



# Trajectories Entropy in Dynamical Graphs with Memory

Francesco Caravelli<sup>1,2,3\*</sup>

<sup>1</sup> Invenia Labs, Cambridge, UK, <sup>2</sup> London Institute of Mathematical Sciences, London, UK, <sup>3</sup> Department of Computer Science, University College London, London, UK

In this paper, we investigate the application of non-local graph entropy in evolving and dynamical graphs. The measure is based upon the notion of Markov diffusion on a graph and relies on the entropy applied to trajectories originating at a specific node. In particular, we study the model of reinforcement–decay graph dynamics, which leads to scale-free graphs. We find that the node entropy characterizes the structure of the network in the two parameter phase–space describing the dynamical evolution of the weighted graph. We then apply an adapted version of the entropy measure to purely memristive circuits. We provide evidence that meanwhile in the case of DC voltage, the entropy based on the forward probability is enough to characterize the graph properties; in the case of AC voltage generators, one needs to consider both forward- and backward-based transition probabilities. We provide also evidence that the entropy highlights the self-organizing properties of memristive circuits, which re-organizes itself to satisfy the symmetries of the underlying graph.<sup>1</sup>

## OPEN ACCESS

### Edited by:

Mikhail Prokopenko,  
University of Sydney, Australia

### Reviewed by:

Juan Pablo Carbajal,  
Eawag, Switzerland  
Dominique Chu,  
University of Kent, UK

### \*Correspondence:

Francesco Caravelli  
francesco.caravelli@invenialabs.co.uk

### Specialty section:

This article was submitted to  
Computational Intelligence,  
a section of the journal  
Frontiers in Robotics and AI

**Received:** 28 November 2015

**Accepted:** 29 March 2016

**Published:** 20 April 2016

### Citation:

Caravelli F (2016) Trajectories Entropy  
in Dynamical Graphs with Memory.  
*Front. Robot. AI* 3:18.  
doi: 10.3389/frobt.2016.00018

**Keywords:** graphs with memory, graph entropy, memristors, symmetry

## 1. INTRODUCTION

The theory of complex networks has found applications in many fields across the natural sciences. Until recently, most of the studies performed in complex networks concerned static graphs. Preferential attachment (Albert and Barabasi, 2002) is the most well-known mechanism for constructing scale-free networks, and it concerns in fact with a specific type of graphs' growth in order to explain the properties of realistic, well-known networks (rich-gets-richer phenomenon). Scale-free networks are defined by the distribution  $P(k)$  of the degrees of connectivity, which obeys a power law  $P(k \gg 1) \approx k^{-\rho}$ , with  $k$  being the number of connections of a given node. The exponent  $\rho$  is typically in the range between 2 and 3; moreover, many realistic networks turn out being also small world (Watts and Strogatz, 1998; Albert and Barabasi, 2002; Caldarelli, 2007; Barrat et al., 2009; Newman, 2010; Estrada, 2011).

Several models, in which preferential attachment is an emergent property, have been proposed in the physics literature (Kleinberg, 1998; Dorogovtsev et al., 2002; Saramaeki and Kaski, 2004; Evans and Saramaeki, 2005; Ikeda, 2008; Chrysafis and Cannings, 2009; Li et al., 2010), and inspired both by ants and memristors, we proposed a model of dynamical graphs in which two competing mechanisms take place: a graph "evaporation" (or decay) and a reinforcement process due to the diffusion of particles on the graph. In this model, memory is represented by non-Markovianity, as the reinforcement process changes the particles' hopping probabilities.

<sup>1</sup> Part of this work was presented at the 7th workshop on Guided Self-Organization, Freiburg, Germany (2014) and at the Symposium on Complexity, Computation and Criticality in Sydney (2015).

Recently, the interest has shifted toward the understanding of dynamical phenomena of graphs. In this article, we present new results on a class of dynamical models that gives rise to scale-free graphs by means of what we call *memory*. Memory is an ubiquitous and necessary requirement to give rise to complex phenomena in plenty of contexts. The main result obtained in Caravelli et al. (2015) is that the interplay between random growth of the graph, decay of the links, and their strengthening performed by random walkers hopping over them leads to the generation of scale-free graphs. Thus, although the model is local (in the sense of evolution rules which are local on the graph), it is a non-local one from the temporal point of view. Interestingly, real condensed-matter systems show some degree of memory in their response functions (e.g., its resistance) when subject to external perturbations (Di Ventra and Pershin, 2013a). Similarly, a memory mechanism is used as an optimization procedure by ants in order to find the shortest path, by reinforcing with pheromones the most walked paths (Schweitzer et al., 1997; Buhl et al., 2004) and has been shown to be employed by networks of memristors (resistors with memory) (Chua, 1971; Strukov et al., 2008) to solve optimization problems such as the maze (Pershin and Di Ventra, 2011) or other shortest path problems (Pershin and Di Ventra, 2013). Recently, it has been argued that several “hard” problems can be solved in polynomial time using memristors (Traversa et al., 2015).

In the case of dynamical graphs, it is hard (if not impossible) to characterize with only a few parameters the topological properties of the dynamics, in particular if the rules of evolution are non-local, as for instance in the case of memristors. Following this line of thought, we tentatively study the properties of the resulting graph using a measure of graph entropy introduced in Caravelli (2014). This measure of graph entropy is based on the idea that for the case of graphs, in order to give a non-local characterization of a node based on the macroscopic connectivity properties, one needs to consider higher order loops in the network. In the present paper, we provide evidence that for the specific case of two specific dynamical graph models, such as the reinforcement–decay and memristive circuit models, the evolution properties can be to some extent characterized by a non-local graph entropy.

The use of non-local information theoretic measures is inspired by the macroscopic notions of entropy introduced in Lindgren (1988) and Lloyd and Pagels (1988) and applied to the Markov chain transition probabilities based on dynamical graphs. If  $M$  denotes the Markov transition matrix, one can define the (local) entropy of a state  $i$  as:

$$S(i) = - \sum_j M_{ij} \log(M_{ij}), \quad (1)$$

with  $M_{ij}$  being the Markov operator. The transition matrix  $M$  can be derived from a graph  $\Gamma$ , given an element-wise non-negative adjacency matrix  $A$ , by normalizing over rows or columns. From the point of view of graph theory, equation (1) is completely local. In order to account for non-local effects, such entropy necessitates an extension to account non-local (global) effects, such as loops. Similar ideas were also considered in the information theory community (eKroot) and in the complex networks community (Braunstein et al., 2006; Anand and Bianconi, 2009). In the present paper, we consider the extension considered in Caravelli (2014),

which relies on the definition of Markov probabilities not on the local diffusion probability at a node but on the probability of trajectories *originated* at a specific node.

The paper is organized as follows. In Section 2.1, we provide the algorithm for the toy model of reinforcement–decay introduced in a previous work, and give further arguments for its robustness. In Section 2.2, we introduce linear memristors and their dynamical behavior. In Section 2.3, we recall the graph entropy measure later used to analyze the properties of dynamical graphs. We then apply the graph entropy in Sections 3.1 and 3.2 for the reinforcement–decay model and memristive circuits, respectively. Conclusions follow.

## 2. MATERIALS AND METHODS

### 2.1. Reinforcement–Decay Dynamical Graphs

In this section, we recall the graph evolution model introduced in Caravelli et al. (2015) in order to study a simplified dynamics mimicking systems of ants (reinforcing walkers) and as a toy model for studying graphs with memory.

*Model* – The algorithm to dynamically modify the graph is based upon the following steps:

- (a) *Initialization*: we start with a weighted random graph of  $N_0$  nodes. The weights are drawn with constant probability in  $[0,1]$ , and  $P \leq N_0$  particles are randomly placed.

After initialization, a cycle of the algorithm consists of the steps of Hopping, Strengthening/Decay, and Growth:

- (b) *Hopping*: we let the particles hop between nodes  $i$  and  $j$  with probability  $p_{ij}$  proportional to the link strength  $p_{ij} = A_{ij}/\sum_j A_{ij}$ , where  $A_{ij}$  is the weighted adjacency matrix of the graph.
- (c) *Strengthening/Decay*: all the links hopped on by the particles in the last  $L$  steps are reinforced by  $\gamma$ . Links with strength less than threshold  $L_d$  decrease their strength by  $\alpha$ , with probability  $p_d$ , and are removed when they reach a negative weight.

A critical ingredient to obtain scale-free graphs was shown to be:

- (d) *Growth*: at this step, a new node is added (and with probability  $p_p$  a new particle is placed on it). The new node connects to each of the previously existing nodes with probability  $p_{nb}$ , here chosen to be one, and with random weights with constant probability in  $[0,1]$ .

The memory feature of this model lays in the non-Markovianity of the particles hopping. Each time, a particle hops on an existing link this is reinforced, and thus the probability of hopping on it afterward increases. We note that the decay process competes with the reinforcing one. The interplay between these two phenomena has been shown to lead to a critical growth of the effective graph, obtaining asymptotic degree distributions, which are scale free. The observation that the growth of the graph is a crucial step for scale-free degree distribution is made by imposing the growth to stop at a certain maximum number of nodes  $N_c$ . After the growth stops, it is observed that graph properties change abruptly, and

the scale-free degree distribution is lost. The reinforcement–decay graph evolution is a rich-gets-richer mechanism and was inspired by traveling ants leaving an evaporating track of pheromones; also memristors have similar dynamical properties. As described in the Appendix of Caravelli et al. (2015), the observation of the scale-free degree distribution law can be made precise in the statistical sense and derived analytically. Here, we provide the background, giving further arguments of why the algorithm is robust.

