Assessing the robustness of critical behavior in stochastic cellular automata

Sidney Pontes-Filho [a,b,*], Pedro G. Lind [a,c,d], Stefano Nichele [a,c,d,e,f]

[a] *Department of Computer Science, OsloMet – Oslo Metropolitan University, P.O. Box 4 St. Olavs plass, N-0130 Oslo, Norway*
[b] *Department of Computer Science, NTNU, Trondheim, Norway*
[c] *AI Lab – OsloMet Artificial Intelligence Lab, Pilestredet 52, N-0166 Oslo, Norway*
[d] *NordSTAR – Nordic Center for Sustainable and Trustworthy AI Research, Pilestredet 52, N-0166 Oslo, Norway*
[e] *Department of Holistic Systems, Simula Metropolitan Center for Digital Engineering, Pilestredet 52, N-0166 Oslo, Norway*
[f] *Department of Computer Science and Communication, Østfold University College, B R A Veien 4, N-1757 Halden, Norway*

ARTICLE INFO

ABSTRACT

There is evidence that biological systems, such as the brain, work at a critical regime robust to noise, and are therefore able to remain in it under perturbations. In this work, we address the question of robustness of critical systems to noise. In particular, we investigate the robustness of stochastic cellular automata (CAs) at criticality. A stochastic CA is one of the simplest stochastic models showing criticality. The transition state of stochastic CA is defined through a set of probabilities. We systematically perturb the probabilities of an optimal stochastic CA known to produce critical behavior, and we report that such a CA is able to remain in a critical regime up to a certain degree of noise. We present the results using error metrics of the resulting power-law fitting, such as Kolmogorov–Smirnov statistic and Kullback–Leibler divergence. We discuss the implication of our results in regards to future realization of brain-inspired artificial intelligence systems.

1. Introduction

Critical phenomena in general share two important features [1]. First, they show an infinite correlation length, which means that information spans throughout several scales, both in time and in space. Second, they occur between two well-defined phases each one assuming a specific range of values of an observable which can be "tuned" or controlled by one or more parameters. These parameters, called control parameters, can drive the system to switch between phases and for specific values, so-called critical values, one observes critical behavior. While most critical phenomena share these two ingredients, the second one is sometimes not observed, which means the system still shows correlation spanning through different scales, but there is no explicit control parameter to be tuned: independently of how we tune the system and initialize it, the system always evolves *towards* the critical state. In other words, the critical state is a stable state, an attractor of the system. This sort of critical behavior is called self-organized criticality (SOC), which is one of the most striking non-linear phenomena found in nature. Since its discovery in the 80s [2,3], several natural phenomena have been reported as showing SOC, ranging from the stock market [4–6] to the brain [7,8].

Indeed, brain functioning involves the coordination of neural activity across several scales, ranging from few neurons to large brain neural networks, leading to a natural resemblance to critical phenomena [9], and since this functioning is the "natural" state of the brain, it may be reasonable to hypothesize that this criticality follows some principles of self-organization. Recent discussions and investigations point indeed towards the possibility that SOC is the main factor for intelligence in the human brain [7] and therefore such findings from physics of critical phenomena may help to investigate how to learn from brain dynamics in order to improve the capacity of computation of artificial intelligence (AI) systems. How can self-organized criticality emerge in simple computational systems?

In this work, we address this question using cellular automata (CAs). CAs comprehend a family of models, in which a set of elementary "cells" form a lattice, typically with one or two dimensions [10]. The lattice iteratively evolves, and each cell takes one of a countable number of states. CAs are, therefore, models with discrete space, discrete time, and discrete state space.

Due to their simple implementation, they have been used to approach several complex phenomena, namely those showing large-scale correlations as a result of short-range, typically nearest neighbor, interactions [11]. The update of each cell considers

---

* Corresponding author at: Department of Computer Science, OsloMet – Oslo Metropolitan University, P.O. Box 4 St. Olavs plass, N-0130 Oslo, Norway.
E-mail address: sidneyp@oslomet.no (S. Pontes-Filho).
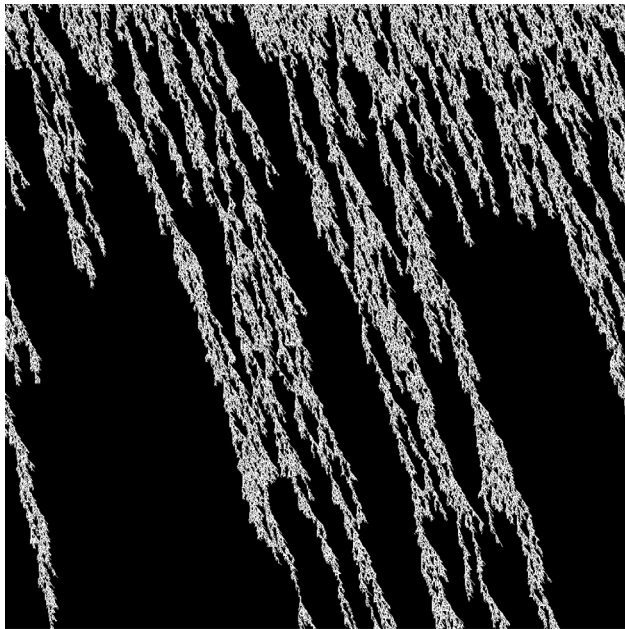
(a) Optimal stochastic CA
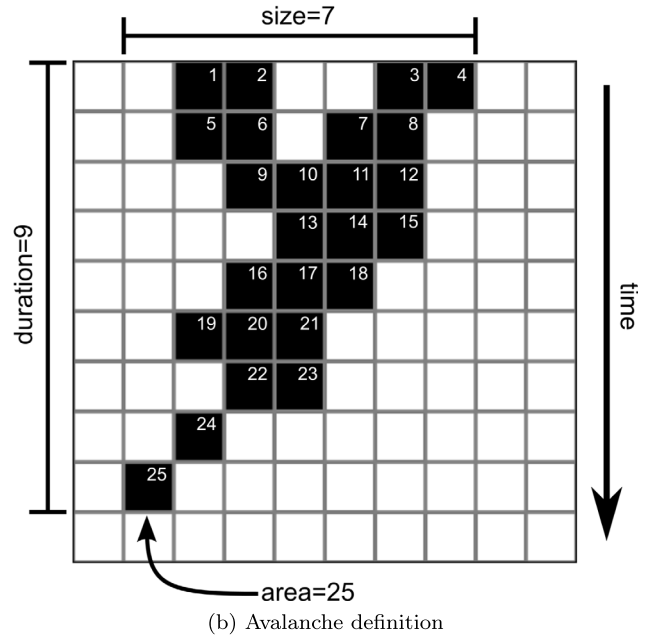
(b) Avalanche definition

