Attractor Regimes of Boolean Recurrent Neural Networks subject to STDP and Global Plasticity

Jérémie Cabessa[1] **and Alessandro E.P. Villa**[2]

[1]DAVID Lab, University of Versailles (UVSQ) – Paris-Saclay, 78035 Versailles, France
[2]NeuroHeuristic Research Group, DESI – HEC, University of Lausanne (UNIL), CH-1015 Lausanne, Switzerland

jeremie.cabessa@uvsq.fr alessandro.villa@unil.ch

***Abstract*—Attractor dynamics underlie memory mechanisms in both neuroscience and machine learning. This study explores the attractor dynamics of Boolean recurrent neural networks governed by two plasticity mechanisms: spike-timing-dependent plasticity (STDP), driven by random background activity, and global plasticity (GP), triggered by specific patterns. In the Boolean framework, attractors can be explicitly computed and quantified for small networks. We demonstrate theoretically and experimentally that Boolean networks can exhibit a vast number of attractors, each encoding a potential memory. The attractor count is strongly modulated by synaptic strengths, emphasizing the pivotal role of plasticity. While STDP predominantly reduces attractor regimes, GP significantly enhances them. Furthermore, attractor regimes can be stabilized through carefully designed input streams, even in the presence of STDP. The processes of gaining, losing, and stabilizing attractors are interpreted as mechanisms of learning, memory loss, and memory consolidation, respectively. These findings highlight the complementary roles of local and global plasticity in shaping high-capacity attractor landscapes.**

***Index Terms*—Boolean recurrent neural networks, finite state automata, attractors, attractor regimes, memory, synaptic plasticity, STDP, learning, neural computation.**

I. Introduction

Attractor dynamics have become a foundational concept in both theoretical and applied machine learning, as well as in neuroscience. In machine learning, attractors are commonly used to implement associative memory mechanisms. Networks are trained so that attractor states correspond to minima of some energy function. Given some initial state, the network dynamics converge to an attractor state that corresponds to the retrieved pattern.

Hopfield networks (HNs) are foundational recurrent neural networks where stable fixed-point attractors correspond to stored memory patterns [1]. The Hopfield model has been extended to include asymmetric connections [2]. This framework has been expanded into modern Hopfield networks, which implement dense associative memory models by using unusual activation functions or, equivalently, by including higher-order interactions in the energy function [3], [4]. More recently, a novel continuous energy function and associated attention-based update rule have been introduced, leading to exponential storage capacity and extremely fast convergence [5].

Boltzmann machines (BMs) generalize Hopfield networks by incorporating stochasticity in their dynamics, enabling them to model complex probability distributions [6]. Restricted Boltzmann Machines (RBMs) enhance BMs by reducing computational complexity [7]. RBMs have been pivotal in unsupervised learning and generative models.

Randomly connected neural networks were early shown to exhibit chaotic attractor dynamics [8]. This dynamic richness forms the foundation of Reservoir Computing (RC). Within this framework, Echo State Networks (ESNs) and Liquid State Machines (LSMs) utilize randomly connected recurrent networks and spiking neural networks as fixed reservoirs, respectively [9], [10], leveraging transient attractor dynamics to encode temporal patterns [11]. By contrast, FORCE learning adapts the reservoir through training, enabling the network to follow desired dynamic trajectories [12].

Continuous Attractor Neural Networks (CANNs) encode continuous variables as a continuum of stable attractor states, with the number of neuronal units increasing [13], [14]. These models enable neural representations of continuous features, such as orientation, motion direction, head direction, and spatial location.

In contrast, Localist Attractor Networks typically represent distinct categories or concepts by associating localized attractor units with specific items [15]. This approach can enhance model interpretability and offer computational advantages compared to distributed attractor networks. Reconsolidation Attractor Networks (RANs) generalize localist attractor networks by adding the capability to dynamically update attractors through a process known as reconsolidation [16].

On the neuroscience side, attractors refer to stable firing patterns in the cerebral cortex, and have been crucially linked to long-term and short-term memory, attention, decision-making, and mental disorders [17]. A recent review discusses the mechanisms of attractor formation, discrete and continuous attractors, role of attractors in representation and memory, and evidence of attractors in the brain, conclusively stating that the brain constructs and uses attractor dynamics for computation [18] (and the references therein).