It is easy to argue that the degree  $k_s$  of the node  $s$  has an effective evolution equation of the form:

$$k_s(t + 1) = k_s(t) - \alpha k_s(t) + p_s \sum_{i=1}^N w_{si}, \quad (2)$$

where  $w_{si}$  are independent random variables drawn from a constant probability distribution  $P(x) = 1$ , and  $p_s = 2\rho/N$  to match the evolution obtained in Caravelli et al. (2015). We note that for  $N \gg 1$ , due to the central limit theorem, we have that  $\sum_{i=1}^N w_{si} \approx \mathcal{N}(\frac{N}{2}, \frac{\sqrt{N}}{\sqrt{12}})$ , where  $1/12$  is the variance of the distribution, and  $\frac{N}{2}$  is its mean. This implies that in the large  $N$  limit, we can describe the evolution of the degree of the graph as a stochastic differential equation of the form:

$$dk_s(t) = \left( p_s \frac{N}{2} - \alpha k_s(t) \right) dt + p_s \tilde{\sigma} dW_t, \quad (3)$$

where  $dW_t$  is an *effective* Wiener differential, and where we defined  $\tilde{\sigma} = \frac{\sqrt{N}}{\sqrt{12}}$ , and introduced  $dk_s(t) = k_s(t + 1) - k_s(t)$ . Using standard formulae for the mean and the variance, we can thus write the compact effective equation, and replacing the right scaling  $p_s \rightarrow 2\rho/N$  to account for the linear growth of the degree:

$$\begin{aligned} dk_s(t) &= (\rho - \alpha k_s(t)) dt + \frac{2\rho}{N} \tilde{\sigma} dW_t \\ &= (\rho - \alpha k_s(t)) dt + \frac{\rho}{\sqrt{3N}} dW_t \end{aligned} \quad (4)$$

The stochastic differential equation of equation (4) of the same form as the one obtained in Caravelli et al. (2015), but with the addition of a stochastic term. We note that equation (4) is a stochastic differential equation of the form:

$$dk = (ak + c)dt + (bk + d)dW_t, \quad (5)$$

for the function  $k(t)$ , where the constants are given by

$$a = -\alpha \quad (6)$$

$$b = 0 \quad (7)$$

$$c = \rho \quad (8)$$

$$d = \frac{\rho}{\sqrt{3N}}. \quad (9)$$

If we define the function  $\Phi_t$  given by,

$$\Phi_t(t, W_t) \equiv \exp\left((a - b^2/2)t + bW_t\right) = \exp(-\alpha t), \quad (10)$$

the solution of the differential equation of equation (4) is given by Kloeden and Platen (1999):

$$\begin{aligned} k_s(t) &= \Phi_t \left( k_s^0 + (c - bd) \int_0^t \Phi_s^{-1} ds + d \int_0^t \Phi_s^{-1} dW_s \right) \\ &= e^{-\alpha t} \left( k_s^0 + \rho \int_0^t e^{\alpha s} ds + \frac{\rho}{\sqrt{3N}} \int_0^t e^{\alpha s} dW_s \right) \\ &= e^{-\alpha t} \left( k_s^0 + \rho \frac{e^{\alpha t} - 1}{\alpha} + \frac{\rho}{\sqrt{3N}} \int_0^t e^{\alpha s} dW_s \right) \end{aligned} \quad (11)$$

where  $k_s^0$  represents the initial condition. Using the fact that  $\langle \int^t I(s) dW_s \rangle_{W_t} = 0$  for all smooth and deterministic processes  $I(s)$ , we obtain the solution as a function of  $t$ :

$$\langle k_s(t) \rangle_{W_t} = e^{-\alpha t} \left( k_s^0 + \rho \frac{e^{\alpha t} - 1}{\alpha} \right). \quad (12)$$

The constant  $k_s^0$  can be related to the node  $s$  using the boundary conditions at a certain initial time  $t_0$ . It is easy to evaluate the variance of this process at time  $t$ , being given by the stochastic term. Using the Itô isometry formula, we note that  $\text{Var}[k_s(t)] = \frac{\rho^2}{3N} \int_0^t e^{2\alpha s} ds = \frac{\rho^2}{3N} \frac{(e^{2\alpha t} - 1)}{2\alpha}$ . This solution is the same as the one obtained in Caravelli et al. (2015), which was confirmed by means of numerical simulations and exhibiting very robust scale-free distributions under parameter perturbations. The robustness can be explained observing that the variance of the process scales as  $1/N$ . This implies that the larger the graph, the smaller the deviation from the asymptotic distribution.

## 2.2. Memristive Circuits

A memristor can be thought simply as a dynamical resistance, which depends on an internal state variable. The internal variable, the “memory,” satisfies a dynamical law, which depends either on the applied voltage or on the current. Memristors are passive components, which perform analog computation: computing in memory is in fact one of the newly formed paradigms in which novel non-Turing computational scenarios involve memory elements, which store and process information in the same physical location (Di Ventra and Pershin, 2013b). Although this new form of computation could, in principle, be performed using CMOS technology, the most promising and energy-effective implementation relies upon the use of memristive, memcapacitive, and meminductive units (Di Ventra et al., 2009). These components can find a plethora of applications in electronics for building bio-inspired circuitry aimed at performing neuromorphic computation.

One of the interesting features of memristors is that these can be used both in analog or digital mode and in combination with CMOS components (Di Ventra and Pershin, 2013a). Several applications have been found for memristors, such as their use in memcapacitive neural networks, and the possibility of performing logical operations directly within memory using memelements. This latter feature is of particular interest to solve the long-standing von Neumann bottleneck problem of modern computers and solve NP-hard problems in polynomial *time* (Traversa et al., 2015). Being these passive component, the use of memelement for low energy dissipating circuits is a further appealing feature.

Memristive components are involved in the solution of optimization problems. It is thus of paramount importance to know how fast circuits made of memristor reach an asymptotic equilibrium configuration of the memristances. This implies that studying the emergence of the solution of an optimization problem can be thought as a relaxation phenomenon. Thus, this problem becomes a typical thermalization problem in non-equilibrium statistical physics. In this paper, we will try to infer global, non-local properties of the dynamics of memristive circuits using insights, which arose in the context of network theory.

A memristive component can be described by the following set of equations

$$V(t) = R(w, t)I(t), \quad (13)$$

$$\dot{w} = f(w, I), \quad (14)$$

where  $I(t)$  is the current in the memristor at time  $t$ ,  $V(t)$  is the applied voltage,  $R$  is memristance, which depends on the state of the system and can vary in time,  $w$  is a set of  $n$  state variables describing the internal state of the system, and  $f$  is a continuous  $n$ -dimensional vector function.

For the present paper, we consider the case of a linear memristor of the HP-type (Strukov et al., 2008), which can be described by the equations:

$$V(t) = R(w, t)I(t) \quad (15)$$

$$\dot{w} = p \frac{\mu R_{on}}{2d^2} I = \frac{p}{\beta} R_{on} I \quad (16)$$

$$\begin{aligned} R(w, t) &= R_{on}(1 - w(t)) + R_{off}w(t) \\ &= R_{on}[1 + (r - 1)w(t)], \end{aligned} \quad (17)$$

where  $p = \pm 1$  and represents the polarity of the memristor,  $R_{on}$  is the limiting resistance when the memristor is in the conducting phase, and  $R_{off}$  is the resistance in the insulating phase; we defined  $r = \frac{R_{off}}{R_{on}}$ , usually assumed to satisfy the relation  $r \gg 1$ .<sup>2</sup>

This is the simplest model proposed in the literature, but more realistic ones include state decay. In fact, realistic physical models include also an extra term proportional to the weight (Ohno et al., 2011; Avizienis et al., 2012; Carbajal et al., 2015). The simplest way to include such decay is to extend the dynamics for internal parameter dynamics equation with a linear term which produces an exponential decay. The extended equation takes the form:

$$\dot{w} = \frac{p}{\beta} R_{on} I + \alpha w. \quad (18)$$

where  $\alpha$  represents the decay parameter. Assuming for instance that  $I$  is constant asymptotically, dynamically one obtains a fixed point in the internal memory parameter,  $w(\infty) \propto I$ , which can be different from the boundary values  $w = 0, 1$ . If  $\alpha > 0$ , in the absence of external current the memristor decays to obtain the boundary resistance value  $R_{off}$ .

<sup>2</sup>We use here the convention where  $R_{on}$  corresponds to  $w = 0$  and  $R_{off}$  to  $w = 1$ . Thus the memristors considered here have switched polarity with respect to those studied by Strukov et al. (2008). This model is dynamically not equivalent to the one in Strukov et al. (2008) because of the switched polarity. However, the two models are related by the linear transformation  $w' = 1 - w$ , which preserves the position of the boundary values of the internal parameters.

A network of pure memristors satisfies the laws of standard, linear circuit theory. Given in fact a generic network of resistances, the circuit voltages and currents satisfy the constraints given by:

$$\vec{v} = \hat{R}(w(t))\vec{i} + \vec{s} \quad (19)$$

$$\hat{G}\vec{i} = 0 \quad (20)$$

where the first equation represents simply the relation between the voltage on each branch  $k$  of the network  $v_k$ , the current in that branch  $i_k$ , the eventual voltage sources  $s_k$ , and each resistance. The matrix  $\hat{G}$  simply implements the conservation of currents at each node, which can be thought as a constraint. For the case of memristors, the matrix  $\hat{R}$ , and each diagonal element contains the resistance of each memristor at time  $t$ ,  $R_{kk}(t) \equiv R_k(w_k(t))$ . If the memristors are of the HP type, then one has to introduce also an equation for the internal memory parameters  $\dot{w}_k(t) = f(w_k, I_k)$ , and impose that constraints on their domain, e.g.,  $0 \leq w_k(t) \leq 1$ .