Fig. 1. (a) Illustration of the evolution of an optimal stochastic CA for critical behavior. The CA started with $N = 1000$ lattice squares and its evolution is plotted for $T = 1000$ time steps from top to bottom. The number of avalanches and the distributions of this illustration is shown in Fig. 2 (first row). (b) Illustration of the three properties characterizing an avalanche: its area, defined by the total number of active sites in the same connected set; its size, given by the total number of different cells (sites) belonging to the avalanche, at least at one iteration; its duration, given by the total number of (successive) iterations that include at least one site in the avalanche. In this example, we have size of 7 cells, a duration of 9 iterations, and a total area of 25 sites. Active cells are black and have state 1. The boundary of the CA also limits an avalanche.

the composition of its state together with the state of its nearest neighbors. From the very beginning, studies on CAs have also reported the emergence of critical behavior in general [12,13] and SOC in particular [14], where a prototypical example is the so-called Game of Life [15]. Recently, the concept of SOC in Game of Life was used in the context of machine behavior to introduce collective robots with simple control and local interactions [16], as well as for discussing the general features of artificial life [17].

The rules governing the evolution of each cell and its coupling with nearest neighbors can be deterministic: state configurations of one cell and its neighboring cells impose always the same state on the cell in the next iteration. A more realistic extension of such a model is to enable the iteration throughout CA's evolution to be updated according to some rules, with a certain *probability* p not necessarily 0 or 1. Such CAs are usually called stochastic CAs [18,19]. An example of a one-dimensional stochastic CA is shown in Fig. 1(a). Indeed, if the brain functioning shows critical behavior, it should involve critical states which can be achieved through dynamical processes driven by some stochastic freedom. Stochastic CAs seem therefore to be a better choice to explore criticality in the context of artificial intelligence.

Recently [18], using a genetic algorithm (please see [20] for an introduction to genetic algorithms), the authors found eight *optimal* probabilities that grant critical behavior in a 1D stochastic CA with 3 neighbors. Due to its freedom, this stochastic CA can have different outcomes, starting from the same initial configuration of the cells composing the CA, all of them showing critical behavior. One question which remains unanswered is how *sensitive* this stochastic CA is to small changes of the optimal probabilities. The brain functioning is driven by dynamical processes with some stochasticity, but it is also robust against changes in its stochastic features. Namely, its critical behavior is observed even under changes in the stochastic dynamics. The goal of this work is to assess the *robustness* of the critical behavior in the *optimal* stochastic CA presented in [18].

We start in Section 2 by introducing the main tools and methods, namely, how the *optimal* stochastic CA is obtained, the tools to uncover the critical behavior in the evolution of one particular CA and the measures to quantitatively assess the robustness of its critical behavior. In Section 3, we present our main results, reporting a broad range of probabilities for which criticality remains in the evolution of the CA. Section 4 concludes the paper.

2. Methods

2.1. Background and prior model

A CA that is one-dimensional, 2-state, and based on local interactions evolves according to an update rule of the form

$$c_{i,t+1} = \mathcal{F}(c_{i-1,t}, c_{i,t}, c_{i+1,t}), \tag{1}$$

where $c_{i,t}$ is either 0 or 1, denoting the state of cell i in iteration t. Periodic boundary conditions, $c_{0,t} = c_{N,t}$ and $c_{N+1,t} = c_{1,t}$, are used where N is the number of cells in the CA. In general, the function $\mathcal{F}$ maps each of the 8 possible 3-tuple at iteration t into the updated state of the middle cell i in the next iteration $t + 1$. If the CA is deterministic, there are exactly $2^8 = 256$ possible choices of $\mathcal{F}$.

If the CA is stochastic, instead of such $\mathcal{F}$-functions, we define a function $\mathcal{P}$ which gives the probability for the state at cell i in iteration $t + 1$ to be *one*, given the present state of the 3-tuple defining its nearest neighborhood:

$$\mathcal{P}(c_{i-1,t}, c_{i,t}, c_{i+1,t}) = \text{Pro}\Big(\big[c_{i,t+1} = 1\big]\big|c_{i-1,t}, c_{i,t}, c_{i+1,t}\Big). \tag{2}$$

Function $\mathcal{P}$ is fully described by a vector of eight probabilities, one for each possible 3-tuple $(c_{i-1,t}, c_{i,t}, c_{i+1,t})$.

