introducing an internal variable of protein concentration levels $c_i(t) \in [0, 1]$ following

$$c_i(t > t_0) = \begin{cases} 1 - [1 - c_i(t_0)] \exp\left[\frac{-(t-t_0)}{\tau}\right] & \text{if } f_i = 0 \\ c_i(t_0) \exp\left[\frac{-(t-t_0)}{\tau}\right] & \text{if } f_i = 1, \end{cases}$$

where $\tau = 0.3$ is a time constant, as well as fluctuating signal transmission delays, meaning that signal transmission between genes is delayed by a random variable $\chi \in [0, \chi_{\max}]$ in addition to the standard signal transmission time,

$$t_d = 1 + \chi.$$

A gene's state is then determined by a threshold function of the protein concentration levels

$$\sigma_i(t) = \begin{cases} 1 & \text{if } c_i(t) \geq 0.5 \\ 0 & \text{if } c_i(t) < 0.5. \end{cases}$$

These two additions allow us to more realistically simulate a genetic network as signal transmission times will

undoubtedly vary in a noisy environment, and the internal protein concentration variable allows the network to not respond to short signal spikes caused by the transmission delay variability. This will allow the network to remain in the same attractor as without noise; however, if the noise level, represented by $\chi_{\max}$, is large enough, the attractor can still be escaped despite the "buffer" afforded by the interal protein concentration levels.
We define a network's attractor, given by n attractor states $\{S_i\}$, as still functioning under noise as long as the network assumes the attractor states S_i in the given order, where we ignore any states the networks assumes for less than half a timestep. We also consider an attractor as no longer functioning if no state has been assumed for more than half a time step within a time of $2(1+\chi_{\max})$ as can sometimes happen for very large levels of noise $\chi_{\max}$. With this, we test an attractor's stability under noise by measuring the average time the network remains in the attractor under a given noise level.

As a comparison benchmark, we use randomized versions of the studied networks. For a given attractor, random networks are created using the same number of nodes as the original network. Each of the nodes in such a network then receives incoming connections from n random nodes where n is a random number between 1 and N generated from a normal distribution with the mean and standard deviation of incoming connections in the original network. Next, another number p for the fraction of incoming signals that elicit an "on"-response is generated from a normal distribution with the mean and standard deviation of the fractions of "on"-responses of each node in the original network. Responses to random signals are then set to "on" until the fraction p is reached while the rest is set to "off". Regardless of the random numbers each node has to have at least one incoming connection and at least one "on"- and one "off"-response so all nodes can participate in the network's dynamics. Finally, we initialize the network in a random state and let its dynamics run without noise until an attractor is found. If the found attractor does not have the same length as the original network's attractor, the randomized network is discarded and the randomization process is repeated until an attractor of equal length is found. We will call this form of randomization the complete randomization.
As we will see that Hamming distances between consecutive states in attractors of completely randomized networks are often significantly larger than those found in genetic networks, we will also use two additiona types of random networks whose attractors are more comparable to those of the original genetic networks: the functionally equivalent randomization and the randomization with similar Hamming distances.
For the functionally equivalent randomization, we produce a random network that has the same number of genes as the original network and exhibits the exact same attractor as an attractor of the original network that we want to compare it to, albeit with different connections between genes.
To create a functionally equivalent network, we first find a random set of connections that can facilitate the attractor by iteratively removing random superfluous connections from a fully connected network. For each node, a target number of incoming connections is randomly selected from a normal distribution with the same average and standard deviation as the incoming connections in the inital network. Incoming connections are removed for each node until either this target number is reached or there are no more removable connections. This set of connections will contain a minimum set of necessary connections, meaning that no connection can be removed from this set while still reproducing the attractor. Note that this does not mean that this set has to be the set with the absolute minimum of necessary connections, i.e., there can be multiple different such sets.
Finally, the genes' responses are set similarly to the randomization described above; however, the responses necessary for the attractor that shall be recreated are set accordingly before other random responses that are irrelevant for the attractor are set to "on" until the node's target number for the fraction of "on"-responses is reached.
For the randomization with similar Hamming distances, for every cyclic attractor found for a network, we create a random network with the same number of nodes and attractor length as the original network. Assuming, we want a comparison network for an m-state attractor of an N gene network, we randomly select one initial attractor state for N genes with the same probability of a node being "on" as in the original attractor. Then, the following $m-1$ attractor states of the randomized network are created iteratively by changing the states of n nodes in the previous attractor state where n is a random number between 1 and N picked from a normal distribution with the mean and standard deviation of Hamming distances between consecutive attractor states in the original attractor. Finally, we generate one last n from the same distribution as before and accept the generated attractor states if the Hamming distance between the final and the first attractor states is not larger than n. Otherwise, we reject the attractor states and generate new attractor states until an acceptable set of states is found. Next, we set the connections and node responses as described above for the functionally equivalent randomization, only with the randomly selected attractor as a goal.

B. Data

We utilize the cellcollective.org database providing 79 Boolean network models at the time of writing. Because we study the stability of cyclic attractors, we filter out all models whose papers do not mention any cyclic behaviour or whose attractors are trivial two-state attractors. Further, as we only want to study attractors of known biological relevance, we also do not use networks with more than two attractors of unknown function or whose attractors are mentioned but whose nature—

attractor length, at minimum—are not documented. The remaining networks are listed in Table I, and their architectures are shown in the supplemental material. We also add one additional popular network that we have studied previously: the fission yeast cell cycle network [38] where we added a restart signal to make the attractor cyclic.

III. RESULTS

For all studied networks, we initialize the networks in random states and run their dynamics without noise and in the limit of $\tau \to 0$, which recreates standard Boolean network update rules as are used in the cellcollective.org database, until an attractor is found. If the found attractor is a steady state, we repeat the process until a cyclic attractor is found. For every found attractor, we measure the time t_{att} the network as well as its completely randomized variant remain in the attractor for different levels of noise $\chi_{\max}$ and $\tau = 0.3$. The simulation is run up to a maximum time of $t_{\max} = 10^4$. Figure 1 shows the average of all attractors found after 10^4 simulation runs. We find that in five cases (Arabidopsis, Budding Yeast, CRP, the biologically relevant attractor of length 10 of Fission Yeast, Mammalian CC), the real world genetic networks are significantly more stable in the presence of noise than their randomized counterparts. For the remaining networks, we find that the OSP and IA+OSR networks are roughly as stable as an average random network and that the attractor of length 6 of Fission Yeast as well as the CC Transcription attractors are already unstable if just the time constant τ is increased, even in the absence of noise.

For the Fission Yeast, this is an agreeable result as this attractor is not biologically relevant. For CC Transcription, however, the reported attractor seems to be unable to function under any level of noise and therefore is likely not the attractor one would find in the real world.

If we take a closer look at the nature of the randomized attractors, we find that the Hamming distances between consecutive attractor states is often significantly larger in random networks than in the genetic networks, as is shown in Figure 2 Therefore, to determine whether this stability is caused by the similarity—and the concomitant lower opportunities for possible errors—of consecutive attractor states, by the exact sequence of attractor states, or by the topology underlying these attractors, we test the two additional randomization methods described in section II A: the functionally equivalent randomization (FE rand.) and the similhar Hamming distance randomization (sHd rand.). In Figure 3, we see that networks created using these two randomizations are indeed more stable in many cases than the simpler randomized networks shown in Figure 1. For Arabidopsis, Budding Yeast, IA+OSR, the 10-length attractor of Fission Yeast, and Mammalian CC these randomizations are comparable to or slightly more stable than the previous randomization while still being less stable than the original networks. Notably, for CRP, the functionally equivalent randomization is roughly as stable as the original network while being significantly more stable than the sHd variant, indicating that the attractor states themselves are enough to guarantee stability here. For OSP, the new randomized networks are even more stable than the original network.

It seems clear, however, that the superior stability of Arabidopsis, Budding Yeast, and Mammalian CC attractors compared to their randomized variants indicates that not only the form of the attractor, but the topology of individual gene interactions itself is optimized for stability under noise.

IV. DISCUSSION

Using Boolean genetic network models from the cellcollective.org database, we tested the hypothesis that genetic networks are evolutionarily designed for robustness to noise by studying the stability of networks with cyclic attractors against noise in signal transmission delays. We found that most genetic networks are more stable under this noise than comparable random networks with the same attractor lengths. One factor contributing to this stability is the average Hamming distance between attractor states which is in most cases lower for real networks than randomized networks, giving the real networks less potential for errors.

Only one network (CC Transcription) was a clear outlier here. This network's attractor is unstable even for very small levels of noise, indicating that either this attractor is not the one found in the real genetic network, that the network has been simplified too much to still retain its stability, or that there are more phenomena present within the real system guaranteeing the attractor sequence.

To uncover further stability decoupled from this, we compared the networks to networks created with two additional randomization methods: one method that creates the exact same attractor, albeit with different network topology, and another method that creates a random attractor with the same length as the original network and similar Hamming distances between attractor states. We observed that these networks are in most cases more stable than the comletely randomized ones, while still being mostly less stable than the original networks, with the random networks with the exact same attractor as the original networks still being marginally more stable than randomized networks with similar Hamming distances.