In this paper, we study the specific network configuration, made of  $L$  layers and which can be defined recursively. Each layer  $k$  contains  $2^k$  memristors connected in triangles, representing thus a tree. The node at the top of the network is connected to one side of the voltage generator, and those at the bottom are connected to the other one and all connected to the ground. We consider this specific graph due to its many symmetries (i.e., intuitively we expect that nodes on each layer should have the same properties and current configurations for instance) and for the sake of simplicity, since its implementation can be easily made recursive.<sup>3</sup>

### 2.3. Graph Entropy

We recall the network entropy measure introduced in Caravelli (2014). The goal is to obtain a graph entropy, based on the Markov transition probability on a graph, which is able to characterize nodes individually.

Given a graph represented by an elementwise non-negative adjacency matrix  $A$ , the Markov forward (backward) transition probabilities can be derived using  $M_{ij} = \frac{A_{ij}}{\sum_j A_{ij}}$  ( $M_{ij} = \frac{A_{ji}}{\sum_j A_{ji}}$ ). As we will see shortly, the entropy is based upon the notion of probability applied to a particular trajectory in the set of states of a Markovian discrete dynamics and originating at a particular state. Given a the Markovian transition probabilities between two states (nodes of the graph), it is possible to derive the probability of certain trajectories  $\gamma$ , here denoted as  $\tilde{M}(\gamma)$ . We denote  $\{\gamma_i^k\}$  the set of trajectories of length  $k$  and originating at  $i$ . For instance, if the path is given by  $\gamma = \{i, j, k\}$ , the probability of that trajectory will be given by  $M(\gamma) = M_{ij}M_{jk}$ . Since  $\sum_{\{\gamma_i^k\}} \tilde{M}(\gamma_i^k) = \sum_{j_1, j_2, \dots, j_k} \tilde{M}(\{i, j_1, \dots, j_k\}) = 1 \forall k$ , it is easy to see that  $(\{\gamma_i^k\})$ ,  $\tilde{M}(\{\gamma_i^k\})$  is a probability space. We note that  $\tilde{M}(\gamma_i^k)$  is a node-dependent quantity, is non-local, and can be interpreted as a probability measure over the space of walk of fixed length originated at a node.

<sup>3</sup>The author is happy to disclose the code made to generate these simulations upon request.

Since the goal is to introduce an entropy over such probability space, we note that the first possibility for such an entropy is given by:

$${}^{(k+1)}\vec{S}_i = -\frac{1}{k} \sum_{\{\gamma_i^k\}} \tilde{M}(\gamma) \log(\tilde{M}(\gamma)); \quad (21)$$

However, due to the Cesàro mean rule, in the limit  $k \rightarrow \infty$  the entropy becomes independent of the initial node  $i$ . Although this quantity, while being important, does not provide a measure which differentiates the nodes. An alternative normalization has thus to be considered.

As shown in Caravelli (2014), interesting properties arise if one chooses a different and multiplicative normalization. We introduce the following normalization:

$${}^{k+1}\vec{S}_\epsilon = -\epsilon^{k-1} \sum_{\{\gamma_i^k\}} \tilde{M}(\gamma) \log(\tilde{M}(\gamma)), \quad (22)$$

where  $\tilde{M}(\gamma)$  is the probability associated with the trajectory  $\gamma$ . The parameter  $\epsilon$  can be chosen arbitrarily, but in order to guarantee convergence in the limit  $k \rightarrow \infty$ , it is necessary to impose  $\epsilon < 1$ . We note that evaluating the entropy from equation (21) is computationally hard, as it involves the evaluation of an exponential number of trajectories of length  $k$ . Thankfully, such entropy satisfies a recursion relation in terms of the Markov transition matrix  $M$ ,  ${}^1\vec{S}$ , and the parameter  $\epsilon < 1$ ,

$${}^{k+1}\vec{S}_\epsilon = \epsilon^k \left( \frac{M}{\epsilon^{k-1}} {}^k\vec{S} + {}^1\vec{S} \right). \quad (23)$$

If we write down all the terms, recursively, we find the more compact formula,

$${}^{k+1}\vec{S}_\epsilon = \sum_{n=0}^k \epsilon^n M^n {}^1\vec{S}, \quad (24)$$

and realizing that we can take the limit  $k \rightarrow \infty$  safely, we obtain a closed expression:

$${}^*\vec{S}_\epsilon = \frac{1}{I - \epsilon M} {}^1\vec{S}, \quad (25)$$

which is finite for all values of  $\epsilon < 1$ , is a fixed point of equation (23), and can be interpreted as a non-local notion of entropy. We note that now each node can have different values of the entropy depending on the structure of the Markov transition matrix  $M$ . The strength of equation (25) is that it is now easier to evaluate the entropy, rather than equation (22). This advantage comes however at the price of a free parameter  $\epsilon$ , that can be given an *a posteriori* interpretation if we consider this as a probability over the space trajectories. Using this interpretation, the average length of a trajectory becomes

$$\langle k \rangle = \sum_{k=1}^{\infty} \epsilon^k k = \frac{\epsilon}{(1 - \epsilon)^2}, \quad (26)$$

which can be inverted for any  $\langle k \rangle > 0$ , for  $\epsilon(\langle k \rangle)$ . In this paper, we consider  $\epsilon$  such that  $\langle k \rangle = N$ , the total number of nodes of the

networks. In general, the limit  $\epsilon \rightarrow 0$  recovers the node entropy considered in other works, as for instance in Pershin and Di Ventra (2013), of which ours can be considered as a non-local extension. We will study this extended notion of graph entropy to characterize the dynamics of memristive circuits.

It is worth discussing explicitly few properties of the entropy of equation (25). We observe first that the limit  $\lim_{\epsilon \rightarrow 1} {}^*\vec{S}_\epsilon$  is unbounded. This is due to the fact that the Perron root of a Markov chain is 1 and that thus the resolvent of the operator does not exist. However, equation (22) for finite  $k$  is indeed bounded from above by  $\epsilon^{k-1} k \log(N)$ , which is the extreme value of the entropy for finite  $k$ , corresponding to a complete graph with  $N$  nodes. We note that this entropy goes to infinity in the limit  $k \rightarrow \infty$  if and only if  $\epsilon \geq 1$ . If we set  $\langle k \rangle = N$ , we can obtain an approximate upper bound for the fixed point entropy of every node by inverting for  $\epsilon(N)$  in equation (26), and given by

$${}^*S \lesssim 2^{1-N} N \left( \frac{2N + \sqrt{4N + 1} + 1}{N} \right)^{N-1} \log(N), \quad (27)$$

which is, for  $N \gg 1$ , of the order of  $S_{max} \approx N \log(N)$ , which unsurprisingly depends in a non-linear way on the total number of nodes. This means that the maximum of the entropy is not well behaved in the thermodynamic limit ( $N \rightarrow \infty$ ), as  $S_{max}/N$  does not exist. As a general feature of entropic measures, observing a decrease in graph entropy implies that some structure is being generated dynamically, rather than converging toward a random configuration. As a last comment, we note that this definition does not lead to a unique definition of transition probability when the network is directed. In this case, one can define the *forward* probability, as the one used above, or the *backward* probability. For a directed adjacency matrix, one has the in-degree and the out-degree, defined as diagonal matrices,

$$D_{ii}^{in} = \sum_j A_{ij}, \quad (28)$$

$$D_{ii}^{out} = \sum_j A_{ji}, \quad (29)$$

and thus one has two definitions of transition probabilities:

$$M^f = (D^{out})^{-1} A, \quad (30)$$

$$M^b = (D^{in})^{-1} A^T. \quad (31)$$

Both these definitions will be used in the following.

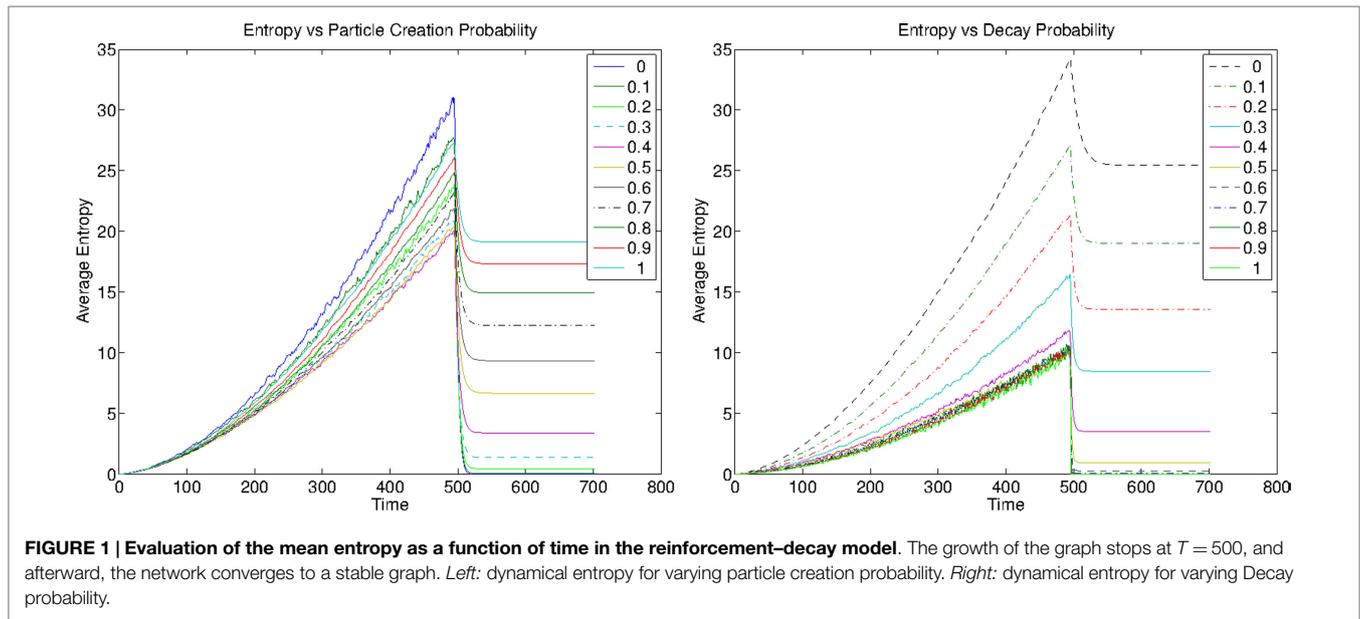
The entropy introduced in this section contains topological informations about the graph flows. Thus, it has the advantage of being *non-local* and with support on the nodes of the graph.