The theory of *cell assemblies* supports the existence of attractor dynamics in biological neural networks. This theory

posits that mental concepts are represented in the brain by strongly interconnected assemblies of neurons [19]–[21]. Cell assemblies (CAs) exhibit self-sustaining persistent activity, referred to as reverberation. The formation of CAs is associated with long-term memory (LTM) encoding, their persistent activity supports short-term memory (STM), and their ignition enables processes such as pattern completion [20], [22]. The existence of cell assemblies and the prevalence of population coding are strongly supported by neuroanatomical, neurophysiological, biophysical, and psychological evidence, as well as computer simulations and theoretical insights [22]–[24].

Synfire chains and synfire rings are specific types of neural assemblies that also support attractor dynamics in neural networks. Synfire chains are neural structures organized in an almost fully connected feedforward manner, enabling volleys of spikes to propagate through successive layers in a synchronized, robust, temporally precise, and self-sustained fashion (in the case of synfire rings) [25]–[29]. Synfire chains and rings have been argued to play a crucial role in information processing and coding in the brain [30]–[35]. These considerations underscore the crucial role of attractor dynamics in neural networks.

In this work, we focus on the attractor dynamics of Boolean recurrent neural networks, building on previous studies [36], [37]. Specifically, we examine Boolean neural networks influenced by two types of plasticity: spike-timing-dependent plasticity (STDP), driven by random background activity, and global plasticity (GP), associated with trigger patterns. In this Boolean context, unlike most related works, attractors can be explicitly computed and counted, for small networks [37].

Our results demonstrate that Boolean neural networks can exhibit a huge number of attractors, each corresponding to a potential memory. The number of attractors fluctuates in response to synaptic changes. Specifically, we show that STDP tends to reduce the attractor regimes of the networks, whereas global plasticity can significantly expand them. When presented with input streams aligned with current attractors, high attractor regimes can be effectively stabilized over extended periods, even in the presence of STDP. The processes of gaining, losing, and stabilizing attractors are interpreted as mechanisms of learning, memory loss, and memory consolidation, respectively. These findings support the idea that achieving high attractor regimes relies on the interplay between complementary plasticity mechanisms. [38].

The paper is organized as follows: Section II reviews the related works. Section III outlines our model, including Boolean networks, attractors, and synaptic plasticity mechanisms. Section IV presents the results, which are discussed in the concluding Section V. The code is publicly available at the following GitHub repository.

II. Related Works

The correspondence between attractors in Boolean networks and cycles in their corresponding dynamical automata has been established and studied in previous works [36], [37], [39]. Chaotic itinerancy in neural networks refers to the phenomenon where the system transitions through a series of quasi-attractors, exhibiting chaotic dynamics between stable states [40], [41]. Additionally, the combined effects of spike-timing-dependent and homeostatic plasticity mechanisms on time-varying attractors in neural networks have been investigated [38].

Our work, which investigates how Boolean networks transition between attractor regimes under the influence of both local and global plasticity mechanisms, aligns with these studies.

III. Model

A. Boolean Recurrent Neural Networks

A *Boolean recurrent neural networks (BRNN)* is a recurrent neural network composed of binary input and reservoir cells [42]. Formally, a BRNN is a tuple

$$\mathcal{N} = \left(U, X, \left\{\mathbf{W}^{[\mathbf{res}]}(\mathbf{t})\right\}_{t\geq 0}, \left\{\mathbf{W}^{[\mathbf{in}]}(\mathbf{t})\right\}_{t\geq 0}, \left\{\mathbf{b}(\mathbf{t})\right\}_{t\geq 0}\right)$$

where

- $U = \{1, \ldots, M\}$ are the indices of M input cells;
- $X = \{1, \ldots, N\}$ are the indices of N reservoir cells;
- $\mathbf{W}^{[\mathbf{in}]}(\mathbf{t}) \in \mathbb{R}^{N\times M}$ is the weight matrix from the input to the reservoir cells at time t;
- $\mathbf{W}^{[\mathbf{res}]}(\mathbf{t}) \in \mathbb{R}^{N\times N}$ is the recurrent weight matrix of the reservoir cells at time t;
- $\mathbf{b}(\mathbf{t}) \in \mathbb{R}^{N}$ is the bias vector of the reservoir cells at time t.