We conclude from this that real life genetic networks are designed to reliably work in a noisy environment, with this stability being caused by a noise resistant attractor being supported by an underlying noise resistant topology.

Further research could of course continue testing this hypothesis for a larger collection of networks or for the reliability of transients leading to an attractor instead of

TABLE I. Short handles used in this paper, full names on the cellcollective.org database and original papers for all networks used in this paper.

Short handle	Full name on cellcollective.org	Original paper
Arabidopsis	Arabidopsis thaliana Cell Cycle	[34]
Budding Yeast	Budding Yeast Cell Cycle 2009	[35]
CC Transcription	Cell Cycle Transcription By Coupled CDK And Network Oscillators	[36]
CRP	Cholesterol Regulatory Pathway	[37][a]
Fission Yeast	—	[38][b]
IA+OSR	Iron Acquisition And Oxidative Stress Response In Aspergillus Fumigatus	[39]
Mammalian CC	Mammalian Cell Cycle	[40]
OSP	Oxidative Stress Pathway	[41][c]

[a] Here, the Boolean functions given in the cellcollective.org database differed from those in the original paper for SREBP_SCAP, Farnesyl_pyrophosphate, and Geranyl_pyrophosphate . We used the Boolean functions from the original paper to reproduce the paper's attractors.
[b] To enable periodic behavior, we added a Start gene, see Supplemental Material
[c] Here, we used the length 7 attractor described in the paper and not the one found on the cellcollective.org database.

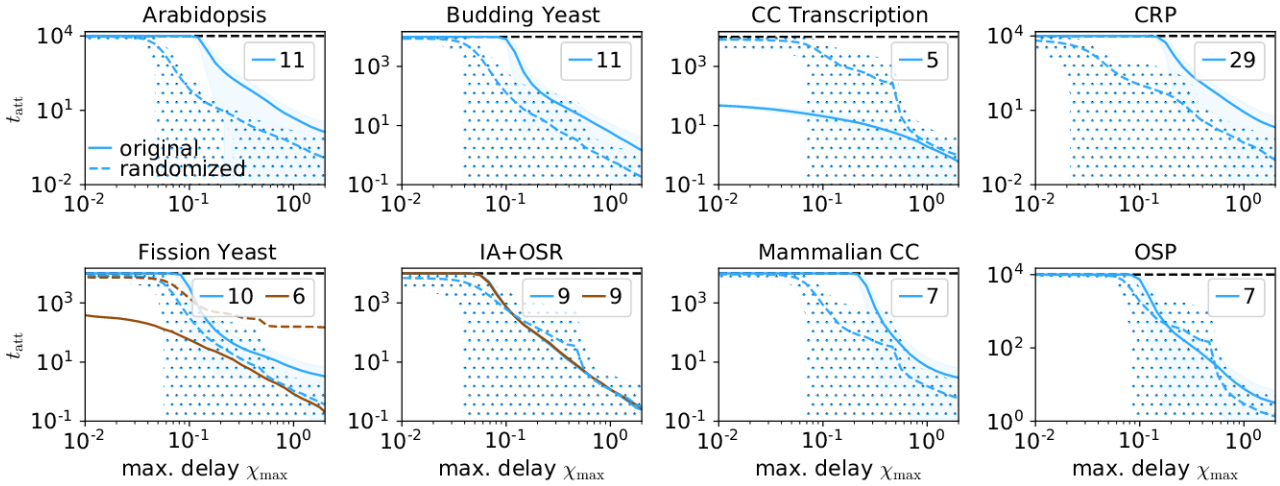


FIG. 1. Average time t_{att} networks remain in their attractors under noise as a function of maximum signal transmission delay χ_{max}, with a cutoff at $t_{max} = 10^4$ (black dashed line). 10^4 different attractors per network were used. Different colors denote different attractors, and the legends show the attractor lengths in order of frequence of appearance. Note that, while IA+OSR has two very similar attractors of length nine, since only the attractor length is relevant for randomization, only one line is shown for the randomized variants. For each genetic network, the shaded area shows the standard deviation of the most common attractor, and the dotted area shows the standard deviation of the corresponding randomized networks. Very rare attractors which were found with less than 1 % of initializations which led to cyclic attractors are not shown. Solid lines show the original genetic networks' stabilities and dashed lines show their completely randomized variants.

cyclic attractors themselves. We also believe that the method for studying stability of a network discussed here could be used to determine whether a model representation of a real life network is accurate as a genetic network unstable against noise is most likely not realistic.

[1] S. A. Kauffman, Journal of theoretical biology **22**, 437 (1969).
[2] R. Thomas, Journal of theoretical biology **42**, 563 (1973).
[3] S. A. Kauffman *et al.*, *The origins of order: Self-organization and selection in evolution* (Oxford University Press, USA, 1993).
[4] R. Albert and H. G. Othmer, Journal of theoretical biology **223**, 1 (2003).
[5] I. Albert, J. Thakar, S. Li, R. Zhang, and R. Albert, Source code for biology and medicine **3**, 1 (2008).
[6] S. Bornholdt, Journal of the Royal Society Interface **5**, S85 (2008).

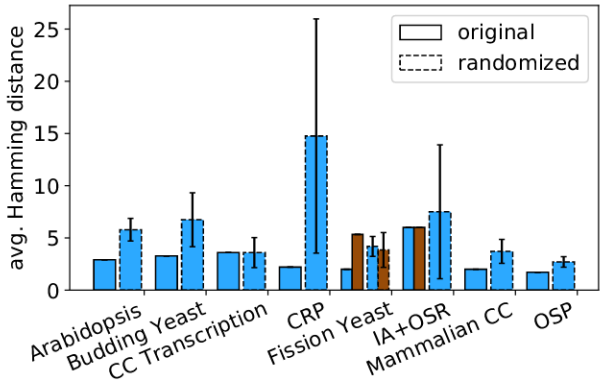


FIG. 2. Average Hamming distance between consecutive attractor states for different networks. Bar colors represent the different attractors shown in Figure 1. Bars with solid edges represent the genetic networks and bars with dashed edges represent their randomized counterparts. Black error bars are shown for the randomized networks and denote the standard deviation of the averages of Hamming distances within each network.

[7] M. Davidich and S. Bornholdt, Journal of Theoretical Biology **255**, 269 (2008).
[8] G. Karlebach and R. Shamir, Nature reviews Molecular cell biology **9**, 770 (2008).
[9] I. Shmulevich and J. D. Aitchison, Methods in enzymology **467**, 335 (2009).
[10] M. Rybarsch and S. Bornholdt, Physical Review E **86**, 026114 (2012).
[11] R.-S. Wang, A. Saadatpour, and R. Albert, Physical biology **9**, 055001 (2012).
[12] P. S. Swain, M. B. Elowitz, and E. D. Siggia, Proceedings of the National Academy of Sciences **99**, 12795 (2002).
[13] K. Klemm and S. Bornholdt, Physical Review E **72**, 055101 (2005).
[14] D. Ball, N. Adames, N. Reischmann, D. Barik, C. Franck, J. J. Tyson, and J. Peccoud, Cell Cycle **12**, 3392 (2013).
[15] L. Chen and K. Aihara, IEEE Transactions on circuits and systems I: Fundamental Theory and Applications **49**, 602 (2002).
[16] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, Proceedings of the National Academy of Sciences **101**, 4781 (2004).
[17] M. Chaves, R. Albert, and E. D. Sontag, Journal of theoretical biology **235**, 431 (2005).
[18] A. Wagner, Proceedings of the National Academy of Sciences **102**, 11775 (2005).
[19] S. Braunewell and S. Bornholdt, Journal of Theoretical Biology **245**, 638 (2007).
[20] J. Macia and R. V. Solé, Journal of the Royal Society Interface **6**, 393 (2009).
[21] S. Braunewell and S. Bornholdt, Journal of theoretical biology **258**, 502 (2009).
[22] W.-B. Lee and J.-Y. Huang, FEBS letters **583**, 927 (2009).
[23] S.-i. Kinoshita and H. Yamada, in *2010 10th International Symposium on Communications and Information Technologies* (IEEE, 2010) pp. 839–843.
[24] K. Mangla, D. L. Dill, and M. A. Horowitz, PLoS One **5**, e8906 (2010).
[25] V. Sevim, X. Gong, and J. E. Socolar, PLoS Comput Biol **6**, e1000842 (2010).
[26] Y. Katzir, Y. Elhanati, I. Averbukh, and E. Braun, Physical biology **10**, 066001 (2013).
[27] J. Norrell and J. E. Socolar, Physical Review E **79**, 061908 (2009).
[28] E. Gehrmann and B. Drossel, Physical Review E **82**, 046120 (2010).
[29] F. Ghanbarnejad and K. Klemm, Physical review letters **107**, 188701 (2011).
[30] E. Ackermann, T. P. Peixoto, and B. Drossel, New Journal of Physics **14**, 123029 (2012).
[31] T. Helikar, B. Kowal, S. McClenathan, M. Bruckner, T. Rowley, A. Madrahimov, B. Wicks, M. Shrestha, K. Limbu, and J. A. Rogers, BMC systems biology **6**, 1 (2012).
[32] T. Helikar, B. Kowal, and J. Rogers, Clinical Pharmacology & Therapeutics **93**, 393 (2013).
[33] L. Glass, Journal of Theoretical Biology **54**, 85 (1975).
[34] E. Ortiz-Gutiérrez, K. García-Cruz, E. Azpeitia, A. Castillo, M. de la Paz Sánchez, and E. R. Álvarez-Buylla, PLoS Comput Biol **11**, e1004486 (2015).
[35] D. Irons, Journal of theoretical biology **257**, 543 (2009).
[36] D. A. Orlando, C. Y. Lin, A. Bernard, J. Y. Wang, J. E. Socolar, E. S. Iversen, A. J. Hartemink, and S. B. Haase, Nature **453**, 944 (2008).
[37] G. Kervizic and L. Corcos, BMC systems biology **2**, 1 (2008).
[38] M. I. Davidich and S. Bornholdt, PloS one **3**, e1672 (2008).
[39] M. Brandon, B. Howard, C. Lawrence, and R. Laubenbacher, BMC systems biology **9**, 1 (2015).
[40] Ö. Sahin, H. Fröhlich, C. Löbke, U. Korf, S. Burmester, M. Majety, J. Mattern, I. Schupp, C. Chaouiya, D. Thieffry, *et al.*, BMC systems biology **3**, 1 (2009).
[41] S. Sridharan, R. Layek, A. Datta, and J. Venkatraj, BMC genomics **13**, 1 (2012).