### 3. RESULTS

#### 3.1. Toy Diffusion Model

We analyze the topological properties of evolving graphs under the reinforcement-decay model using the graph entropy of Section 2.3.

In particular, we simulate the system for various values of the two main parameters: the particles creation probability  $p_p$  and



the decay probability  $p_d$ . We plot the values of the entropy as a function of time in **Figure 1**. These were obtained after averaging over 30 simulations, for fixed parameters  $\gamma = 0.1$ ,  $L_d = 1$ . At each time step, the strength of links is decreased with probability  $p_d$  of a value  $\gamma$ , which we set to  $\gamma = 0.1$ . The graph grows at each time step up to a maximum number of nodes  $N_c$ , here chosen to be 500, and then the links which did not reach the critical stable value “evaporate.” The new link parameter  $p_{nl}$  is set to 0.1 in the simulations. For each new node, a new particle is added to the graph at the new node with probability  $p_p$ . The growth halts artificially to observe how the graph evolves afterward, and to see the behavior of the entropy. In **Figure 1**, we show the evolution of the graph entropy for varying  $p_d$  and  $p_p$ .

As a first comment, we note that as the graph grows the mean graph entropy, defined as  $\langle S \rangle = \frac{1}{N} \sum_{i=1}^N \langle S_e \rangle_i$  (i.e., the average over the nodes), grows as well. Although, the mean of the entropy does not have a thermodynamical interpretation, this gives a succinct graphical representation of the outcome. The growth is in general due to both the decrease in the number of nodes and to the lack of structure due to the random growth. In addition, we stress the important effect of the decay, which competes with the reinforcing of hopping particles. Meanwhile decay decreases the amount of order, particles create structure. We note that after the growth stops, the entropy decreases to a stable value. This implies that the effect of decay is of creating “structure” by removing most of the unstable links.

Notably, we observe that for each set of parameters, the graph entropy curve takes different values, both during the growth and after. This implies that such graph entropy measure is enough to characterize the asymptotic state of the graph in the simulations, we have performed. We note that the non-local graph entropy is changing both in the case in which we vary the decay probability parameter, as in **Figure 1** (left), and for the case in which we vary the particle creation probability, as in **Figure 1** (right). Thus, in all

the performed simulations, the average entropy per node is indeed characterizing the parameter space.

### 3.2. Purely Memristive Circuits

In the previous section, we observed that the non-local graph entropy provides characterizes the dynamics of the reinforcement-decay model. We thus use the entropy to study the dynamics of purely memristive circuits.

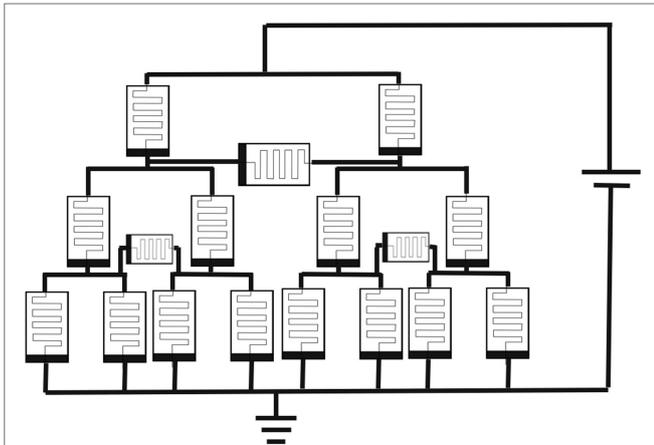
In order to apply the entropy to the case of memristive circuits, we consider the following mapping between a flow on a network and a stochastic matrix. Given a network with a unique directed edge between two nodes, we introduce the matrix  $I_{ij}$ , given by the flow of currents between nodes  $(i, j)$ , and defined as a matrix which is elementwise non-negative. For each couple of nodes  $(i, j)$  connected by a resistance or memristance, only one of the elements of  $I_{ij}$  or  $I_{ji}$  are non-zero, depending on the directionality of the current. Since  $I$  is a non-negative matrix, we can obtain a stochastic matrix by normalizing by rows (columns), obtaining  $\tilde{I}$ , satisfying  $\sum_j \tilde{I}_{ij} = 1$  ( $\sum_i \tilde{I}_{ij} = 1$ ). A notion of entropy on the set of currents was introduced in Pershin and Di Ventra (2013) in order to study the self-organizing properties of memristive circuits. Here, we intend to extend this analysis to understand non-local spatial properties using the graph entropy measure of the previous section.

In order to simulate the dynamical evolution of the memristive circuits, in the present paper, we use nodal analysis (Horowitz and Hill, 1989) to solve for the currents at each time step, and a Runge-Kutta 4 method to implement the dynamics. We then construct the Markov matrix from the resulting currents and calculate the node entropy dynamically.

A particular feature we aim to explore is the role of symmetry. As previously described in **Figure 2** we show the triangular network with  $L = 3$  layers, which can be easily generalized to a higher number of layers. In each layer, there are both horizontal

memristors closing on a triangular mesh, and vertical memristors connecting to a new layer. For a triangular network with  $L$  layers, there are  $3 \cdot (2^L - 1)$  memristors to simulate. In the last layer, the horizontal memristors are not placed, as these have zero voltage difference applied to them due to the connection to the ground. Let us consider first the case in which there is no state decay,  $\alpha = 0$ , and in which all the polarities of the memristors are aligned alike  $p = 1$ . In particular, we simulate homogeneous memristors parameters, with  $R_{on} = 100 \Omega$ ,  $R_{off} = 16000 \Omega$ ,  $\mu = 10^{-3}$ ,  $d = 10^{-11}$ , and with a time step given by  $dt = 10^{-2} \text{ s} = cs$  and with constant external voltage  $V = 20$ .

As a first example, we consider the case  $L = 4$  under the forcing of an AC voltage generator, with  $V(t) = 20 \sin(10t)$ . In **Figure 3** (left panel), we show the entropy evolution for each node (top)

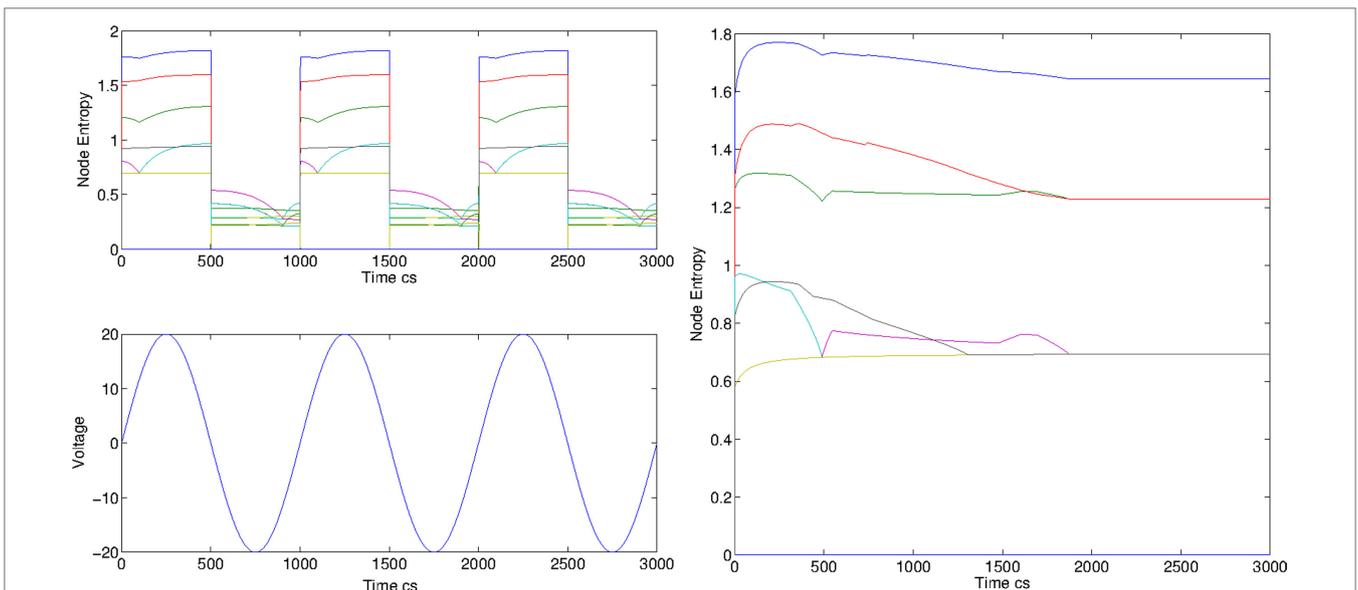


**FIGURE 2 | Memristive circuit defined recursive as on a tree and made of  $L$  layers.** The nodes in the bottom layer are connected to the ground.