The network $\mathcal{N}$ is said to be *static* if the weights and biases $\mathbf{W}^{[\mathbf{in}]}(\mathbf{t})$, $\mathbf{W}^{[\mathbf{res}]}(\mathbf{t})$ and $\mathbf{b}(\mathbf{t})$ remain constant over time t; it is called *evolving* otherwise. A Boolean recurrent neural network is illustrated in Figure 1.

The dynamics of $\mathcal{N}$ is given by the system of equations

$$\mathbf{x}(\mathbf{t}+\mathbf{1}) = \theta\left(\mathbf{W}^{[\mathbf{res}]}(\mathbf{t})\cdot\mathbf{x}(\mathbf{t}) + \mathbf{W}^{[\mathbf{in}]}(\mathbf{t})\cdot\mathbf{u}(\mathbf{t}) + \mathbf{b}(\mathbf{t})\right) \quad (1)$$

where $\mathbf{u}(\mathbf{t}) \in \mathbb{B}^M$ is the binary input of $\mathcal{N}$ at time t, $\mathbf{x}(\mathbf{t}) \in \mathbb{B}^N$ and $\mathbf{x}(\mathbf{t}+\mathbf{1}) \in \mathbb{B}^N$ are states of $\mathcal{N}$ at time t and $t+1$, respectively, and

$$\theta(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

is the hard-threshold activation function applied componentwise.

Any infinite sequence of inputs

$$\bar{\mathbf{u}} = \left(\mathbf{u}(\mathbf{0}), \mathbf{u}(\mathbf{1}), \mathbf{u}(\mathbf{2}), \ldots\right) \in \left(\mathbb{B}^M\right)^\infty$$

induces a corresponding infinite sequence of states of $\mathcal{N}$

$$\mathcal{N}(\bar{\mathbf{u}}) = \left(\mathbf{x}(\mathbf{0}), \mathbf{x}(\mathbf{1}), \mathbf{x}(\mathbf{2}), \ldots\right) \in \left(\mathbb{B}^N\right)^\infty$$

where $\mathbf{x}(\mathbf{0})$ is the initial state and $\mathbf{x}(\mathbf{t}+\mathbf{1})$ is given by Equation (1), for all $t > 0$. The sequence $\mathcal{N}(\bar{\mathbf{u}})$ is called the *dynamics* of $\mathcal{N}$ for the input stream $\bar{\mathbf{u}}$.

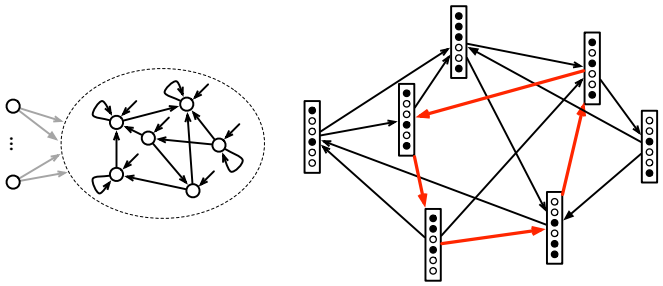

Fig. 1: A Boolean recurrent neural network (left) and its corresponding automaton (right). The different dynamics of the network correspond to the different paths in the automaton. The attractors of the network correspond to the cycles in the automaton.

B. Attractors

An *attractor* of a BRNN $\mathcal{N}$ is a set of states where the network's dynamics can become confined, without necessarily exhibiting periodic behavior (see Figure 2). Formally, a set of states $A = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k\} \subseteq \mathbb{B}^N$ is an *attractor* of $\mathcal{N}$ if there exists an input stream $\bar{\mathbf{u}} = (\mathbf{u}(0), \mathbf{u}(1), \mathbf{u}(2), \ldots)$ and an index $i_0 \in \mathbb{N}$ such that the induced network dynamics $\mathcal{N}(\bar{\mathbf{u}}) = (\mathbf{x}(0), \mathbf{x}(1), \mathbf{x}(2), \ldots)$ satisfies $\mathbf{x}(i) \in A$ for all $i \geq i_0$.