In Fig. 1(a), we present one example of a stochastic CA with periodic boundary condition, states uniformly initialized, and state
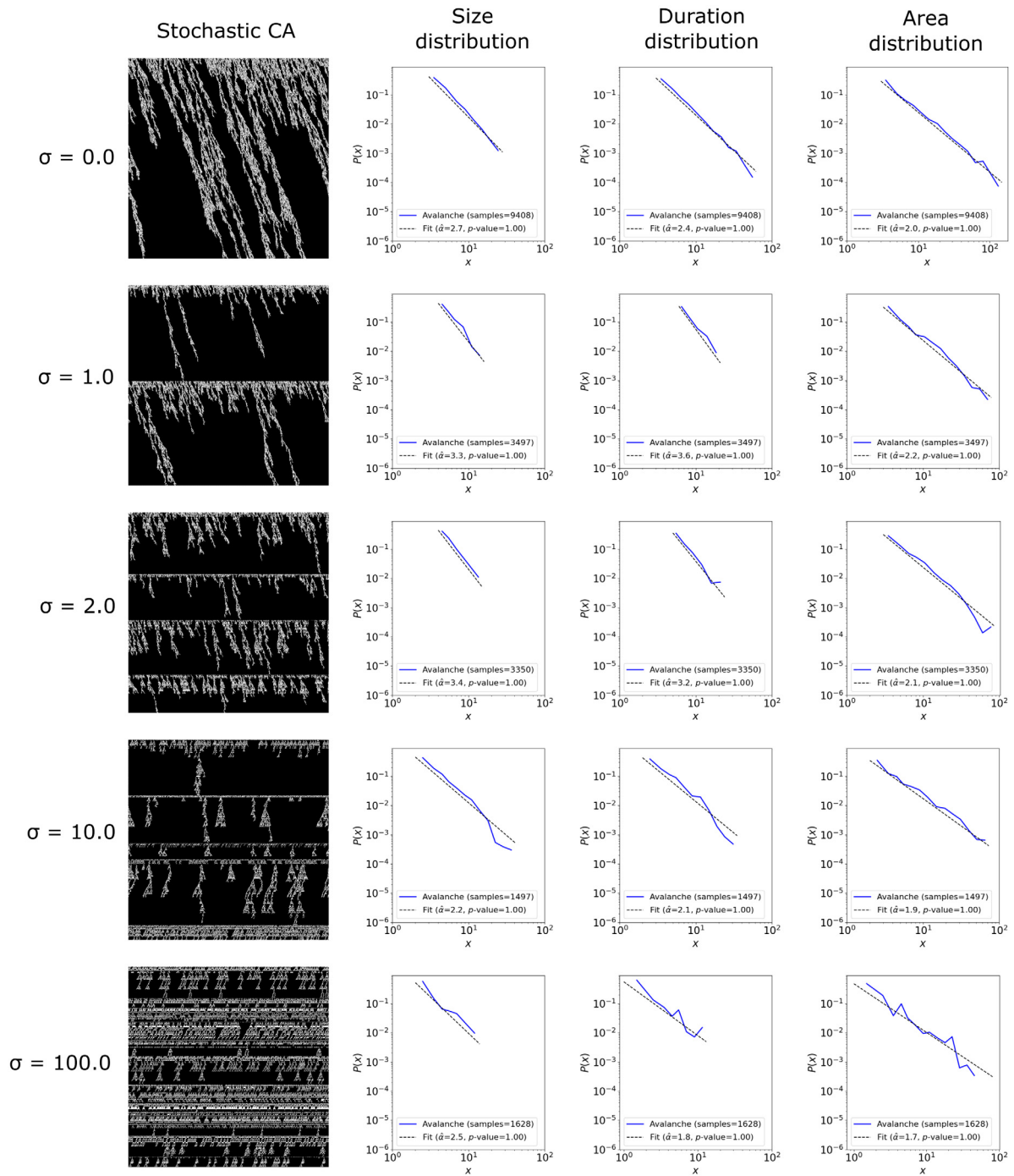
Fig. 2. Stochastic CAs and their avalanche distributions of the cells with state 1 (black cells) while affected by Gaussian noise. Here the CAs are produced with $N = 1000$ lattice squares and $T = 1000$ time steps. Large avalanches that happen only one time are ignored in these distributions. The dashed lines indicate the power-law fit, with a slope defined as $\hat{\alpha}$ and a goodness-of-fit given by the p-value (see text).

transition probabilities shown in Table 1. This choice of proba- bilities *maximizes* the critical behavior of the CA in such a con- figuration of boundary condition and initialization. There is a dependency in this configuration because if the CA starts with too many cells in the strong quiescent state (state 1), the rest- ing state can be reached faster, then requiring re-initialization because all the cells have the same state and become static. With the re-initialization, the CA will maintain its activity and more avalanches can be produced. Also, some boundary conditions can make the CA never reach the resting state. The criticality of the CA will be assessed by measuring the statistical distribution of features characterizing an avalanche in the CA evolution. We

define an avalanche during the evolution of a specific CA as the connected cluster of active states – i.e. state-1 cells – throughout the chain of cells composing the CA and throughout its time evolution. Fig. 1(b) illustrates an avalanche together with its three main features. The area of the avalanche corresponds to the total number of active states throughout the CA and during the full evolution which forms a connected set. The avalanche size is given by the total number of adjacent cells which are activated ($c = 1$) at least once, during the avalanche. The avalanche duration is given by the total number of successive iterations from the beginning until the end of the avalanche, including at least one site in the avalanche. The boundary of the CA is also

Table 1
Selected stochastic CA in [18].

State at t	$\mathcal{P}$ (Eq. (2))
(0,0,0)	0.394221
(0,0,1)	0.094721
(0,1,0)	0.239492
(0,1,1)	0.408455
(1,0,0)	0.000000
(1,0,1)	0.730203
(1,1,0)	0.915034
(1,1,1)	1.000000

the end of an avalanche. In the example of Fig. 1(b) the avalanche has a size of 7 cells, a duration of 9 iterations, and a total area of 25 sites. Our initial goal was to have a CA as wide and lasting as possible because we could get a broader range of avalanche size, duration and area. However, due to memory and time limitations, our selected number of lattice squares is $N = 1000$ and the number of time steps is $T = 1000$.

By keeping track of the avalanches emerging during the evolution of a stochastic CA, we can record the distribution of their area, size and duration. These three types of avalanche measurements are more similar to neuronal avalanches [7]. Moreover, the closer the distribution is from a power-law the closer the CA behavior is from a critical regime. Therefore, we recently introduced a fitness score (defined in Eqs. (4) and (5)), to assess how "critical" a particular CA realization is [18], based on different measures of the error associated with the fit. Using a genetic algorithm, we found the set of eight probabilities that leads to the most critical CA realizations (see Table 1).

Recently we implemented a Python library called *EvoDynamic*, to simulate several dynamical systems [18], such as cellular automata, random Boolean networks, and echo state networks. This library is based on the *TensorFlow* deep neural network framework [21], and it includes genetic algorithms that can be used to evolve dynamical systems towards a desired behavior or dynamics.