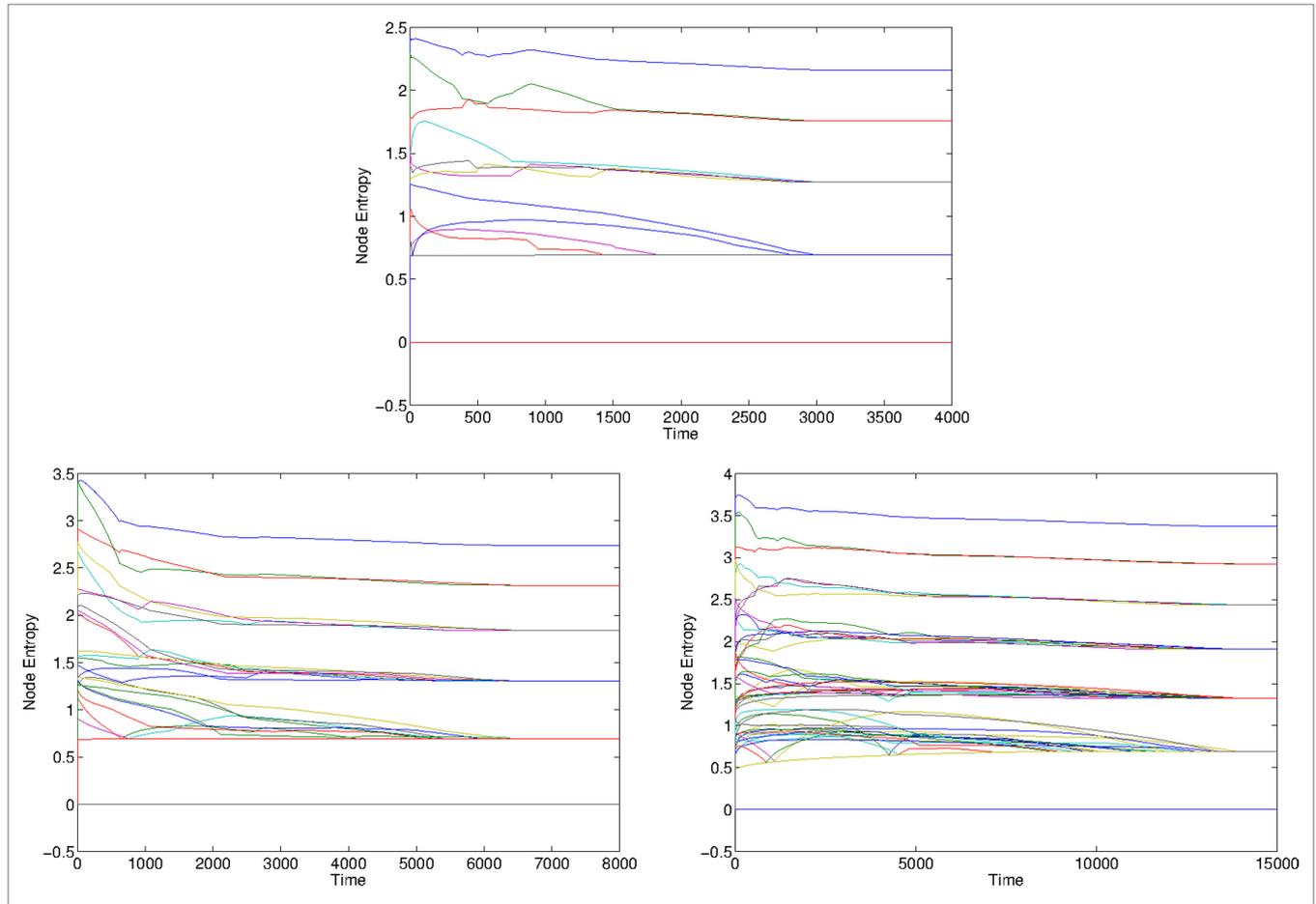
and the applied voltage (bottom), for a random initialization of the memristors state. It is easy to see that the negative and positive voltage sides of the generator correspond to different phases of the circuit. In fact, the node entropy changes dramatically and periodically with time, following the change of currents signs. The case of a DC generator is shown in **Figure 3** (right panel) for an initial random configuration. Interestingly, from the entropy we can identify the four different entropy layers. We note that initially being each memristor randomly initialized, each node has different entropy level. Dynamically, however, the non-local entropy measure is sensitive to changes to far nodes of the network, as it can be easily observed. Each entropy level reaches an asymptotic value which depends on the layer only. The fourth layer has zero value for each node. This is given to the fact that each node is connected to the ground, and only a unique current is connected to each node.

A similar behavior is observed also for the case of a higher number of layers, as shown in **Figure 4** for the cases  $L = 5, 6$ , and  $7$ . Few peculiar things can be noted. The first observation is that the number of layers is equal to the number of asymptotic levels of the entropy, and the number of nodes with that level of entropy corresponds exactly with the number of nodes in that layer (for the  $L$ -th layer, there are  $2^{L-1}$  nodes). Moreover, we note that the time it takes for the entropy to reach its asymptotic value increases with the number of layers. Visually, one can observe that it is  $\approx 20$  s for  $L = 4$ ,  $30$  s for  $L = 5$ ,  $\approx 60$  s for  $L = 6$  and  $\approx 120$  s for  $L = 7$ . This is confirmed in **Figure 5** by calculating the histogram of the asymptotic entropy value for the cases  $L = 4, 5, 6, 7$ .

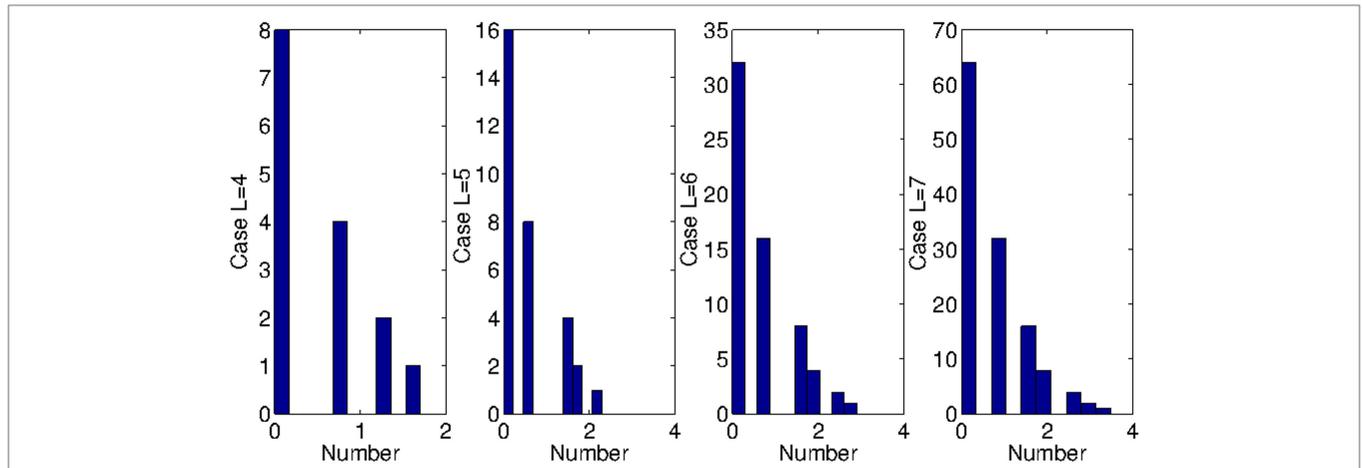
We observed numerically that, independently of the initial condition, the thermalization time increased with the number of layers; we define the thermalization time as the time it takes to the entropy to reach its asymptotic and stable value. Specifically, heuristically we observed that if the frequency of the generator in



**FIGURE 3 | Evolution of the node entropy for AC and DC voltage, measured in centiseconds for random initial conditions for  $L = 4$ .** Left: evolution of the entropy for the tree graph of **Figure 2** when an AC voltage is applied to the network with  $V = 20 \sin(10t)$  volts. Right: entropy evolution for the tree graph of **Figure 2** for a DC voltage  $V = 50$  V.