It was early established that Boolean recurrent neural networks are computationally equivalent to finite state automata [42]–[44]. For any Boolean network, there exists a finite automaton that is computationally equivalent, and vice versa. The translation from a given network $\mathcal{N}$ to its corresponding automaton $\mathcal{A}$ is given as follows: the states of $\mathcal{A}$ correspond to the internal states of $\mathcal{N}$, and there is a transition from state $\mathbf{x}$ to $\mathbf{x}'$ labeled by $\mathbf{u}$ in $\mathcal{A}$ if and only if $\mathcal{N}$ transitions from $\mathbf{x}$ to $\mathbf{x}'$ when receiving input $\mathbf{u}$ according to Equation (1). This construction is illustrated in Figure 1. The reverse translation from a given automaton $\mathcal{A}$ to its corresponding network $\mathcal{N}$ is of no interest for our purposes [44].

According to this construction, the different dynamics of $\mathcal{N}$ correspond precisely to the infinite paths in $\mathcal{A}$. In particular, the cyclic dynamics – i.e., the *attractors* – of $\mathcal{N}$ correspond exactly to the cyclic paths – i.e., the *cycles* – of $\mathcal{A}$ [36]. Therefore, the set of attractors of any Boolean recurrent neural network $\mathcal{N}$ can be computed and counted explicitly as follows:

(1) Construct the finite automaton $\mathcal{A}$ corresponding to $\mathcal{N}$;

(2) Compute all simple cycles of $\mathcal{A}$.

Note that a network $\mathcal{N}$ with N reservoir cells yields an automaton $\mathcal{A}$ with at most 2^N states, and an automaton with 2^N states has $\Theta\big(2^N!\big)$ simple cycles.[1] These considerations demonstrate that Boolean neural networks can exhibit an extraordinarily large number of attractors, each of which corresponds to a potential memory.

[1]A complete directed graph with n nodes contains exactly $\sum_{i=1}^{n-1} \binom{n}{n-i+1} (n-i)! \in \Theta(n!)$ simple cycles [45].

Fact 1. *The number of attractors of a BRNN with N cells is in $\Theta\big(2^N!\big)$.*

To efficiently compute the simple cycles of the automaton, we use Johnson's algorithm that finds all simple cycles in a directed graph in time bounded by $O\big((n+e)(c+1)\big)$, where n, e, and $c$are the number of vertices, edges, and simple cycles in the graph, respectively [45]. Obviously, we focus on small Boolean networks (less than 8 nodes) to keep our problem tractable.

These considerations show that the number of attractors in a BRNN fluctuates in response to synaptic changes. As the network modifies its synaptic configuration, the associated automaton is altered, revealing a new cyclic structure that corresponds to the new attractor regime of the network.

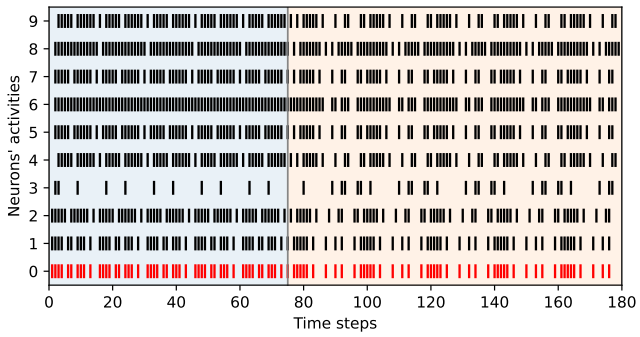

Fig. 2: A Boolean recurrent neural network (the BGT network of Section IV-A) transitioning between two different attractors (blue and orange regions). Two periodic input streams (red spikes of the blue and orange regions) drive the network into two attractor dynamics (blue and orange regions). Formally, an attractor is a set of spiking configurations that recur over time.