EvoDynamic has a general and single implementation that can simulate various dynamical systems because they are, in essence, dynamical graphs or networks. Since the weighted adjacency matrix and mapping function are specific to a dynamical system, they need to be adapted to simulate a stochastic elementary cellular automaton because the implementation of EvoDynamic is based on artificial neural networks. To make this possible, the implementation of an artificial neural network is generalized to incorporate the possibility of simulating CA. More specifically, EvoDynamic uses the general form of a feed-forward artificial neural network (ANN) without bias, $\mathbf{l}_{i+1} = a(\mathbf{W} \cdot \mathbf{l}_i)$, with $\mathbf{l}_i$, the layer index i of the ANN; $\mathbf{W}$, a weight matrix; and a, an activation function, and interprets it as a non-linear dynamical system with discrete time steps, namely:

$$\mathbf{c}_{t+1} = f(\mathbf{A} \cdot \mathbf{c}_t). \tag{3}$$

where $\mathbf{c}_t$ is a group of cells (or nodes in a graph) in iteration t, $\mathbf{A}$ is the weighted adjacency matrix, and f represents the (non-linear) mapping function. For the stochastic CA, the weighted adjacency matrix $\mathbf{A}$ connects the center cell $c_{i,t}$ with its three neighbors $(c_{i-1,t}, c_{i,t}, c_{i+1,t})$. To identify which neighbor has state 0 and 1 after the matrix multiplication between $\mathbf{A}$ and $\mathbf{c}_t$, the weights assigned to $(c_{i-1,t}, c_{i,t}, c_{i+1,t})$ are (4, 2, 1). Therefore, each of the eight neighborhood combinations is represented by a unique number from 0 to 7, which the mapping function f maps it to its corresponding probability for the random generation of state 0 or 1 in the next iteration.

The EvoDynamic framework was used to find the eight probabilities in Table 1 which maximizes the critical behavior of a

stochastic CA, applying a so-called genetic algorithm. The genetic algorithm, in general, mimics a "natural selection" process, searching for optimal parameter values (the probabilities) by changing the parameter values and maximizing a pre-defined fitness function that calculates a fitness score. Instead of maximizing the function through a "supervised" path, such as a gradient descendent scheme, the genetic algorithm updates the values of the probabilities, starting from some set of initial values, and then applying random perturbations. If the perturbation increases the fitness score, the set of probabilities increases the chance to be selected as a "parent" for the next generation of new sets of probabilities. In the end, the genetic algorithm retrieved the "genotype" of the stochastic CA with the best fitness score, composed of the eight probabilities p_i ($i = 0, \ldots, 7$) for each of the eight 3-tuple $(c_{i-1,t}, c_{i,t}, c_{i+1,t})$.

The fitness function is based on the fitness measures of the avalanche size and duration with a power-law function and is heuristically defined as [18]

$$S_{temp} = (R^2)^2 + D^2 + B + U, \tag{4}$$

$$S = \begin{cases} S_{temp} + L, & S_{temp} > 3.5 \\ S_{temp}, & \text{otherwise.} \end{cases} \tag{5}$$

where R^2 is the coefficient of determination of complete linear fitting [22], D is the normalized coefficient of the Kolmogorov–Smirnov (KS) statistic [23], B is the percentage of non-zero bins with size one in the avalanche histograms, U is the percentage of unique states through time, and L is the normalized log-likelihood ratio of the comparison between the power-law model and the exponential model for estimating the avalanche distributions [23]. These fitness function objectives are normalized to the range [0, 1] if necessary and the genetic algorithm is applied to indirectly maximize them through the fitness function. The squared values in Eq. (4), R^2 and D, are the most important ones for the fitness function and were empirically chosen. In Eq. (5), the adjusted log-likelihood ratio L is only calculated if $S_{temp} > 3.5$ because this is a computationally intensive process; and if L is not trustworthy (p-value of the ratio is greater or equal to 0.1), then this measurement is ignored (set as zero). In the end, we obtain the fitness score S.

2.2. Adapted model with stochastic transition rates

To test the robustness of the stochastic CA for remaining in criticality, Gaussian noise with a varying standard deviation σ is applied to affect the probabilities p_i of the CA for every time step. Since the Gaussian noise will make the perturbed probabilities pass the valid range between zero and one, normalization to the Gaussian mean (original probability) is used. Thus, the equation for the normalization is

$$\tilde{\mu}_i = \log\left(\frac{p_i}{1 - p_i}\right). \tag{6}$$

With the normalized mean $\tilde{\mu}_i$, we sample x_i from a Gaussian distribution, such as

$$x_i \sim \mathcal{N}(\tilde{\mu}_i, \sigma). \tag{7}$$

The random variable x_i is sampled each time step, preserving the definition of CA, i.e. at each time step all cells follow the same (probabilistic) updating rule. To make x_i a valid perturbed probability $\tilde{p}_i$, the sigmoid function is applied to it, then