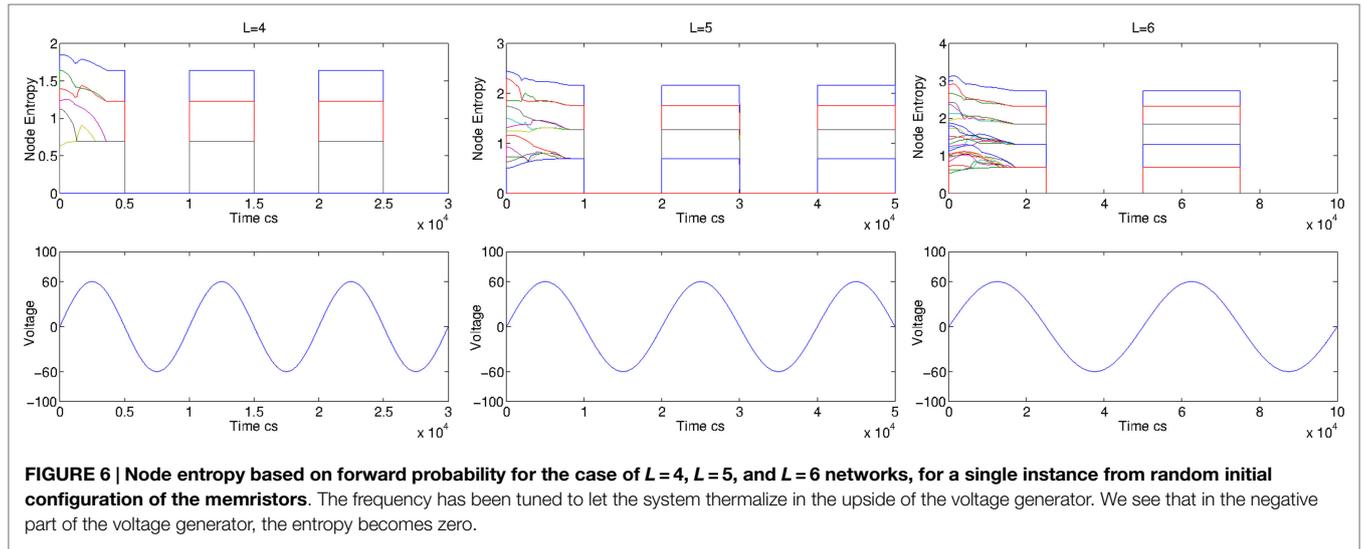
**FIGURE 4 | Graph Entropy for the tree circuit of Figure 2 for  $L = 5$  (top),  $L = 6$  (bottom left), and  $L = 7$  (bottom right), in the case of DC generator, with  $V_0 = 20\text{ V}$ .** The initial configurations are chosen at random. We observe the emergence of  $L$  layers of entropy, showing that the system self-organized in order for the nodes in each layers to have equal entropy.



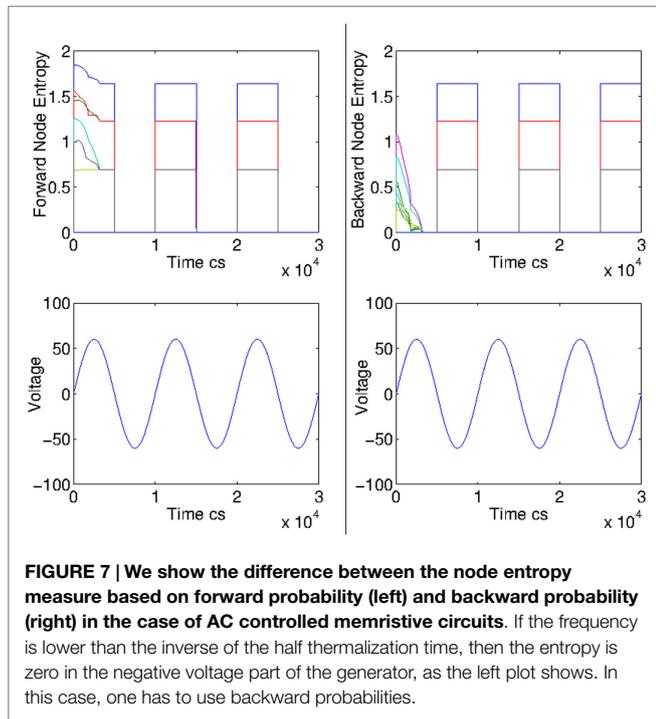
**FIGURE 5 | Histograms of the asymptotic entropy values for  $L = 4, 5, 6, 7$ .** We observe that the maximum entropy increases, and that the number of values on each level grows as  $2^L$ .

the AC mode is longer than the inverse of twice the thermalization time, then in the negative voltage part of the generator, the node entropy becomes zero. This is shown for instance in **Figure 6**

for the case of  $L = 4, L = 5$ , and  $L = 6$ , and where the frequency has been tuned to be larger than the observed thermalization time. In the negative side of the generator, the system can be



**FIGURE 6 | Node entropy based on forward probability for the case of  $L = 4$ ,  $L = 5$ , and  $L = 6$  networks, for a single instance from random initial configuration of the memristors.** The frequency has been tuned to let the system thermalize in the upside of the voltage generator. We see that in the negative part of the voltage generator, the entropy becomes zero.



**FIGURE 7 | We show the difference between the node entropy measure based on forward probability (left) and backward probability (right) in the case of AC controlled memristive circuits.** If the frequency is lower than the inverse of the half thermalization time, then the entropy is zero in the negative voltage part of the generator, as the left plot shows. In this case, one has to use backward probabilities.

described by, instead of the forward probabilities, by the backward probabilities, implying that the stochastic matrix is derived from the transpose of the current matrix  $I_{ij}$ . The difference is shown in the plot of **Figure 7**, which illustrates that the positive voltage of the generator is characterized by the forward probabilities matrix, but the negative side is characterized by the backward probabilities.

Our analysis has important implications. In particular, we observe that asymptotically the memristive circuit considered here converges to a state which respects the symmetry of the network. The fact it converges to a specific value, although surprisingly unique and robust, can be thought as a feature of the dynamical system. Moreover, since it starts from random

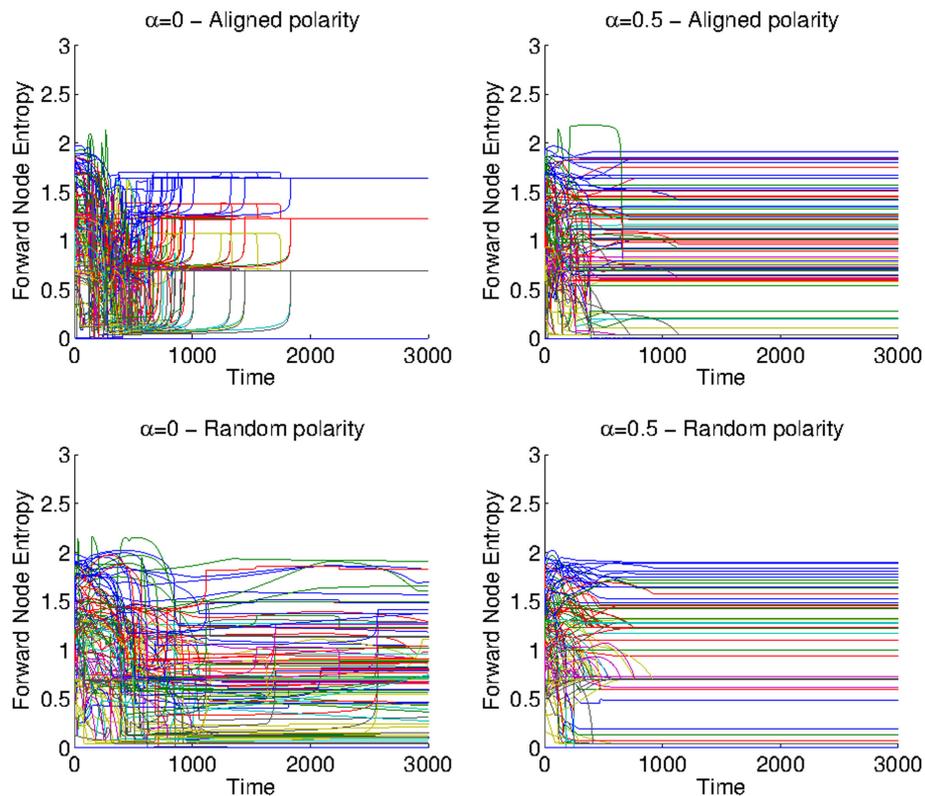
configuration of the memristors, this implies that the symmetry-respecting state is a robust attractor of the system and with a large basin of attraction. We also observe the network rearranges itself in order to make superfluous some of the memristances. It is easy in fact to realize that since the nodes in a layer must be equal, the memristances arranged horizontally and belonging to the same layer are connected to nodes of equal voltage. This implies that no current is flowing in these memristances.

An extension of the model above can be made to account for both state decay and different polarities. In order to check whether the state is truly symmetry respecting, we simulate for the case  $L = 4$  with the polarity assigned at random and with equal probability. In the top row of **Figure 8**, we show the case with aligned polarities for  $V = 50$  Volts.

In **Figure 8** (top left), we show the case without state decay for constant  $V = 50V$ ,  $\alpha = 0$ , for comparison with the case  $\alpha = 0.5$ , which is shown in **Figure 8** (top right). In general, we observe that both state decay and switched polarities remove the symmetry-respecting properties of the dynamics.

The bottom row of **Figure 8** shows the case with randomly assigned polarities, both for the cases without (bottom left,  $\alpha = 0$ ) and with (bottom right,  $\alpha = 0.5$ ) state decay. The difference with the top left picture is notable. Both in the cases with switched polarities and decay, the network symmetry is lost, as now each memristor can have two different polarities.

However, a careful analysis shows that this not true for all values of the decay constant, as we see in **Figure 9**. We observe that the state decay does not modify the asymptotic value of the entropy for small values ( $\alpha \approx 0.01$ ) and large values ( $\alpha \approx 100$ ) of the decay constant; although the behavior of the resistances is different in the two cases, as we show in **Figure 9** (left and right). We observe however the dynamics of the entropy is clearly different, showing that the two models are inequivalent. The case of intermediate ( $\alpha \approx 1$ ) values of the decay constant are shown in **Figure 9** (center), a dependence on the initial condition is present. In general, in this case the resistances converge either to  $R_{on}$  and  $R_{off}$  asymptotically, and the entropy is observed to take different values from those observed for  $\alpha = 0$ .