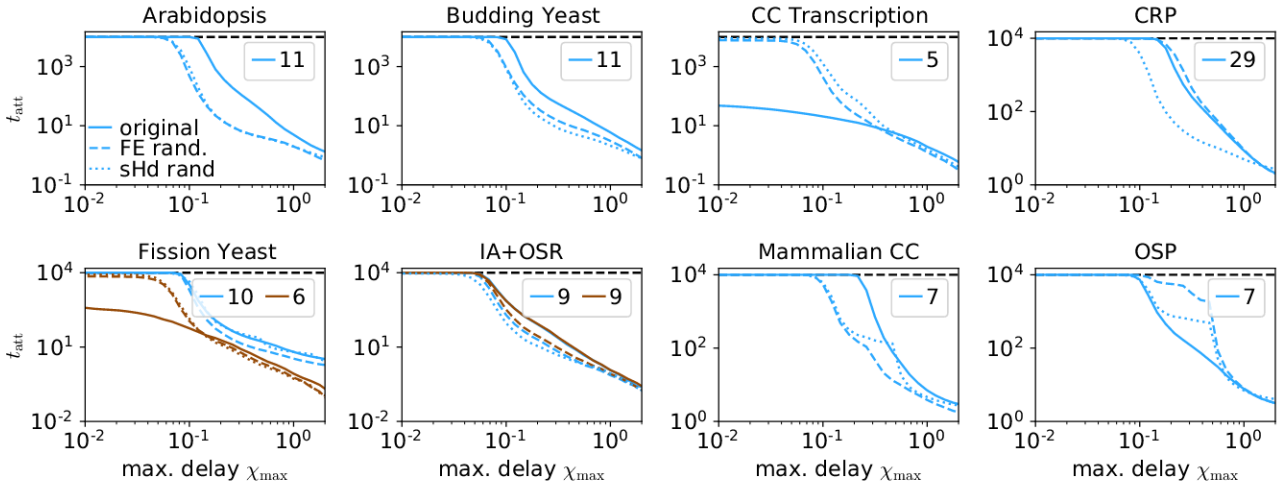


FIG. 3. Average time t_{att} networks remain in their attractors under noise as a function of maximum signal transmission delay $\chi_{\max}$, with a cutoff at $t_{\max} = 10^4$ (black dashed line). 10^4 different attractors per network were used. Different colors denote different attractors, and the legends show the attractor lengths in order of frequence of appearance. Very rare attractors which were found with less than 1 % of initializations which led to cyclic attractors are not shown. Solid lines show the original genetic networks' stabilities, dashed lines show their functionally equivalent (FE) randomized variants, and dotted lines show the randomized with similar Hamming distances (sHd) variants.

Supplemental Material to: Robustness to noisy signal transmission delays in genetic networks

Lorenz Baumgarten* and Stefan Bornholdt†

Institut für Theoretische Physik, Universität Bremen, 28759 Bremen, Germany
(Dated: February 7, 2022)

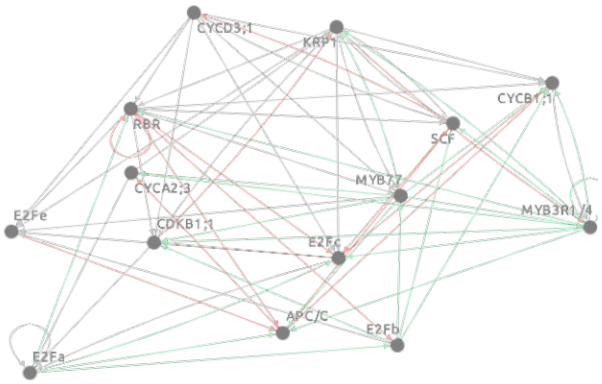
TABLE I: Gene activation rules for the fission yeast network.

Gene	Activation rule
Start	Ste9 ∧ ¬(Start ∨ Sk ∨ PP)
SK	Start
Ste9	((PP ∧ Ste9) ∧ ¬((Cdc2/Cdc13* ∧ Sk) ∨ (Cdc2/Cdc13* ∧ Cdc2/Cdc13) ∨ (Cdc2/Cdc13 ∧ Sk))) ∨ ((PP ∨ Ste9) ∧ ¬(Cdc2/Cdc13* ∨ Sk ∨ Cdc2/Cdc13))
Cdc2/Cdc13	¬(Ste9 ∨ Rum1 ∨ Slp1)
Rum1	((PP ∧ Rum1) ∧ ¬((Cdc2/Cdc13* ∧ Sk) ∨ (Cdc2/Cdc13* ∧ Cdc2/Cdc13) ∨ (Cdc2/Cdc13 ∧ Sk))) ∨ ((PP ∨ Rum1) ∧ ¬(Cdc2/Cdc13* ∨ Sk ∨ Cdc2/Cdc13))
PP	Slp1
Cdc25	(Cdc2/Cdc13 ∧ Cdc25) ∨ ((Cdc2/Cdc13 ∨ Cdc25) ∧ ¬PP)
Slp1	Csc2/Csc13*
Cdc2/Cdc13*	Cdc25 ∧ ¬(Ste9 ∨ Rum1 ∨ Slp1 ∨ Wee1/Mik1)
Wee1/Mik1	(PP ∧ Wee1/Mik1) ∨ ((PP ∨ Wee1/Mik1) ∧ ¬ Cdc2/Cdc13)