$$\tilde{p}_i = \frac{1}{1 + e^{-x_i}}. \tag{8}$$

While there are other possible choices, this choice maps a Gaussian distributed variable into a sigmoid function between 0 and 1 which for the average of x_i, $\tilde{\mu}_i$ retrieves the initial value of p_i.
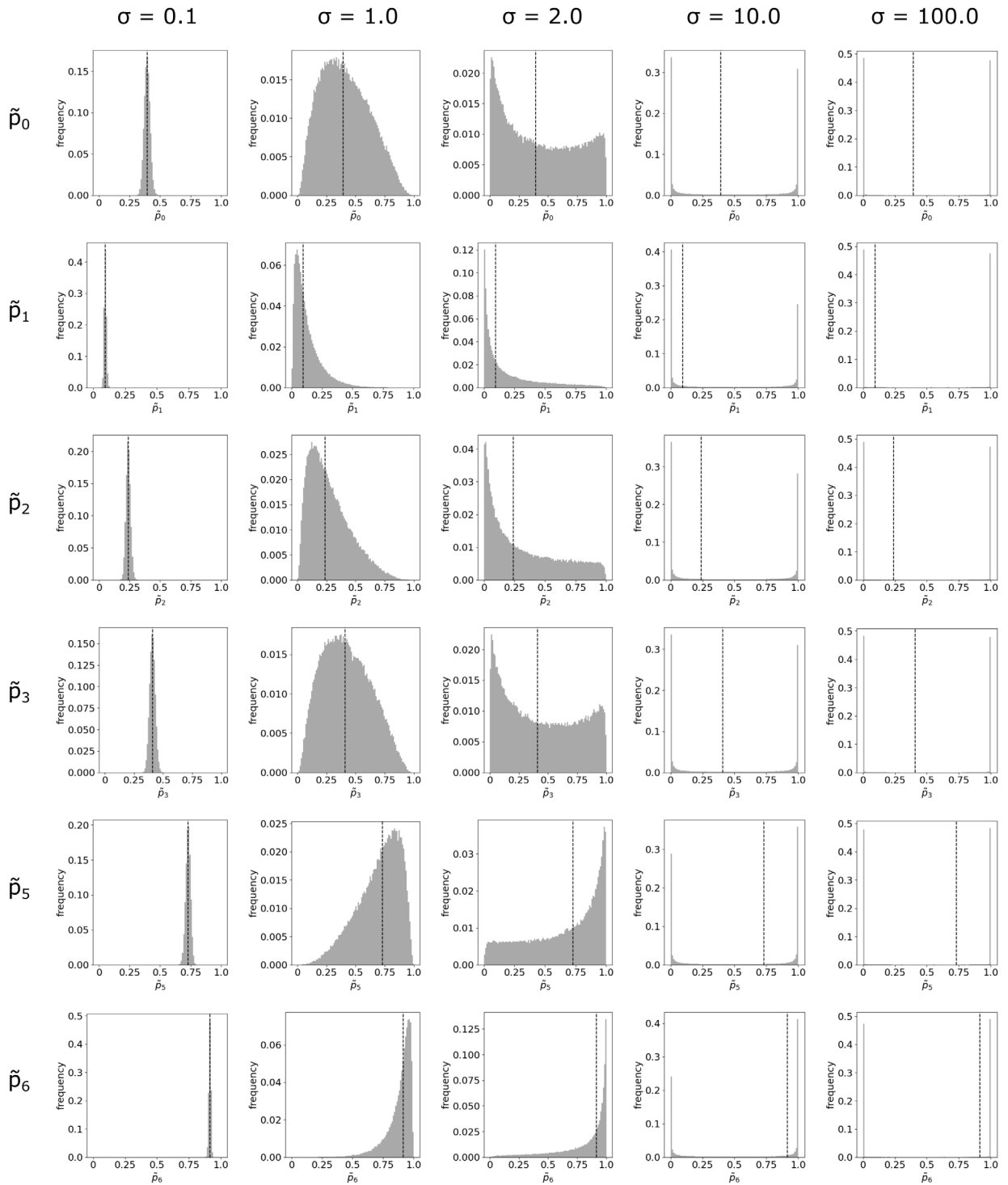
Fig. 3. Histogram of the valid perturbed probability $\tilde{p}_i$ for different values of σ. The vertical dashed line indicates the unperturbed probability ($\sigma = 0$). The probabilities $\tilde{p}_4$ and $\tilde{p}_7$ are not shown because they are always 0 and 1, respectively, independent of the value of σ. Each histogram was generated with 100,000 samples.

Having settled this, our research question can be reformulated as follows: for $\sigma = 0$ we have exactly the optimal solution, i.e. a power-law of the size of clusters, their duration, and their area, then, by increasing $\sigma > 0$ one will eventually destroy the power-laws observed for the optimal solution, pushing the system away from criticality. So, is there a transition from critical to non-critical stage tuning the σ? We note that the maximum value of σ for which criticality is still observed can be thought of as a measure of the robustness of the optimal (critical) stage.

3. Results

For producing the perturbed stochastic CAs and the distribution of their area, size and duration; we repeat the simulations for the values of the standard deviation σ from 0.1 to 2.0 in step size of 0.1, and also for values of 5, 10, 20, 50, 100, 200, 500, and 1000. For each of those values of σ, we perform 1000 CA simulations with a uniform random initialization. This is presented for some values of σ in Fig. 2. Its first row shows an illustrative realization
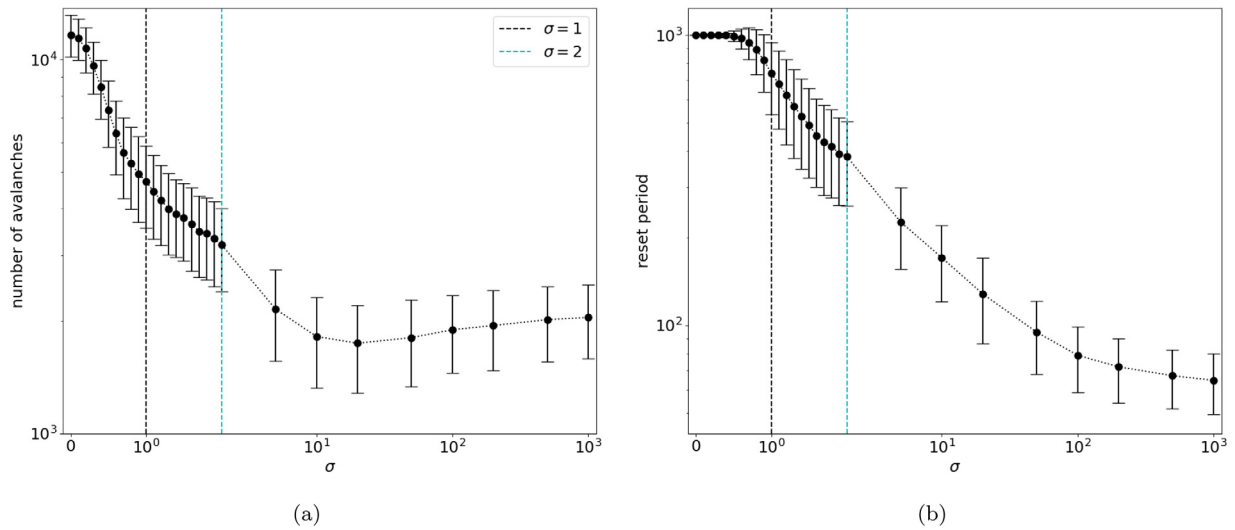
Fig. 4. Mean and standard deviation of (a) the number of avalanches and (b) reset period.