**FIGURE 8 | Comparison between simulations without state decay and aligned polarities (top left), only state decay  $\alpha = 0.5$  (top right), randomized polarities (bottom left), and randomized polarities with state decay (bottom right).** The input voltage is  $V_0 = 50$  V. The plots were obtained from plotting all the results from 100 Monte Carlo simulations with similar dynamical parameters but where the initial states have been randomized.

To conclude, we provide a sensitivity analysis of the graph entropy as a function of the applied voltage in the case of a DC generator, no state decay and aligned polarities. In **Figure 10**, we plot the asymptotic entropy values for each node in the case  $L = 5$  (left) and the derivative with respect to the voltage (right). The plots have been obtained after a Monte Carlo and averaged over 100 initial random initialization of the memristor internal parameters. We note that there is a value of the voltage applied by the generator, which we identify as  $V_c$ , after which the asymptotic entropy does not change. Although we stress that the actual value of  $V_c$  depends on the parameters characterizing the memristors, after varying the parameters we find a similar behavior. In **Figure 10** (right), we see that the derivative of the entropy becomes zero at  $V_c \approx 150$  V. Further analysis, beyond the scope of this paper, is required to understand whether this transition presents any form of critical behavior.

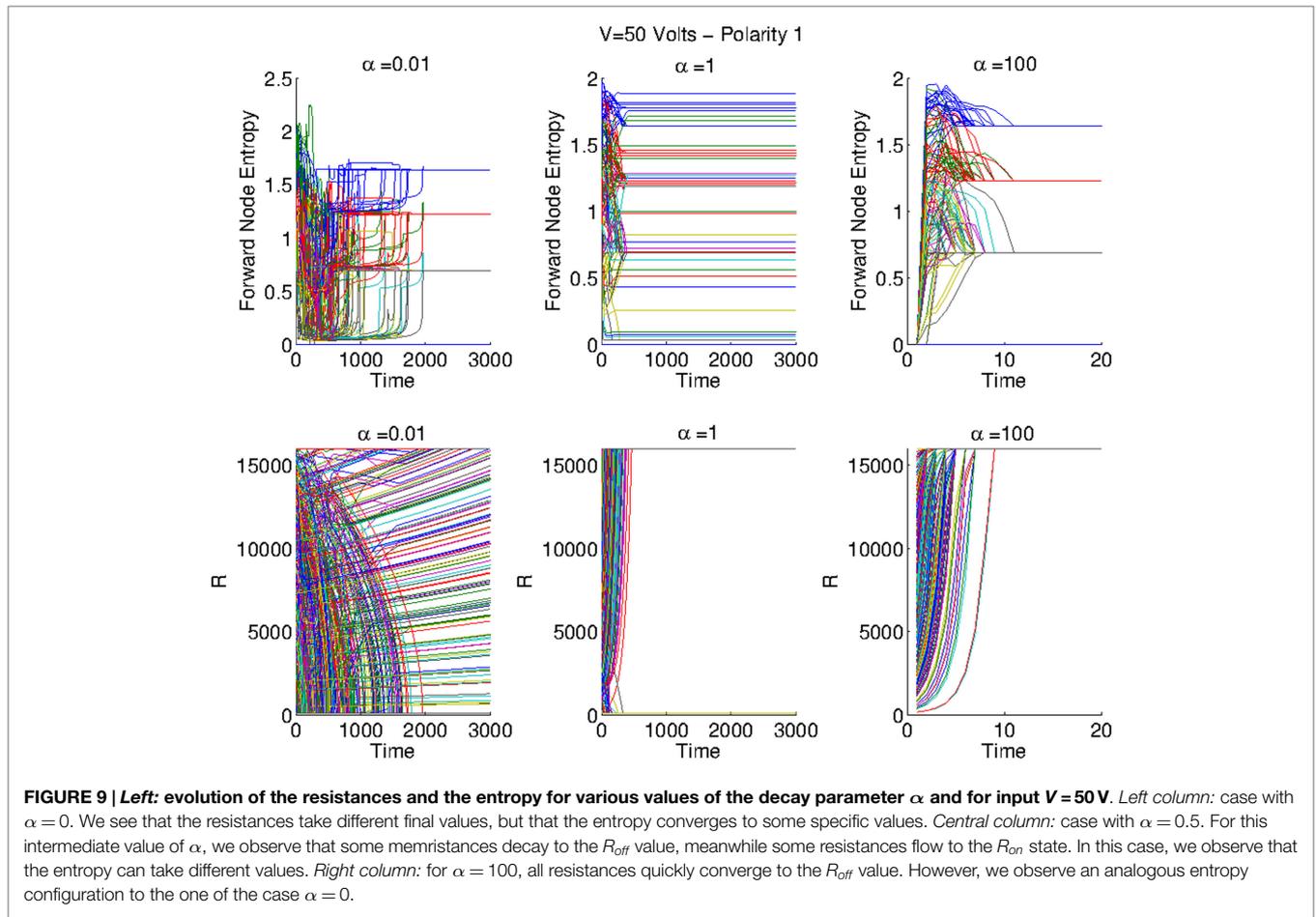
## 4. DISCUSSION

In this paper, we applied the entropy measure introduced in Caravelli (2014) to study the dynamics of evolving graphs with memory. In particular, we studied the toy model introduced in Caravelli et al. (2015), inspired by evaporating ant pheromone trails, a process known to be able to solve problems as finding the shortest path between their nest and food. The pheromone track has a characteristic decay time but is reinforced every time by ants.

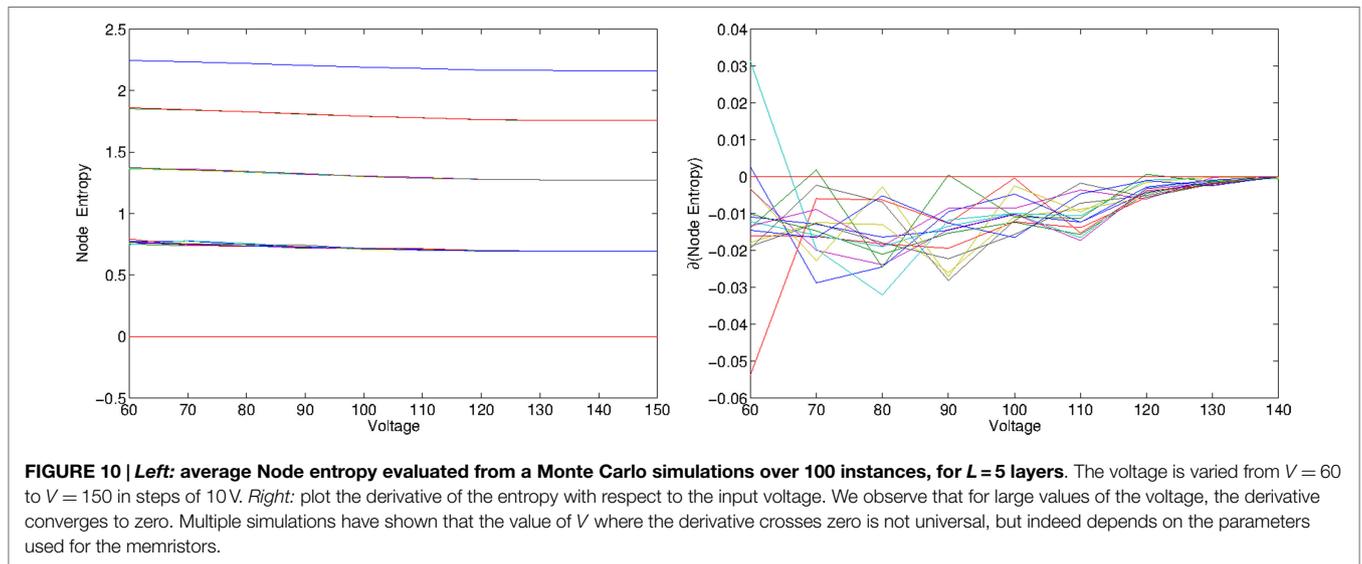
We have also applied the non-local entropy measure to the case of pure memristive circuits (Chua, 1971; Strukov et al., 2008). A local version of the entropy used in this paper was in fact introduced in Pershin and Di Ventra (2013) to study the self-organization properties of memristors. This non-local extension can be thought of as the centrality operator applied to the local definition of entropy of a node in a graph and depends on an external constant which controls the amount of non-locality. We fixed this parameter to allow loops large enough to enclose all the nodes in the network. This also fixes the maximum entropy one can expect from a node, and we have provided a formula for this extreme value.

The evolution of the node entropy was studied numerically both for the reinforcement–decay model and for the case of memristive circuits in a tree-like structure. We considered these two models and their relaxation to the asymptotic entropy values for each node. Although our results not being of general character, we have observed that the non-local entropy measure studied here characterizes the asymptotic graph structure. For the case of the reinforcement–decay case, we have found that for the entropy distinguishes dynamically different evolutions with different parameters.

The same analysis has been performed for the case of memristive circuits, and applied to the specific case of tree-like structures which can be defined recursively, and shown in **Figure 4**. In fact, in general the entropy reveals that the memristive circuit self-organizes, reaching a configuration in which each node in the



**FIGURE 9 | Left: evolution of the resistances and the entropy for various values of the decay parameter  $\alpha$  and for input  $V = 50$  V.** *Left column:* case with  $\alpha = 0$ . We see that the resistances take different final values, but that the entropy converges to some specific values. *Central column:* case with  $\alpha = 0.5$ . For this intermediate value of  $\alpha$ , we observe that some memristances decay to the  $R_{off}$  value, meanwhile some resistances flow to the  $R_{on}$  state. In this case, we observe that the entropy can take different values. *Right column:* for  $\alpha = 100$ , all resistances quickly converge to the  $R_{off}$  value. However, we observe an analogous entropy configuration to the one of the case  $\alpha = 0$ .