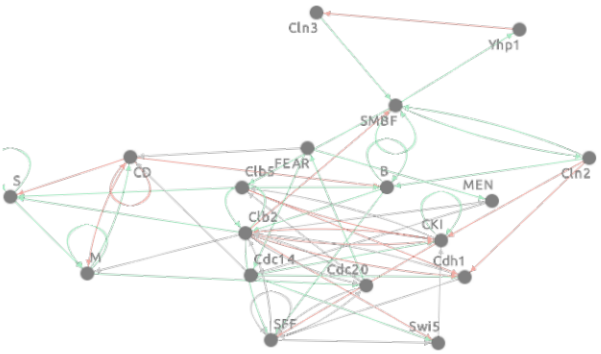
* lbaumgarten@itp.uni-bremen.de

† bornholdt@itp.uni-bremen.de

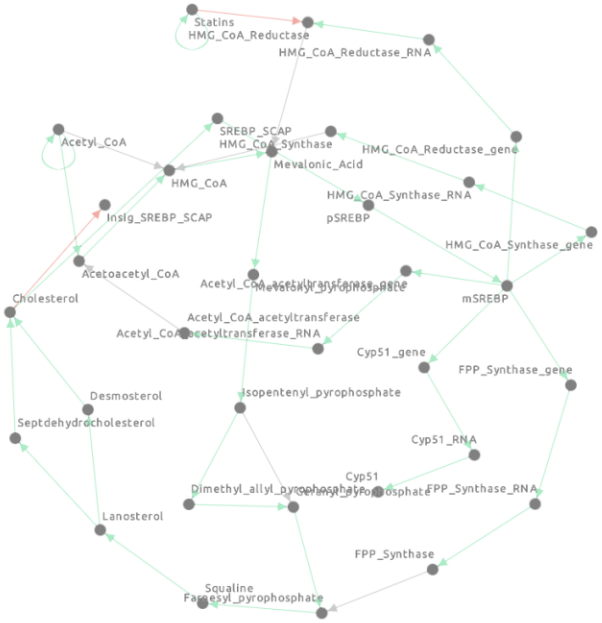
Arabidopsis



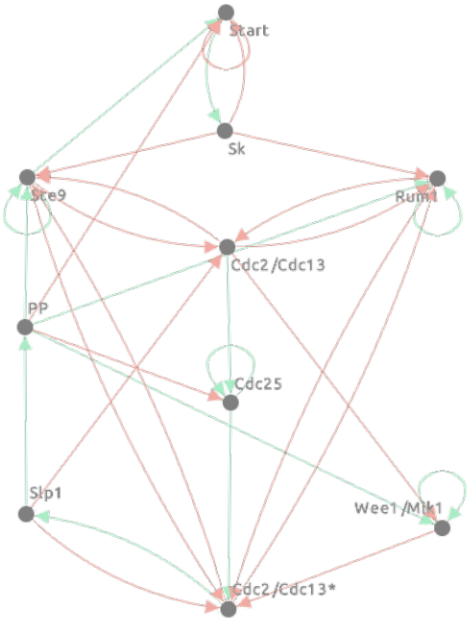
Budding Yeast



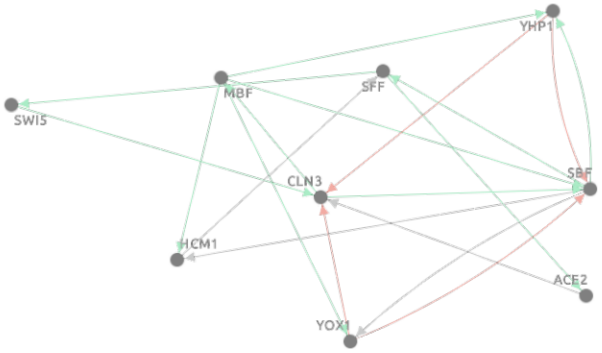
CRP network



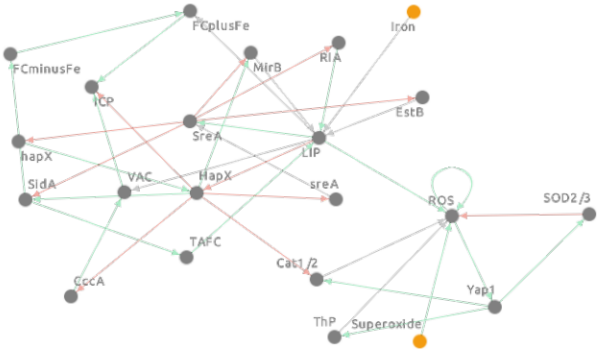
Fission Yeast



CC Transcription



IA+OSR



Mammalian CC

OSP network

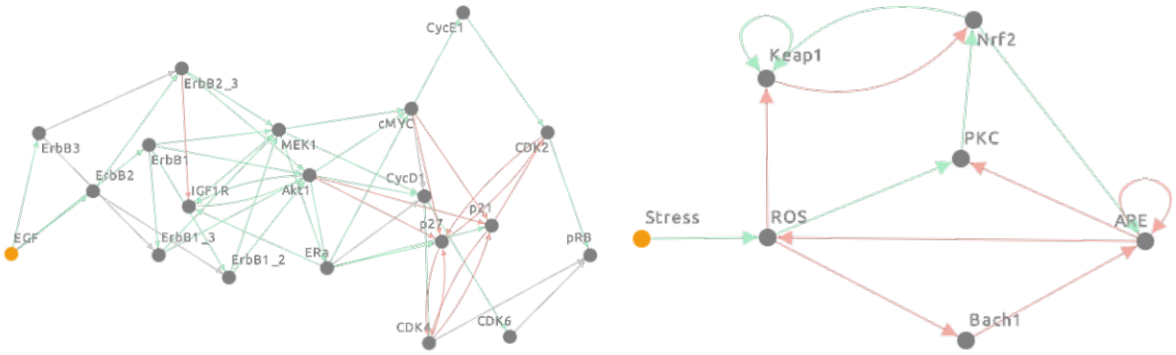


FIG. 9: Architectures of the studied networks; graphics created using cellcollective.org.

Universal computation using localized limit-cycle attractors in neural networks

Lorenz Baumgarten and Stefan Bornholdt. "Universal computation using localized limit-cycle attractors in neural networks". In: *International Journal of Unconventional Computing, accepted for publication, arXiv preprint arXiv:2112.05558* (2022).

Supplemental movies for this paper can be found at `https://arxiv.org/src/2112.05558v1/anc`.

Universal computation using localized limit-cycle attractors in neural networks

Lorenz Baumgarten* and Stefan Bornholdt†

Institut für Theoretische Physik, Universität Bremen, 28359 Bremen, Germany

(Dated: December 13, 2021)

Neural networks are dynamical systems that compute with their dynamics. One example is the Hopfield model, forming an associative memory which stores patterns as global attractors of the network dynamics. From studies of dynamical networks it is well known that localized attractors also exist. Yet, they have not been used in computing paradigms.

Here we show that interacting localized attractors in threshold networks can result in universal computation. We develop a rewiring algorithm that builds universal Boolean gates in a biologically inspired two-dimensional threshold network with randomly placed and connected nodes using collision-based computing. We aim at demonstrating the computational capabilities and the ability to control local limit cycle attractors in such networks by creating simple Boolean gates by means of these local activations. The gates use glider guns, i.e., localized activity that periodically generates "gliders" of activity that propagate through space. Several such gliders are made to collide, and the result of their interaction is used as the output of a Boolean gate. We show that these gates can be used to build a universal computer.

I. INTRODUCTION

Computation in nature occurs in highly irregular environments that differ significantly from regular human-constructed computation methods. In this spirit, the field of unconventional computing [1–3] explores alternative methods of computation to the ubiquitous von-Neumann architecture of modern computers. A common and promising strategy is using biology as inspiration for new computation schemes, as in the field of neuromorphic computing [4], and the sub-field of amorphous computing with its large numbers of irregularly spatially distributed, unreliable, and locally communicating parts [5–7]. Such irregularly placed and only partially connected parts can, for example, be found in neural brain networks.
Unconventional computing schemes find uses in a variety of fields such as managing robot swarms [8, 9], engineering biological devices [10], medical image analysis [11], or information storage [12], and a multitude of other ideas, such as cellular neural networks [13], for example, have been developed.

In particular, highly parallelizable computation networks, such as memristor networks [14–17], that can be trained like artificial neural networks [18, 19], appear highly promising. We take recent advances in this field as inspiration for creating a new unconventional computing scheme in irregular, randomly constructed neural networks.

A major mechanism of computation in neural networks is computing with attractors, where the global attractors of the dynamical network represent the result of a computation [20, 21]. This computing paradigm is perhaps best exemplified by the Hopfield model [22] in which patterns are stored as global attractors of the network dynamics. It has long been discussed that computation in the brain takes advantage of using attractors, including non-fixed point (or limit-cycle) attractors [23].

One prominent property of attractors in asymmetric neural networks is that, under certain circumstances, they may occur as localized excitations. Such localized attractors, or localized persistent activity, have been observed in neural networks [24–35], and have been discussed in diverse systems, such as genetic networks [36–38] and immune networks [39–41],

We here expand the idea of attractor computation to co-existing, localized attractors. In an example system, we use multiple spatially localized periodic (or limit-cycle) attractors, as opposed to the conventionally used global attractors in artificial neural networks such as the Hopfield model.

As a proof of concept, to demonstrate the possibility of localized attractor computation in irregular neural networks, we will make use of collision-based computing, which utilizes moving particle-like localized activity islands, as have been observed in attractor neural networks in [35]. We do not, however, suggest that the algorithm and resulting dynamics described in this paper accurately reflect a brain's function; we merely propose a biologically inspired new unconventional computing method.
Collision-based computing is the computation of signals propagating through space, usually called gliders, solitons, or wave-fragments depending on context, by interaction on impact with each other or obstacles. It is the subject of research in a variety of different systems such as non-linear [42–44] and chemical media such as the Belousov-Zhabotinsky medium [45–49] and liquid marbles [50], and biological systems such as biopolymers [51–53] and slime molds [54–56], see [2, 3, 57–59] for reviews.
The field emerged in the wake of Fredkin and Toffoli's paper [60] introducing the idea of a ballistic computer—the billiard ball model—, in which Boolean logic gates

* lbaumgarten@itp.uni-bremen.de
† bornholdt@itp.uni-bremen.de

were implemented by collisions between billiard balls and reflectors; Margolus' following paper [61] creating a cellular automaton implementation of the billiard ball model; and Berlekamp, Conway, and Guy creating Boolean logic gates using gliders in the game of life [62]. Since then, various other collision-based computing schemes for cellular automata [42–44, 63–69] or in preconstructed mazes [70] have been developed . Unlike our systems, however, these automata operate on regular lattices.
We demonstrate how limit cycles can be manipulated by rewiring algorithms to achieve desired results. For this, we will create Boolean gates operating on limit cycle glider guns and show that universal computation using these gates is possible.