of the optimal stochastic CA with the probabilities in Table 1, showing the distribution of avalanche size, duration and area. A distribution is plotted as the avalanche measurement x with its occurrence probability $P(x)$. The power-law fit is indicated with dashed lines and we can see that a power-law distribution fits well the empirical histogram. The power-law fit is also indicated by its estimated slope $\hat{\alpha}$ and goodness-of-fit given by the p-value. This p-value is introduced by Clauset et al. [23], stating that a p-value greater than 0.1 indicates a valid power-law fit.

Fig. 3 presents the histograms of the valid perturbed probability $\tilde{p}_i$ calculated through Eq. (8) for some values of σ. When $\sigma = 2.0$, it is noticeable that the original Gaussian distribution becomes distorted. By increasing σ even more, the distribution of the valid perturbed probability $\tilde{p}_i$ becomes concentrated in 0 and 1.

Notice that, in case the strong quiescent state fully occupies the CA, its cells' states are re-initialized. For larger values of σ, since the CA reaches the inactivity state quicker than the optimal and unperturbed CA, the re-initialization happens more frequently. In Fig. 4, we plot the number of avalanches as well as the number of iterations before a re-initialization caused by all cells being in the quiescent state. As one sees in this figure, due to a more frequent re-initialization the average avalanche size, duration and area is reduced. In Fig. 2, this is not noticeable in the avalanche distributions because the large avalanches that happen just one time during the simulations are ignored. In those samples of the simulations, all goodness-of-fit p-value remains 1.0 as for the unperturbed CA. The standard deviation σ of the Gaussian noise to the probabilities strongly reduces the number of avalanches, especially from σ between 0.3 and 10.0. After that, the number of avalanches stabilizes around 2,000. This is because the avalanches became so short that they reach the resting state with all cells in the strong quiescent state much faster, then re-initializing the CA more often and having a similar number of avalanches. Such a reduction in the number of avalanches of state 1 is also due to the decrease in the occurrence of state 0 because the avalanches of state 1 need to be surrounded by cells with state 0 or boundary. We can perceive that the state 0 patterns became not only shorter but also thinner. To confirm this behavior and to have a sense of what happens at individual cell scale, Fig. 5 shows the occurrence rates and their standard deviation of state 0 and 1, and the transitions between these two states in both directions, from 0 to 1 ($0 \rightarrow 1$) and from 1 to 0 ($1 \rightarrow 0$). The occurrence rate of state 1 tends to increase until $\sigma = 2.0$
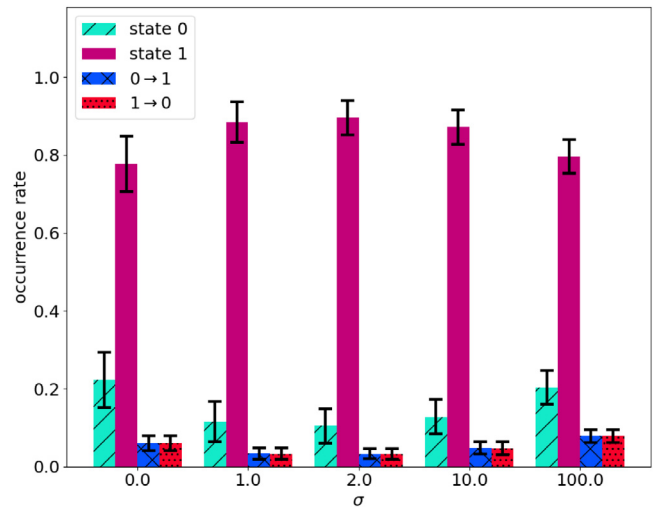

Fig. 5. Occurrence rates in the cells and their standard deviations of state 0 and 1, and the transitions between them that are from 0 to 1 ($0 \rightarrow 1$) and from 1 to 0 ($1 \rightarrow 0$). The statistics are calculated for each cell in 1000 CA simulations with $N = 1000$ and $T = 1000$.

because it is the strong quiescent state. The occurrence rate of state 0 and the transitions are inversely or directly proportional to the rate of state 1 because they are interdependent. However, for $\sigma > 2.0$, the re-initialization of the CA counterbalances the trend of changes in those rates, even though it does not occur with the number of avalanches.

For the systematic evaluation of the robustness, we analyze the distributions of duration, size and area of the avalanches, by fitting to them a power-law and estimate their slope $\hat{\alpha}$, as well as their uncertainty through a Kolmogorov-Smirnov test and the corresponding Kullback-Leibler (KL) divergence [24,25].