**FIGURE 10 | Left: average Node entropy evaluated from a Monte Carlo simulations over 100 instances, for  $L = 5$  layers.** The voltage is varied from  $V = 60$  to  $V = 150$  in steps of 10V. *Right:* plot the derivative of the entropy with respect to the input voltage. We observe that for large values of the voltage, the derivative converges to zero. Multiple simulations have shown that the value of  $V$  where the derivative crosses zero is not universal, but indeed depends on the parameters used for the memristors.

$k$ -th layer has the same entropy. We distinguished the case of AC and DC controlled circuits. In the case of AC circuits, we have noted that there are two regimes: the one in which the frequency is larger or smaller than the inverse of half the thermalization time. If the frequency in the AC-controlled circuit is low enough, the

circuit thermalizes, and in the negative voltage side of the sinusoid the entropy of the network becomes zero. We have shown this is an artifact of the entropy, and that it is necessary to consider both forward and backward probabilities in the definition of the Markov transition matrix.

An observed feature is that the network re-organizes to respect the graph symmetries. We observed that this statement is true as long as the assigned polarities are aligned and one does not consider the effect of state decay. We considered the case in which the polarities of the memristors were chosen at random, and the dynamics of the internal parameter has been extended to consider state decay. In both cases, the symmetry of the graph is destroyed, and thus, one observes that the asymptotic entropy configuration can take any value depending on the initial condition. This has been observed to be true only for values of the decay constant of order one, but for small values, the self-organizing properties of memristors persist.

We have shown that in general the asymptotic level of the entropy for each node depends, in the DC case, on the voltage applied. We have seen that if the voltage is high enough, the entropy converges to a fixed value. This value changes depending on the parameter of the model, and thus is not universal. The analysis of this phenomenon goes beyond the scope of this paper, and will be analyzed in future works, but shows that the entropy considered in this paper has interesting properties.

The findings of this article suggest that non-local measures of self-organization are not only useful at *compressing* information about the properties of graphs which are changing dynamically, but that these can provide several insights about the dynamics itself. We should stress that these findings cannot be made general (i.e., to graphs and model other than the one studied), and they turn out being only descriptive, rather than *predictive*. However,

we believe that the graph entropy studied here does provide a compact way of obtaining information about the network to classify the dynamics. The entropy values considered here scale linearly in the number of nodes  $N$ , although in principle, the number of dynamical variables scale quadratically in  $N$ . For graphs in which the number of edges is much larger than the number of nodes, this might allow for instance the use of *heat maps* to classify and study the type of dynamics.

In much more general terms, our study provides a connection between self-organization, non-locality, and entropy-based measures. We thus hope our work will motivate further theoretical and experimental studies along these directions.

## AUTHOR CONTRIBUTIONS

FC has designed the work, derived the main analytical results, and performed the simulations.

## ACKNOWLEDGMENTS

FC would like to thank Alioscia Hama, Massimiliano Di Ventra, and Fabio Traversa for discussion on the results of this work, and the anonymous referee for raising few interesting points which improved the quality of the paper.

## FUNDING

FC acknowledges support from Invenia Labs.

## REFERENCES

- Albert, R., and Barabasi, A. L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74, 47. doi:10.1103/RevModPhys.74.47
- Anand, K., and Bianconi, G. (2009). Entropy measures for complex networks: toward an information theory of complex topologies. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 80, 045102. doi:10.1103/PhysRevE.80.045102
- Avizienis, A. V., Sillins, H. O., Martin-Olmos, C., Shieh, H. H., Aono, M., Stieg, A. Z., et al. (2012). Neuromorphic atomic switch networks. *PLoS ONE* 8:e42772. doi:10.1371/journal.pone.0042772
- Barrat, A., Barthelemy, M., and Vespignani, A. (2009). *Dynamical Processes on Complex Networks*. Cambridge: Cambridge University Press.
- Braunstein, S. L., Ghosh, S., and Severini, S. (2006). The laplacian of a graph as a density matrix: a basic combinatorial approach to separability of mixed states. *Ann. Comb.* 10, 291–317. doi:10.1007/s00026-006-0289-3
- Buhl, J., Gautrais, J., Solé, R. V., Kuntz, P., Valverde, S., and Deneubourg, J. L. (2004). Efficiency and robustness in ant networks of galleries. *Eur. Phys. J. B* 42, 123–129. doi:10.1140/epjb/e2004-00364-9
- Caldarelli, G. (2007). *Scale-Free Networks*. Oxford: Oxford University Press.
- Caravelli, F. (2014). Ranking nodes according to their path-complexity. *Chaos Solitons Fractals* 73, 90–97. doi:10.1016/j.chaos.2014.12.021
- Caravelli, F., Hama, A., and Di Ventra, M. (2015). Scale-free networks as an epiphenomenon of memory. *Europhys. Lett.* 109, 28006. doi:10.1209/0295-5075/109/28006
- Carbajal, J. P., Dambre, J., Hermans, M., and Schrauwen, B. (2015). Memristor models for machine learning. *Neural Comput.* 27, 725–747. doi:10.1162/NECO\_a\_00694
- Chrysafis, O., and Cannings, C. (2009). Weighted self-similar networks under preferential attachment. *Physica A* 388, 2965–2974. doi:10.1016/j.physa.2009.03.030
- Chua, L. O. (1971). Memristor—the missing circuit element. *IEEE Trans. Circuits Theory* 18, 507–519. doi:10.1109/TCT.1971.1083337
- Di Ventra, M., Pershin, Y., and Chua, L. (2009). Circuit elements with memory: memristors, memcapacitors and meminductors. *Proc. IEEE* 97, 1717. doi:10.1109/JPROC.2009.2022882
- Di Ventra, M., and Pershin, Y. V. (2013a). On the physical properties of memristive, memcapacitive and meminductive systems. *Nanotechnology* 24, 255201. doi:10.1088/0957-4484/24/25/255201
- Di Ventra, M., and Pershin, Y. V. (2013b). The parallel approach. *Nat. Phys.* 9, 200–202. doi:10.1038/nphys2566
- Dorogovtsev, S. N., Goltsev, A. V., and Mendes, J. F. F. (2002). Pseudofractal scale-free web. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 65, 066122. doi:10.1103/PhysRevE.65.066122
- Estrada, E. (2011). *The Structure of Complex Networks: Theory and Applications*. Oxford: Oxford University Press.
- Evans, T. S., and Saramaeki, J. P. (2005). Scale free networks from self-organisation. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 72, 026138. doi:10.1103/PhysRevE.72.026138
- Horowitz, P., and Hill, W. (1989). *The Art of Electronics*, 2nd Edn. Cambridge: Cambridge University Press.
- Ikeda, N. (2008). Network formation determined by the diffusion process of random walkers. *J. Phys. A Math. Theor.* 41, 235005. doi:10.1088/1751-8113/41/23/235005
- Kleinberg, J. M. (1998). Authoritative sources in a hyperlinked environment. *J. ACM*, 46, 604–632. doi:10.1145/324133.324140
- Kloeden, P., and Platen, E. (1999). *Numerical Solution of Stochastic Differential Equations*, Vol. 23. Berlin: Springer.
- Li, M., Gao, L., Fan, Y., Wu, J., and Di, Z. (2010). Emergence of global preferential attachment from local interaction. *New J. Phys.* 12, 043029. doi:10.1088/1367-2630/12/4/043029
- Lindgren, K. (1988). Microscopic and macroscopic entropy. *Phys. Rev. A* 38, 9. doi:10.1103/PhysRevA.38.4794
- Lloyd, S., and Pagels, H. (1988). Complexity as thermodynamic depth. *Ann. Phys.* 188, 186–213. doi:10.1016/0003-4916(88)90094-2

- Newman, M. E. J. (2010). *Networks: An Introduction*. Oxford: Oxford University Press.
- Ohno, T., Hasegawa, T., Nayak, A., Tsuruoka, T., Gimzewski, J. K., and Aono, M. (2011). Sensory and short-term memory formations observed in a ag2s gap-type atomic switch. *Appl. Phys. Lett.* 99, 203108. doi:10.1063/1.3662390
- Pershin, Y., and Di Ventra, M. (2011). Solving mazes with memristors: a massively parallel approach. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 84, 046703. doi:10.1103/PhysRevE.84.046703
- Pershin, Y., and Di Ventra, M. (2013). Self-organization and solution of shortest-path optimization problems with memristive networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.* 88, 013305. doi:10.1103/PhysRevE.88.013305
- Saramaeki, J. P., and Kaski, K. (2004). Scale-free networks generated by random walkers. *Physica A* 341, 80–86. doi:10.1016/j.physa.2004.04.110
- Schweitzer, F., Lao, K., and Family, F. (1997). Active random walkers simulate trunk trail formation by ants. *BioSystems* 41, 153–166. doi:10.1016/S0303-2647(96)01670-X
- Strukov, D. B., Snider, G., Stewart, D. R., and Williams, R. S. (2008). The missing memristor found. *Nature* 453, 80–83. doi:10.1038/nature06932
- Traversa, F. L., Ramella, C., Bonani, F., and Di Ventra, M. (2015). Memcomputing np-complete problems in polynomial time using polynomial resources and collective states. *Sci. Adv.* 1:e1500031. doi:10.1126/sciadv.1500031
- Watts, D. J., and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature* 393, 440–442. doi:10.1038/30918
- Conflict of Interest Statement:** The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.
- Copyright © 2016 Caravelli. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.