II. MODEL

We study a network of N nodes randomly distributed in a two-dimensional square of space whose side length we define as 1. The nodes have directed connections between each other in such a way that the probability P of a connection existing from node A to node B is proportional to an exponential function

$$P(d) = K \exp(-\lambda d)$$

of the distance d between A and B. The parameters K and λ are chosen to result in specific values for the average degree k and the clustering coefficient C. We choose a relatively high clustering coefficient and average degree due to our observations of localized attractors in the networks we studied in [71].
Nodes are either excitatory or inhibitory, meaning that, if they are active, they send a positive or negative signal to all nodes they have efferent connections to. A node i's state σ_i is determined by its incoming signal

$$S_i = \sum_j c_{ij}\sigma_j(t)$$

via

$$\sigma_i(t+1) = \begin{cases} 1 & \text{if } S_i > h \\ 0 & \text{otherwise} \end{cases},$$

where c_{ij} is ± 1 if there is a connection from node j to node i and zero otherwise, and h is the threshold. All nodes are updated synchronously in discrete time steps.

In all our simulations, we use an initial network with $N = 2000$ nodes, threshold $h = 2$, average degree $k = 10$, clustering coefficient $C \approx 0.4$, and a chance of nodes being excitatory or inhibitory of 50 % each. In Movie S1 in the supplemental material, we show an animation of localized attractors in a similar, untrained random network.
To create logic gates, we will encode incoming signals in glider guns that periodically produce propagating patterns called gliders. These gliders will collide and interact with each other to produce a desired output.

Let us first discuss how glider guns are created and afterwards discuss two different strategies to utilize these glider guns for logic gates.

III. GLIDER GUNS

To create a glider gun, we denote a node as an input node whose state will be defined from outside instead of by the network dynamics and which will serve to activate a glider gun, meaning it will periodically produce an activation that will propagate through space. This node will send a signal of strength $h + 1$ instead of strength one to the nodes it is connected to, i.e., $c_{ij} = h + 1$ or $c_{ij} = 0$ where node j is the input node, so that its signal is sufficient to activate nodes in its vicinity given no other incoming signals.
We also define a target point towards which the glider will move with constant velocity within T time steps. The glider need not necessarily stop at the target point; therefore, the target point only defines a glider's direction and speed, not its destination. Throughout this paper T will be chosen as $T = 10$.
To set a glider gun's corresponding period, we rewire connections in an area around the input node randomly and measure the period of the limit cycle that is reached when initially only the input node is active. If the resulting period is further from the desired result than before rewiring, the rewiring is undone. This is repeated until the desired period is produced. Here, and throughout this paper, rewiring is done by choosing two connections and swapping their target nodes [72], so long as that does not result in redundant connections between two nodes, preserving all nodes' degrees. Rewirings are also only done if they do not result in connections above a certain length L to preserve the network's spatial character.
To now create gliders, we divide space into three regions: region I in which we do not want activity, region II in which we do want activity and region III where anything is allowed to happen. We define a fitness function f as

$$f = \frac{\sum_{\mathcal{A}} \sum_i g(S_i, x_i, y_i)}{|\mathcal{A}|} \text{ with}$$

$$g(S, x, y) = \begin{cases} \min(h - S, 0) & \text{if } (x, y) \text{ in I} \\ \max(S - (h+1), 0) & \text{if } (x, y) \text{ in II} \\ 0 & \text{otherwise} \end{cases}.$$

where (x_i, y_i) are the node i's coordinates, $\mathcal{A}$ is the set of network states in the network's limit cycle, and $|\mathcal{A}|$ is the number of states in the limit cycle.
Now the three regions need to be defined: Region III is the region immediately around the input node, with a radius D, for which we choose $D = 0.07$ in our simulations. The length D also governs the maximum length of formed connections $L = 3D$. For the glider gun to periodically produce gliders, some periodic activity is required, and it does not make sense to promote or suppress activity here.

Region II are the gliders themselves. It consists of areas of radius D that are periodically created at the input node and move with fixed velocity towards and past the target point. The initial position of these areas is chosen to maximize the fitness produced by region II. If, for example at the start of a new glider shot, region II and III overlap, nodes in the overlap are counted as in region II. Region I is the rest of space.
Now, we start with a completely deactivated network and activate the input node. Then, the network dynamics are run until a limit cycle is reached, and the fitness within the limit cycle is calculated. Afterwards a rewiring operation is done, and the previous calculation is repeated with the same starting conditions. If the limit cycle's period changes, no limit cycle is found within a set amount of time steps, the network does not return to an inactive state after deactivating all input nodes at a random point in the limit cycle, or the fitness is lower after rewiring, the rewiring is undone. This is repeated until a satisfactory glider gun has been created. Because we want our computations to function regardless of when input nodes are activated or deactivated, between two of such rewiring attempts, we deactivate the input node and wait a random number of time steps before reactivating the input node and measuring fitnesses. Here, and in the rest of this paper, a random number of time steps is always a number between zero and the end of the first limit cycle that is reached.
An example of such a glider gun is shown in Figure 1.

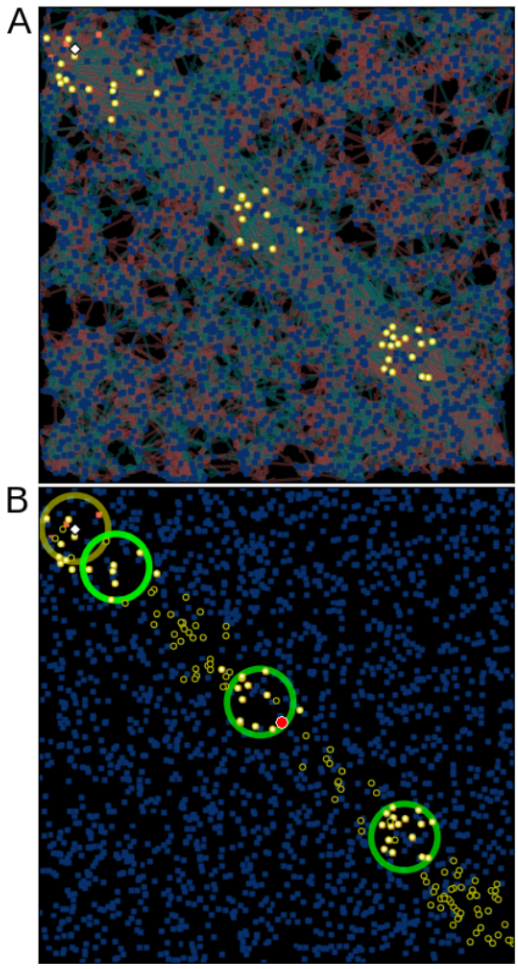


FIG. 1. A glider gun with period 7 created by the algorithm discussed above. Blue and red squares are nodes that are either off or on throughout the entire limit cycle, respectively. Yellow circles indicate nodes that change their state at least once during the limit cycle and are filled if the node is currently on; currently inactive nodes that change their state are omitted in (A). White diamonds indicate active input nodes. In (A), the underlying connections are shown: Red and blue arrows indicate excitatory and inhibitory connections, respectively. In (B), region II is marked by a yellow ring around the input, region III is marked by green rings, and the target point is indicated by a red dot.

IV. LOGIC GATES

In this section, we will discuss how to utilize glider guns to build logic gates. For this, we have developed two different strategies.
Both strategies use multiple input nodes activating glider cannons that aim at the same target point. The gliders will therefore collide at the target point and interact with each other. This interaction will define the output.
We want our gates to function regardless of when the input nodes are activated, and therefore we use glider cannons with different prime number periods. The reasoning behind this is that, using two primes, all possible phase differences between a state of one glider gun and a state of another glider gun will occur at some point.
This is not exactly correct since, once the glider guns interact with each other, their periods will not be as clear cut as before. Instead, a macro limit cycle involving the dynamics of all active glider guns will be created. In an effort to force the individual glider guns to retain their periodic behavior, we will only accept rewiring if, when multiple glider guns are active, the macro limit cycle's period is the same period as if the individual glider guns were not interacting with each other. This macro period is the least common multiple of individual periods, which is, since we are using prime numbers, the product of the periods.

In the first strategy, we define one output for every input node. An output is counted as TRUE if the glider passes the target point and FALSE otherwise. In the second strategy, we define a common output region with width D for multiple glider cannons. Since for a common output the glider gun signals have to merge at the target point and since the glider guns have different periods, it is not clear which period the output signal should have. Therefore, instead of a signal moving towards the output region, we will assign the area between the target point and output region statically to region II if a positive output is required.