The Kullback–Leibler divergence is defined as:

$$D_{KL} = \sum_x P(x) log \left(\frac{P(x)}{Q(x)} \right) \tag{9}$$

where $P(x)$ is the probability distribution of the empirical data and $Q(x)$ is the probability distribution function of the estimated
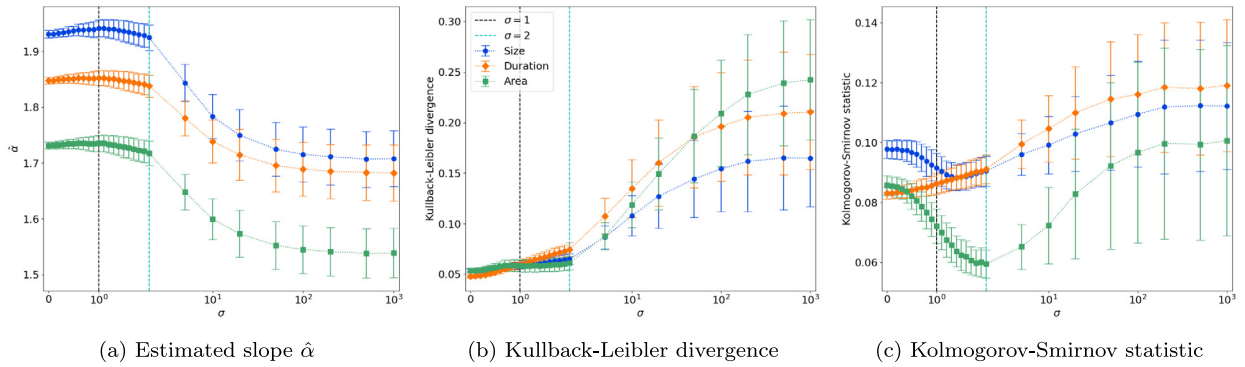
(a) Estimated slope $\hat{\alpha}$

(b) Kullback-Leibler divergence

(c) Kolmogorov-Smirnov statistic

Fig. 6. Mean and standard deviation of the measurements of the power-law estimation for the avalanche distributions of state 1.

(a) Kullback-Leibler divergence
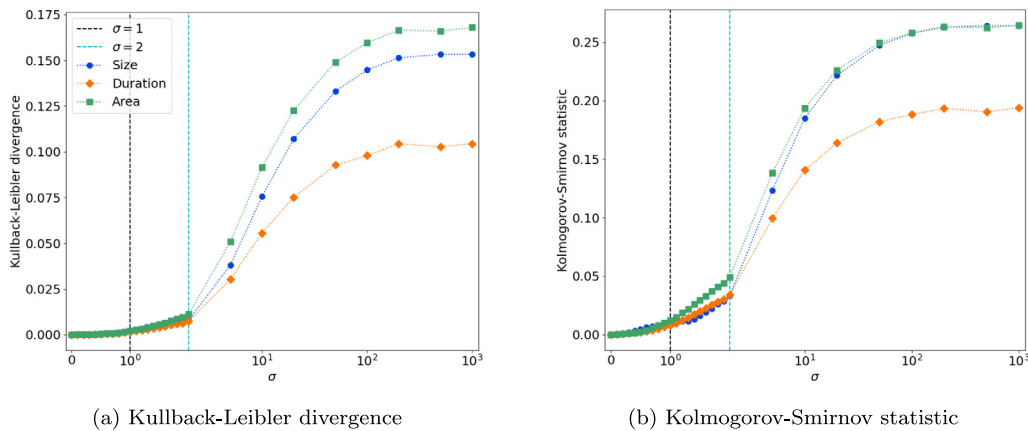
(b) Kolmogorov-Smirnov statistic

Fig. 7. Comparison between the empirical distributions of each σ and the empirical distribution with $\sigma = 0$ used as a baseline.

power-law, and the supremum measuring the KS test is given by

$$D_{KS} = \sup_x |P_C(x) - Q_C(x)| \tag{10}$$

where $P_C(x)$ is the cumulative distribution of $P(x)$ and $Q_C(x)$ is the cumulative distribution function of $Q(x)$.

Results are shown in Fig. 6: the slope is plotted in Fig. 6(a), while the KL divergence is plotted in Fig. 6(b) and the KS test (supremum) statistic is shown in Fig. 6(c). Clearly, the slopes are almost constant for all distributions – size, duration and area – till a value of $\sigma \gtrsim 1$. Above $\sigma = 2$, the estimated slopes reduce drastically, indicating the deviation from critical behavior.

The KL divergence D_{KL} presents a constant (small) value for $\sigma \lesssim 1$, indicating a constant small error associated with the power-law fit. Beyond $\sigma = 2$ this error increases considerably. The KS statistic D_{KS} shows a more fluctuating behavior. Indeed, counter-intuitively, before reaching $\sigma = 1$, the supremum decreases to values smaller than the ones observed for $\sigma < 1$. This may be due to the occurrence of many small avalanches and one single large avalanche.

A comparison between the avalanche distributions of each σ and the avalanche distributions of the stochastic CA with $\sigma = 0$ is shown in Fig. 7. Therefore, avalanche distributions with $\sigma = 0$ are used as a baseline for KL divergence (Fig. 7(a)) and KS statistic (Fig. 7(b)). This comparison indicates how the dynamics changes when additional noise is applied, and it confirms our observations from the KL divergences with their power-law fits.

4. Discussion

The optimal stochastic CA without perturbations ($\sigma = 0$) and probabilities shown in Table 1 can maintain its critical behavior

even when perturbed ($\sigma > 0$), but only up to a certain point. By analyzing Figs. 6 and 7, we can conclude that $\sigma = 1$ is the breaking point between behaving and not behaving similarly to the unperturbed one. This can be noticed especially in Fig. 6(a). The estimated slopes $\hat{\alpha}$ for size, duration and area remain almost unvaried until $\sigma = 1$, while their standard deviations slowly increase as a result of the fluctuations in the values of the probabilities. Fig. 4(a) shows that the number of avalanches starts to decrease from approximately 12,000 to around 4000 in the range $\sigma = [0, 1]$, even though the behavior is still maintained with respect to the estimated slopes $\hat{\alpha}$, KS statistic, and KL divergence. This cannot be seen in the reset period (Fig. 4(b)) because the number of time steps simulated is 1000, making it the maximum reset period possible.

Neuronal stochastic variability [26] happens all over the brain and on all scales. Since the optimal stochastic CA investigated in this work presented robustness up to perturbations with $\sigma = 1$, biological neural networks may also allow for similar robustness to perturbations. Therefore, both systems may preserve criticality even in noisy conditions. Because self-organized criticality is possibly one of the main factors for the emergence of intelligence in the human brain [7], the evaluation of robustness to stochastic variability can be an important measurement for indicating the presence of intelligence in artificial systems.

In future work, we plan to evaluate the performance of the optimal stochastic CA under perturbations in a machine learning framework called reservoir computing [27–29]. Reservoir computing is a biologically-plausible neural model inspired by the functioning of cortical microcircuits [30]. There is indeed evidence that reservoir computing achieves better performances when it produces critical dynamics [31–33]. This benchmark

would inform how the artificial intelligence system can maintain its accuracy while increasing the probability noise. Therefore, such investigation may confirm that this robustness is also essential for future artificial intelligence systems, and in particular inform the realization of novel brain-inspired neuromorphic hardware.

CRediT authorship contribution statement

Sidney Pontes-Filho: Conceptualization, Methodology, Software, Visualization, Validation, Formal analysis, Data curation, Writing – original draft. **Pedro G. Lind:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Stefano Nichele:** Supervision, Resources, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

[1] K. Christensen, Complexity and Criticality, Imperial College Press, 2005.

[2] P. Bak, C. Tang, K. Wiesenfeld, Self-organized criticality: An explanation of the 1/f noise, Phys. Rev. Lett. 59 (4) (1987) 381.

[3] P. Bak, C. Tang, K. Wiesenfeld, Self-organized criticality, Phys. Rev. A 38 (1) (1988) 364.

[4] B.B. Mandelbrodt, Fractals and Scaling in Finance, Springer-Verlag New York, 1997.

[5] J.P. da Cruz, P.G. Lind, The bounds of heavy-tailed return distributions in evolving complex networks, Phys. Lett. A 377 (2013) 189–194.

[6] J.P. da Cruz, P.G. Lind, The dynamics of financial stability in complex networks, Eur. Phys. J. B 85 (2012) 256.

[7] K. Heiney, O. Huse Ramstad, V. Fiskum, N. Christiansen, A. Sandvig, S. Nichele, I. Sandvig, Criticality, connectivity, and neural disorder: a multifaceted approach to neural computation, Front. Comput. Neurosci. 15 (2021) 7.

[8] A.J. Fontenele, N.A.-P. de Vasconcelos, T. Feliciano, L.A.-A. Aguiar, C. Soares-Cunha, B. Coimbra, L.D. Porta, S. Ribeiro, A.J. Rodrigues, N. Sousa, P.V. Carelli, M. Copelli, Criticality between cortical states, Phys. Rev. Lett. 122 (2019) 208101.

[9] L. Cocchi, L.L. Gollo, A. Zalesky, M. Breakspear, Criticality in the brain: A synthesis of neurobiology, models and cognition, Prog. Neurobiol. 158 (2017) 132–152.

[10] S. Wolfram, A New Kind of Science, Wolfram Media, 2002.

[11] G. Vichniac, Simulating physics with cellular automata, Physica D 10 (1984) 96–116.

[12] H. Chaté, P. Manneville, Criticality in cellular automata, Physica D 45 (1–3) (1990) 122–135.

[13] J. Singha, N. Gupte, Chimera states in coupled map lattices: Spatiotemporally intermittent behavior and an equivalent cellular automaton, Chaos 30 (11) (2020) 113102.

[14] P. Bak, K. Chen, M. Creutz, Self-organized criticality in the'Game of Life, Nature 342 (6251) (1989) 780–782.

[15] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for Your Mathematical Plays, Vol. 4, AK Peters/CRC Press, 2004.

[16] I. Rahwan, M. Cebrian, N. Obradovich, J. Bongard, J.-F. Bonnefon, C. Breazeal, J.W. Crandall, N.A. Christakis, I.D. Couzin, M.O. Jackson, et al., Machine behaviour, Nature 568 (7753) (2019) 477–486.

[17] C. Gershenson, V. Trianni, J. Werfel, H. Sayama, Self-organization and artificial life, Artif. Life 26 (3) (2020) 391–408.

[18] S. Pontes-Filho, P. Lind, A. Yazidi, J. Zhang, H. Hammer, G.B. Mello, I. Sandvig, G. Tufte, S. Nichele, A neuro-inspired general framework for the evolution of stochastic dynamical systems: Cellular automata, random boolean networks and echo state networks towards criticality, Cogn. Neurodyn. 14 (5) (2020) 657–674.

[19] C.G. Langton, Computation at the edge of chaos: Phase transitions and emergent computation, Physica D 42 (1–3) (1990) 12–37.

[20] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.

[21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, TensorFlow: A system for large-scale machine learning, in: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association, Savannah, GA, 2016, pp. 265–283, URL: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

[22] S. Wright, Correlation and causation, J. Agric. Res. 20 (1921) 557–580.

[23] A. Clauset, C.R. Shalizi, M.E. Newman, Power-law distributions in empirical data, SIAM Rev. 51 (4) (2009) 661–703.

[24] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.

[25] D.J. MacKay, D.J. Mac Kay, et al., Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.

[26] M.D. McDonnell, J.H. Goldwyn, B. Lindner, Neuronal stochastic variability: influences on spiking dynamics and network activity, Front. Comput. Neurosci. 10 (2016) 38.

[27] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comp. Sci. Rev. 3 (3) (2009) 127–149.

[28] O. Yilmaz, Reservoir computing using cellular automata, 2014, arXiv preprint arXiv:1410.0162.

[29] S. Nichele, A. Molund, Deep reservoir computing using cellular automata, 2017, arXiv preprint arXiv:1703.02806.

[30] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Comput. 14 (11) (2002) 2531–2560.

[31] L.E. Suárez, B.A. Richards, G. Lajoie, B. Misic, Learning function from structure in neuromorphic networks, Nat. Mach. Intell. 3 (9) (2021) 771–786.

[32] T.E. Glover, P. Lind, A. Yazidi, E. Osipov, S. Nichele, The dynamical landscape of reservoir computing with elementary cellular automata, in: ALIFE 2021: The 2021 Conference on Artificial Life, MIT Press, 2021.

[33] J. Boedecker, O. Obst, J.T. Lizier, N.M. Mayer, M. Asada, Information processing in echo state networks at the edge of chaos, Theory Biosci. 131 (3) (2012) 205–213.