For both these strategies, the area of radius D around the target point is also added to region III, meaning any behavior is permitted here. This allows, for example, the signal from one glider gun to remain within this region to then catch a signal from another glider gun and interact with it without the need for the signals to arrive simultaneously.
Also, for strategy two and for gliders in strategy one whose output is either supposed to be FALSE or for whom no desired output is defined, the glider shots are terminated at the target region, while these shots' region II is overwritten by the target region's region III. This means that these shots are only forced into existence outside the target region. This, for example, makes it easier for an AND-interaction to occur because otherwise both signals would compete for activating all nodes in the interaction region by themselves, as opposed to only in the case when both signals are present.
Both these strategies have advantages and disadvantages: For strategy one, if the desired output requires only one input node to be active, say an $A\neg B$ gate, the resulting output has the period of the active glider gun and can therefore simply be routed towards another gate for further computations. Since the $A\neg B$ gate is universal, any Boolean operation can be created using this principle. On the other hand, if an output requires multiple symbols to be active, the resulting output signal will in general have the rather large and unwieldy period of the macro limit cycle. This output will likely need to be read out and converted into a new input signal to start a new glider gun. This could be accomplished by simply setting nodes at the glider's end to permanently be in region II and therefore be able to permanently activate a glider gun—fig. 2 (D) shows that permanent activity of nodes in such an area is possible—, given that the previous input remains, or by defining an input node that only turns off when it has not received a signal for a period of time longer than the macro limit cycle.
For strategy two, the same period length problems apply. For this strategy's advantages, let us discuss how one would create an XOR-gate using the two separate strategies:
For strategy two, the common output region can simply be trained to output XOR, removing the need to create more complicated circuits. For strategy one, an XOR-gate cannot be realized on one of the outputs since the output belonging to an input node A can only be TRUE if there is an incoming signal from input node A. Therefore, one has to either reroute the outputs of an $A\neg B$- and a $B\neg A$-function to the same output region and add them together or use the possible gates, for example the universal $A\neg B$ gate, to build a circuit with an XOR output.
Fortunately, multiple gates can easily be combined. The direction and speed of different gliders in our algorithm is simply constant for convenience's sake; however, nothing dictates that a glider cannot change direction or speed, and therefore it is easily possible to reroute signals to arbitrary points in the network or to delay or accelerate them, should it be required.

V. ALGORITHM

The algorithm to create gates is similar to the one for creating glider guns, but needs to be expanded to deal with various issues that can occur when multiple input nodes are active.
Note that we will distinguish between inputs and input nodes. An input is one combination of active or inactive input nodes.
Firstly, we need to ensure that the gate works correctly for all possible inputs, so the calculation of the network's fitness will now consist of activating some combination of input nodes, measuring the fitness for this input, and repeating this process for all possible inputs (excluding all input nodes being inactive). Again, between different inputs, all input nodes are deactivated and the network dynamics are run for a random number of time steps before the next input is activated. The final fitness is then the sum of fitnesses for the individual inputs.
One important property we want our gates to have is for them to function regardless of when and in which order input nodes are activated. Therefore, instead of simultaneously activating all input nodes in a specific input, individual input nodes are activated in random order and with a random number of time steps between them. With the random number of time steps any possible glider gun interaction, during the previously active guns' transient dynamics or within the limit cycle, can occur.
Because of this large number of possible activation patterns, it is unreasonable to calculate the fitness for all of them for every rewiring attempt; instead, we only calculate whether the fitness increases for one set of activation patterns per rewiring attempt. This, unfortunately, may lead to rewirings worsening the fitness for different activation patterns. This can also lead to the macro limit cycle's period changing or the network not returning to an inactive state without inputs. When it is detected that either of those two happened, previous rewirings are sequentially undone in reverse order until the problem no longer occurs for the activation pattern for which this was detected.
Another issue that may occur is that some activation patterns may have a significantly lower fitness than others, and a rewiring that improves this pattern will often lower other activation patterns' fitnesses to a similar value. To avoid this, an activation pattern that has a significantly lower fitness than the previous pattern will be skipped.
To speed up the rewiring, when searching for a valid rewiring step, activation patterns are reused until a rewiring step that actually improved — instead of just preserving — the fitness is found, so as to not be forced to recalculate the fitness before rewiring at every step.
When choosing connections to rewire, one of the connections chosen has to originate from a node that at any

point during the calculation of the fitness, during the transient or the limit cycle, has been active to further speed up the algorithm, since rewiring connections that do not transmit any signal has no effect.
Additionally, not all connections in the network are considered for rewiring. Instead, the lowest distance in the direction to the target point that any cannon shot has reached during any of the inputs normalized by the distance between the corresponding input node and the target point is calculated. Only connections which lead to nodes within a region depending on this distance are considered for rewiring. The algorithm alternates between choosing this region as a region around the points that lie at this minimum distance in the direction from the input nodes to the target point with radius D and as the entire path of the gliders up to those points. When calculating this distance, cannon shots that are not meant to pass the target point are disregarded as long as they get close enough to the target point. For shared outputs, once all cannon shots get close enough to the target point, the minimum reached distance from the target point to the end of the output region for an input with desired output TRUE is used instead. Alternating between these two regions has the advantage that, for the region around the minimum distance point, there is a good chance for the rewiring to result in the cannon shot traveling farther after rewiring while the other region can optimize the path that has already been created.
Also, and this is vital for the algorithm to function, by only rewiring up to the lowest distance reached, when multiple cannon shots have to pass the target point, a situation in which one cannon shot already reaches far past the target point while the other has not passed the target point yet is not created. In such a situation, any rewiring around the target point necessary to make the second shot pass the target point, that would negatively affect the first shot would significantly lower the fitness because it would cut off the first shot significantly earlier than before while only slightly increasing the distance that the second shot travels. In such a situation, it is difficult to find a rewiring that improves the second shot without ruining the already established first shot.
Finally, when activation patterns are skipped because they have a significantly lower fitness than previous patterns, after skipping configurations 100 times in a row, it is assumed that something has gone wrong and previous rewirings are sequentially undone similar to when an activation pattern results in the wrong period, until the problem does not occur any longer.

VI. RESULTS

In this section, we will present the results of an AND-gate on both outputs and an $A\neg B$-gate on one output and a $B\neg A$-gate on the other one for strategy one as well as an AND- and an XOR-gate for strategy two. Snapshots for all these gates are shown in Figure 2, and the fitness as well as the error rate as a function of rewiring attempts is shown in Figure 3. Animations of these gates can be found in the supplemental material, movies S2–S13.

By periodically measuring error rates under the same conditions used during rewiring, i.e., activation of input nodes at random times and in random orders and with random numbers of time steps between inputs, and stopping the rewiring algorithm when minimal error rates are achieved, it is easily possible for all these gates to achieve perfect performance. The results of an input is counted as an error if any of the outputs is not the desired result, if the macro limit cycle's period differs from the intended period, or if the network would not return to the deactivated state after deactivating all input nodes. When measuring error rates, all possible inputs are used equally frequently, except for the zero input, which is already implicitly covered by the third error condition.

VII. CONCLUSION

We have demonstrated the possibility of computation with attractors in irregular two-dimensional threshold networks. For this, we constructed a rewiring algorithm that enables us to control the behavior of localized limit cycle attractors within such networks. With this algorithm, we first created glider guns to propagate signals in space and then used these glider guns to build Boolean gates. We have developed two strategies for such gates, both involving the collision of multiple gliders from different glider guns. In the first strategy, every glider gun has its own associated output, whereas in the second strategy, the entire gate only has one common output.
We have built multiple Boolean gates with either of these strategies and argued that these gates can easily be combined to build a universal computer. We have also demonstrated that these gates can achieve perfect performance, in the absence of noise, even given random activation and deactivation times of the incoming inputs. This is, to our knowledge, the first application of localized activity in such networks, and we hope that it may therefore be useful to gain insight on the operation of brain networks in which localized activity as a response to external stimuli can also be observed.
Further, the computation method described in this paper is merely one option for utilizing localized attractors for computation in threshold networks. A number of different computation schemes are also conceivable and may hopefully be explored in the future. We hope that this simple demonstration can spark new ideas for amorphous computation schemes using localized activity in neural network structures.
The gates shown here would most likely not work if the updates to the nodes were not synchronized or if the signals or node states were subject to noise, and neither such synchronization nor a noise-free environment are to be expected in real-world applications. However, in bi-

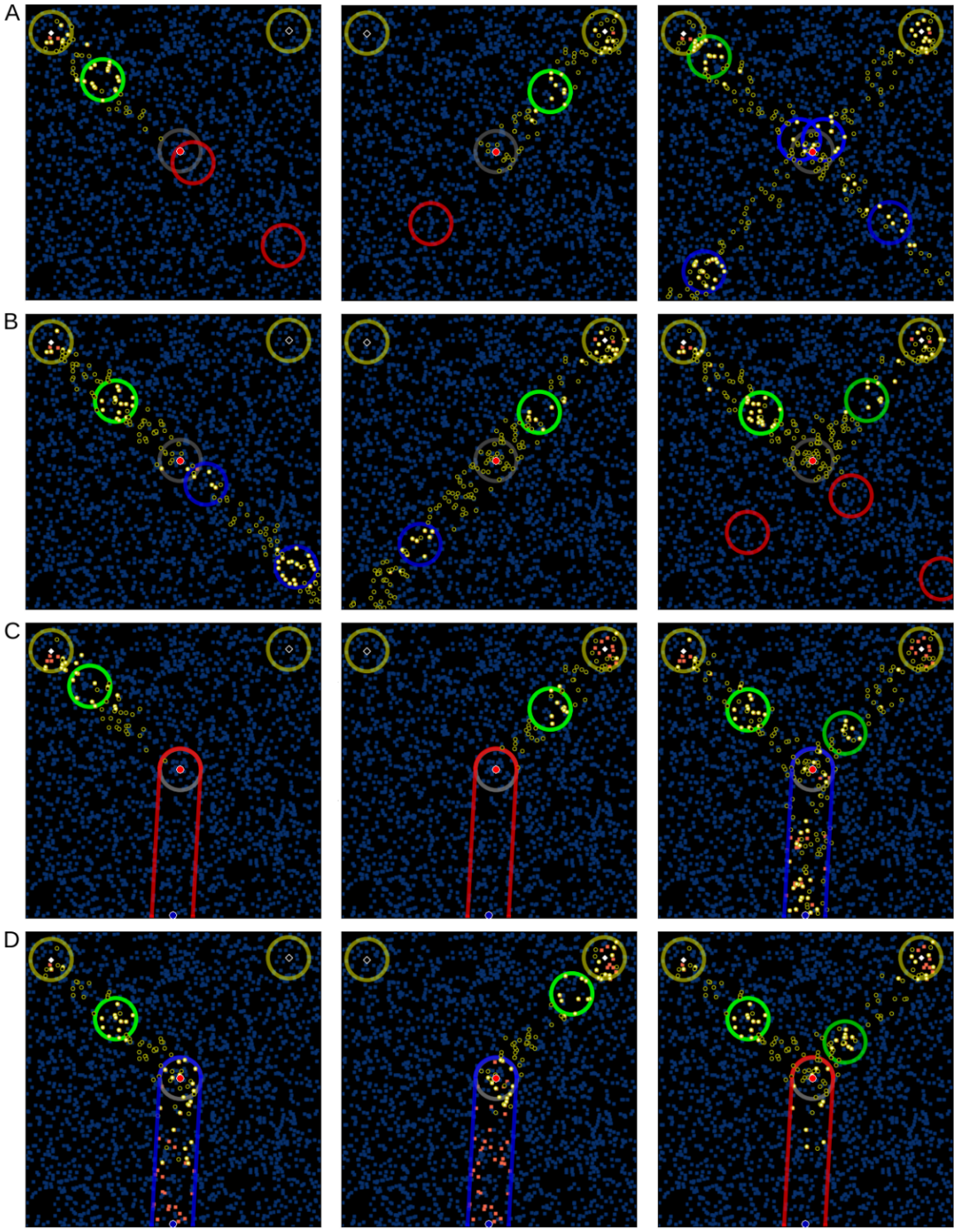


FIG. 2. Snapshots of (A) an AND-gate for both input nodes, (B) a gate with an $A\neg B$ output for input node A and a $B\neg A$ output for input node B, (C) a common AND output, and (D) a common XOR output. In all of these, the left glider gun has period seven, and the right one has period eleven. Black diamonds denote deactivated input nodes. Gray circles indicate the region III around the target point. In (A) and (B), red rings merely indicate where a shot had been, had it not been stopped at the target point, and belong to region I, and blue rings indicate shots that were supposed to pass the target region and belong to region II. In (C) and (D), red and blue lines indicate the areas that are fixed as regions I or II, respectively, and blue circles mark the center of the output region. Remember that gray rings overwrite green and red areas and are in turn overwritten by blue areas. The movies S2–S13 in the supplemental material show animations of the gates shown.

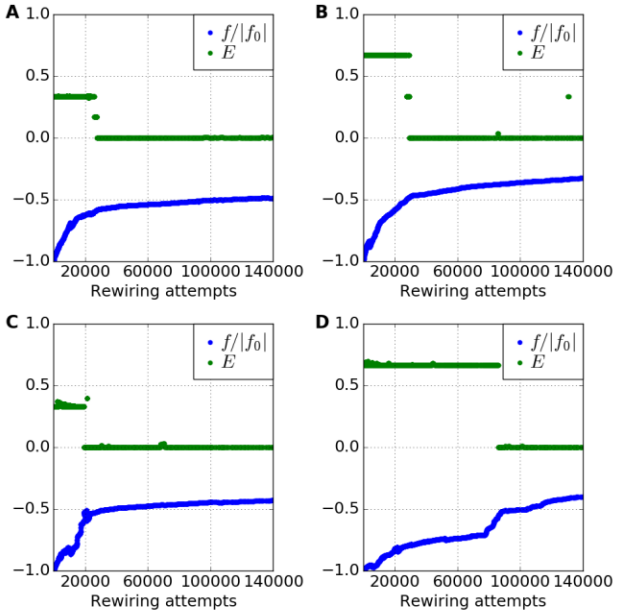


FIG. 3. Fitness f, normalized by initial fitness f_0, and error rate E as a function of rewiring attempts for (A) an AND-gate for both input nodes, (B) a gate with an $A\neg B$ output for input node A and a $B\neg A$ output for input node B, (C) a common AND output, and (D) a common XOR output. The initial error rate for example for (A) is 1/3 because initially the cannon shots cannot reach the output region, which is the wanted result for inputs in which only one input node is active. Therefore, the result is initially correct for two out of three possible inputs.

ology, genetic networks can reliably function under such conditions [73, 74], and another interesting model for reliable behavior from noisy elements has been demonstrated recently with the game of life, a cellular automaton with gliders similar to those used in this work, which has been successfully implemented to reliably function in a noisy environment [75]. Both these findings point towards the possibility of reliably managing network dynamics like the ones presented in this paper in the presence of noise. Besides the question of noisy implementations of our model, a second line of possible future research is the evolutionary creation of the network itself. The algorithm presented here is a stepwise evolutionary algorithm, using mutation and subsequent selection in an overall algorithmic process. An interesting question is how a developmental algorithm, perhaps on the basis of only locally available information, could address the problem.

[1] T. Toffoli, in *Encyclopedia of Electrical and Electronics Engineering*, edited by J. Webster (Wiley & Sons, 1998) pp. 455–471.

[2] A. Adamatzky, *Advances in Unconventional Computing, Volume 1: Theory*, Emergence, Compexity and Computation, Vol. 22 (Springer, 2017).

[3] A. Adamatzky, *Advances in Unconventional Computing, Volume 2: Prototypes, Models and Algorithms*, Emergence, Compexity and Computation, Vol. 23 (Springer, 2017).

[4] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, preprint arXiv:1705.06963 (2017).

[5] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. K. Jr., T. F, R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss, Communications of the ACM **43**, 74 (1995).

[6] R. Nagpal and M. Mamei, in *Methodologies and Software Engineering for Agent Systems*, Multiagent Systems, Artificial Societies, and Simulated Organizations, Vol. 11, edited by F. Bergenti, M. Gleizes, and F. Zambonelli (Springer, 2004).

[7] H. Abelson, J. Beal, and G. Sussman, in *Encyclopedia of Complexity and Systems Science*, edited by R. Meyers (Springer, 2009).

[8] M. Otte, The International Journal of Robotics Research **37**, 1017 (2018), https://doi.org/10.1177/0278364918779704.

[9] H. Hamann, G. Valentini, and M. Dorigo, in *Swarm Intelligence*, edited by M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle (Springer International Publishing, Cham, 2016) pp. 173–184.

[10] J. Macia, R. Manzoni, N. Conde, A. Urrios, E. de Nadal, R. Solé, and F. Posas, PLOS Computational Biology **12**, 1 (2016).

[11] S. Mitra and B. U. Shankar, Information Sciences **306**, 111 (2015).

[12] M. Nugent, R. Porter, and G. Kenyon, Physica D: Nonlinear Phenomena **237**, 1196 (2008), novel Computing Paradigms: Quo Vadis?

[13] L. O. Chua and L. Yang, IEEE Transactions on Circuits and Systems **35**, 1257 (1988).

[14] R. Kozma, R. E. Pino, and G. E. Pazienza, *Advances in neuromorphic memristor science and applications*, Vol. 4 (Springer Science & Business Media, 2012).

[15] A. Adamatzky and L. Chua, *Memristor networks* (Springer Science & Business Media, 2013).

[16] I. Vourkas and G. C. Sirakoulis, IEEE Circuits and Systems Magazine **16**, 15 (2016).

[17] L. Chua, G. C. Sirakoulis, and A. Adamatzky, *Handbook of Memristor Networks* (Springer Nature, 2019).
[18] M. A. Nugent and T. W. Molter, PloS ONE **9**, e85175 (2014).
[19] C. Yakopcic, M. Z. Alom, and T. M. Taha, in *2017 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2017) pp. 1696–1703.
[20] K. Murali, S. Sinha, V. Kohar, B. Kia, and W. Ditto, PLoS One **13** (2018), e0209037.
[21] M. Sathish Aravindh, A. Venkatesan, and M. Lakshmanan, Phys. Rev. E **97**, 052212 (2018).
[22] J. J. Hopfield, Proceedings of the National Academy of Sciences **79**, 2554 (1982).
[23] J. Hertz, in *Handbook of Brain Theory and Neural Networks*, edited by M. A. Arbib (MIT Press, 1995) pp. 230–234.
[24] A. Samsonovich and B. L. McNaughton, Journal of Neuroscience **17**, 5900 (1997).
[25] B. Ermentrout, Reports on Progress in Physics **61**, 353 (1998).
[26] D. Hansel and H. Sompolinsky, "Modeling feature selectivity in local cortical circuits," (1998).
[27] P. E. Sharp, H. T. Blair, and J. Cho, Trends in neurosciences **24**, 289 (2001).
[28] X.-J. Wang, Trends in Neurosciences **24**, 455 (2001).
[29] N. Brunel, Cerebral Cortex **13**, 1151 (2003).
[30] Y. Roudi and A. Treves, Journal of Statistical Mechanics: Theory and Experiment **2004**, P07010 (2004).
[31] J. Rubin and A. Bose, Network: Computation in Neural Systems **15**, 133 (2004).
[32] H. Schrobsdorff, *Localization of Neural Activity* (Diplomarbeit, Universität Göttingen, 2005).
[33] K. Koroutchev and E. Korutcheva, Physical Review E **73**, 026107 (2006).
[34] M. S. González Rodríguez, (2011).
[35] R. Monasson and S. Rosay, Phys. Rev. E **89**, 032803 (2014).
[36] S. A. Kauffman, Physica D: Nonlinear Phenomena **10**, 145 (1984).
[37] S. A. Kauffman *et al.*, *The origins of order: Self-organization and selection in evolution* (Oxford University Press, USA, 1993).
[38] S. Kauffman, **2**, 131 (2003).
[39] G. Weisbuch, R. J. De Boer, and A. S. Perelson, Journal of Theoretical Biology **146**, 483 (1990).
[40] A. U. Neumann and G. Weisbuch, Bulletin of Mathematical Biology **54**, 699 (1992).
[41] G. Weisbuch and M. Oprea, Bulletin of Mathematical Biology **56**, 899 (1994).
[42] M. Jakubowski, K. Steiglitz, and R. Squier, Complex Syst. **10**, 1 (1996).
[43] M. Jakubowski, K. Steiglitz, and R. Squier, in *Advances in Unconventional Computing, Volume 2: Prototypes, Models and Algorithms*, Emergence, Complexity and Computation, Vol. 23, edited by A. Adamatzky (Springer, 2017) pp. 261–295.
[44] G. Martínez, A. Adamatzky, F. Chen, and L. Chua, Complex Syst. **21**, 118 (2012).
[45] Á. Tóth and K. Showalter, J. Chem. Phys. **103**, 2058 (1995).
[46] R. Toth, C. Stone, B. de Lacy Costello, A. Adamatzky, and L. Bull, IJNMC **1**, 1 (2009).
[47] A. Adamatzky, Chaos, Solitons & Fractals **21**, 1259 (2004).
[48] O. Steinbock, P. Kettunen, and K. Showalter, J. Phys. Chem. **100**, 18970 (1996).
[49] B. de Lacy Costello, R. Toth, C. Stone, A. Adamatzky, and K. Bull, Phys. Rev. E **79**, 026114 (2009).
[50] T. Draper, C. Fullarton, N. Phillips, B. D. L. Costello, and A. Adamatzky, Mater. Today **20**, 561 (2017).
[51] S. Siccardi, J. Tuszynski, and A. Adamatzky, Phys. Lett. A **380**, 88 (2016).
[52] S. Siccardi and A. Adamatzky, in *Advances in Unconventional Computing, Volume 2: Prototypes, Models and Algorithms*, Emergence, Complexity and Computation, Vol. 23, edited by A. Adamatzky (Springer, 2017) pp. 309–346.
[53] D. De, T. Sadhu, and J. Das, Mater. Today-Proc. **3**, 3276 (2016).
[54] A. Adamatzky, in *Applications, tools and techniques on the road to exascale computing*, Advances in Parallel Computing, Vol. 22, edited by K. D. Bosschere, E. D'Hollander, G. Joubert, D. Padua, and F. Peters (IOS Press, 2012) pp. 41–56.
[55] J. Jones and A. Adamatzky, Biosystems **101**, 51 (2010).
[56] A. Adamatzky, Math. Comput. Model. **55**, 884 (2011).
[57] A. Adamatzky, *Collision-Based Computing* (Springer, 2002).
[58] A. Adamatzky and J. Durand-Lose, in *Handbook of Natural Computing*, edited by G. Rozenberg, T. Bäck, and J. Kok (Springer, Berlin, Heidelberg, 2012).
[59] A. Adamatzky, *Computing in Nonlinear Media and Automata Collectives* (IoP, 2001).
[60] E. Fredking and T. Toffoli, Int. J. Theor. Phys. **21**, 219 (1982).
[61] N. Margolus, Phys. D **10**, 81 (1984).
[62] E. Berlekamp, J. Conway, and R. Guy, *Winning ways for your mathematical plays, volume 2 Games in particular* (Academic Press, 1982).
[63] R. Squier and K. Steiglitz, Complex Syst. **7**, 297 (1993).
[64] L. Zhang and A. Adamatzky, Chaos, Solitons & Fractals **41**, 1191 (2009).
[65] E. Sapin, O. Bailleux, J.-J. Chabrier, and P. Collet, IJUC **3**, 79 (2007).
[66] W. Hordijk, J. Crutchfield, and M. Mitchell, in *Parallel Problem Solving from Nature — PPSN V. PPSN 1998. Lecture Notes in Computer Science, vol 1498*, edited by A. Eiben, T. Bäck, M. Schoenauer, and H. Schwefel (Springer, Berlin, Heidelberg, 1998).
[67] A. Adamatzky, Int. J. Theor. Phys. **37**, 3069 (1998).
[68] A. Adamatzky, A. Wuensche, and B. D. L. Costello, Chaos, Solitons and Fractals **27**, 287 (2006).
[69] G. Martinzes, A. Adamatzky, and K. Morita, in *Reversability and Universality*, Emergence, Complexity and Computation, Vol. 30, edited by A. Adamatzky (Springer, 2018) pp. 199–220.
[70] A. Becker, E. Demaine, S. Fekete, J. Lonsford, and R. Morris-Wright, Nat. Comput. **18**, 181 (2019).
[71] L. Baumgarten and S. Bornholdt, Physical Review E **100**, 010301 (2019).
[72] S. Maslov and K. Sneppen, Science **296**, 910 (2002).
[73] K. Klemm and S. Bornholdt, Proceedings of the National Academy of Sciences **102**, 18414 (2005).
[74] S. Braunewell and S. Bornholdt, Journal of Theoretical Biology **245**, 638 (2007).
[75] B. W.-C. Chan, preprint arXiv:1812.05433 (2018).

SUPPLEMENTAL MATERIAL

This section contains explanations of the supplementary movie files.

Movie S1

Local attractors occur in a two dimensional irregular neural network at large clustering coefficient C. We show an animation of a random network with $k = 40$, $N = 4000$, $h = 2$, $C = 0.5$, and a probability of nodes being excitatory or inhibitory of 50 % each. To better illustrate the spatially disjoint nature of the attractors, only connections between nodes are shown whose states change in the cyclical attractor.

Movies S2–S13

The movies S2–S13 show animations of the gates shown in Figure 2. The possible combinations of active input nodes are shown in a separate movie each, resulting in three movies per gate. Movies S2–S4 show a gate with an AND output for both input nodes; movies S5–S7 show a gate with an $A\neg B$ output for input node A and a $B\neg A$ output for input node B; movies S8–S10 show a gate with a common AND output; movies S11–S13 show a gate with a common XOR output.

Declaration of contributions

For all papers except for "Self-organized criticality in neural networks from activity-based rewiring", I have conducted all the research, created the figures, and written the text shown in this work, with aid in the form of discussion and text editing for the final versions from Stefan Bornholdt. For the last paper, most of the initial research was done by Stefan Landmann; I joined the project after Stefan Landmann had left the group, found an error in the initial version of the paper, redid the simulations, updated figures and the text accordingly and assisted in the editing for the final publication of the paper.

Revisions in the published version

The revisions in this version are as follows: The "Supplemental Material to: Robustness of noisy signal transmission delays in genetic networks" was titled correctly, a missing paragraph in the Kurzfassung was added, and the word "ähnlicherer" was correctly hyphenized.