C. Synaptic Plasticity

We consider two types of synaptic plasticity mechanisms: (1) a classical *spike-timing-dependent plasticity (STDP)* rule, which adjusts the reservoir synaptic weights locally based on random background activity, and (2) a *global plasticity (GP)* rule, which modifies all the reservoir synaptic weights whenever a specific trigger input pattern is received.

The *spike-timing-dependent plasticity (STDP)* rule strengthens or weakens synapses based on the spiking patterns of the pre- and post-synaptic neurons. This rule is given by:

$$\begin{aligned} \Delta w_{ij}^{\text{res}}(t+1) &= \eta_{ij} \cdot \big(x_j(t) \cdot x_i(t+1) - x_j(t+1) \cdot x_i(t)\big) \\ w_{ij}^{\text{res}}(t+1) &= \text{clip}\left[w_{ij}^{\text{res}}(t) + \Delta w_{ij}^{\text{res}}(t+1)\right] \end{aligned}$$

for any $w_{ij}^{\text{res}} \neq 0$, where η_{ij} is a noisy learning rate given by $\eta_{ij} = \eta \cdot \epsilon$ with $\epsilon \sim \mathcal{U}\left([0.95, 1.05]\right)$ and $\eta \in \mathbb{R}$, and the clip function limits the value of $w_{ij}^{\text{res}}(t+1)$ to the interval $[w_-^{\text{res}}, w_+^{\text{res}}]$, with $w_-^{\text{res}}, w_+^{\text{res}} \in \mathbb{R}$.

Each time a trigger pattern is received, the network unergoes a *global plasticity (GP)* rule that adjusts the reservoir synapses

for the duration of the pattern. This rule is modeled as a simulated annealing process given as follows:

$$\begin{cases} \mathbf{W}^{[\mathbf{res}]}(\mathbf{t}+\mathbf{1}) = \mathbf{W}^{[\mathbf{res}]}(\mathbf{t}) + \boldsymbol{\epsilon}\,, \text{ with } \boldsymbol{\epsilon} \sim \mathcal{U}\left([-\epsilon, +\epsilon]^{M\times N}\right) \\ \Delta E(t+1) = \text{attr}\left(\mathbf{W}^{[\mathbf{res}]}(\mathbf{t})\right) - \text{attr}\left(\tilde{\mathbf{W}}^{[\mathbf{res}]}(\mathbf{t}+\mathbf{1})\right) \\ P_{\text{accept}}(t+1) = \exp\left(-\frac{\Delta E(t+1)}{T_t}\right) \\ T_{t+1} = T_t \cdot \alpha\,, \text{ with } T_0 \in \mathbb{R} \text{ and } 0 < \alpha < 1 \end{cases}$$

where $\mathbf{W}^{[\mathbf{res}]}(\mathbf{t}+\mathbf{1})$ is a noisy update of $\mathbf{W}^{[\mathbf{res}]}(\mathbf{t})$, which is accepted if it results in more attractors than $\mathbf{W}^{[\mathbf{res}]}(\mathbf{t})$ (computed using the attr function) or with a probability $P_{\text{accept}}(t+1)$, which depends on the decreasing temperature T_{t+1}.

IV. Results

A. Experimental setup

We consider Boolean recurrent neural networks with one input cell and $N \in \{5, 6, 7\}$ internal cells. The synaptic weights are drawn from a normal distribution, i.e., $\mathbf{W}^{[in]}, \mathbf{W}^{[res]} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, and the bias vector $\mathbf{b} = \mathbf{0}$. We also consider a Boolean model of the basal-ganglia thalamocortical (BGT) network described in a previous work [37]. The network contains 9 Boolean nodes, each one representing a brain area – the superior colliculus (SC), the thalamus (Thalamus), the thalamic reticular nucleus (NRT), the cerebral cortex (C. Cortex), the striatopallidal and the striatonigral components of the striatum (Str-D1 and Str-D2), the subthalamic nucleus (STN), the external part of the pallidum (GPe), and the output nuclei of the basal ganglia formed by the GABAergic projection neurons of the intermediate part of the pallidum and the substantia nigra pars reticulata (GPi/SNR). Its connectivity is given by the following weight matrix:

Source		**Target**									
#	Node	0	1	2	3	4	5	6	7	8	9
0	IN	·	1	1	·	·	·	·	·	·	·
1	SC	·	·	1	·	·	·	·	·	·	·
2	Thalamus	·	·	·	1	·	1	1	1	1	1
3	NRT	·	·	-1	·	·	·	·	·	·	·
4	GPi/SNr	·	-1	-1	-1	·	·	·	·	·	·
5	STN	·	·	·	·	2	·	2	·	·	2
6	GPe	·	·	·	-½	-½	-½	·	-½	-½	·
7	Str-D2	·	·	·	·	·	·	-1	·	·	·
8	Str-D1	·	·	·	·	-½	·	-½	·	·	·
9	C. Cortex	·	½	1	½	·	½	·	½	½	·

These networks are submitted to background input streams and trigger patterns. For the background activity, we generate a random binary input stream $\bar{u} = \big(u(0), u(1), u(2), \dots\big)$ of length 1000, where each inter-spike interval $\text{ISI}_i \sim \text{Pois}(\lambda = 2)$. For the trigger patterns, we generate a random binary sequence $\bar{p} = \big(p(0), p(1), p(2), \dots\big) \sim \text{Binomial}(n = 50, p = 0.5)$, and then insert it at P non-overlapping random positions inside the input stream $\bar{u}$, where $P \in \{0, 1, 3, 5, 7, 9\}$ (blue regions in Figure 5).

The networks incorporate STDP and GP mechanisms (see Section III-C). For STDP, we consider four possible learning rates $\eta \in [0.1, 0.01, 0.001, 0.0001]$, and for GP, we set the temperature $T_0 = 10$ and the cooling rate $\alpha = 0.995$.

A simulation involves the random generation of a Boolean network $\mathcal{N}$ (including input and reservoir weights) and a background binary input stream $\bar{u}$ interspersed with P trigger patterns. The network processes this input and undergoes STDP or GP, depending on whether it receives background activity or a trigger pattern (blue and non-blue regions in Figure 5). At each time step, these plasticity mechanisms modify the network's synaptic weights, thereby modifying its corresponding automaton, and thus altering its number of attractors, referred to as the *attractor regime* (cf. SectionIII-B). The successive attractor regimes are computed at every time step, and their evolution over time is analyzed (see Figure 5). Each simulation was repeated 10 times with different random seeds. The code is publicly available at the following GitHub repository.

B. Attractor regimes

The maximal numbers of attractors achieved by Boolean networks with $N \in \{5, 6, 7\}$ internal nodes and by the BGT network with 9 internal nodes are reported in Figure 3. The left scales of the successive plots (top to bottom) highlight a drastic increase in the maximal number of attractors, ranging from dozens to millions, as the network size grows from 5 to 9 nodes. These results highlight the capability of Boolean networks to achieve extraordinarily large numbers of attractors, each representing a potential memory, as formally stated in Fact 1. The average (rather than maximal) numbers of attractors attained by the networks were also computed and yielded similar trends (not displayed here), which indicates that high attractor regimes are not isolated events. Moreover, there is a positive correlation between the maximal number of attractors and the number of trigger patterns, which is expected since the GP mechanism is designed to expand the attractor regime. This phenomenon will be further explored in Section IV-C. By contrast, the STDP learning rate η does not exhibit a clear effect on the maximal number of attractors, since STDP is active only during background activity – a condition under which the attractor regime generally tends to decrease (see Section IV-D).

The distribution of attractors' lengths of a regime of the BGT network with 1889 attractors is represented in Figure 4. The attractors are composed of 19.22 states on average.

C. Effect of global plasticity (GP)

Every time the network encounters a trigger pattern, it undergoes the global plasticity (GP) process for the duration of the pattern (see Section III-C). GP is a network-level mechanism designed to amplify the number of attractors, implemented as a simulated annealing process. Specifically, the network adjusts all its reservoir synaptic weights and, with high probability, transitions into successive configurations that increase the number of attractors.

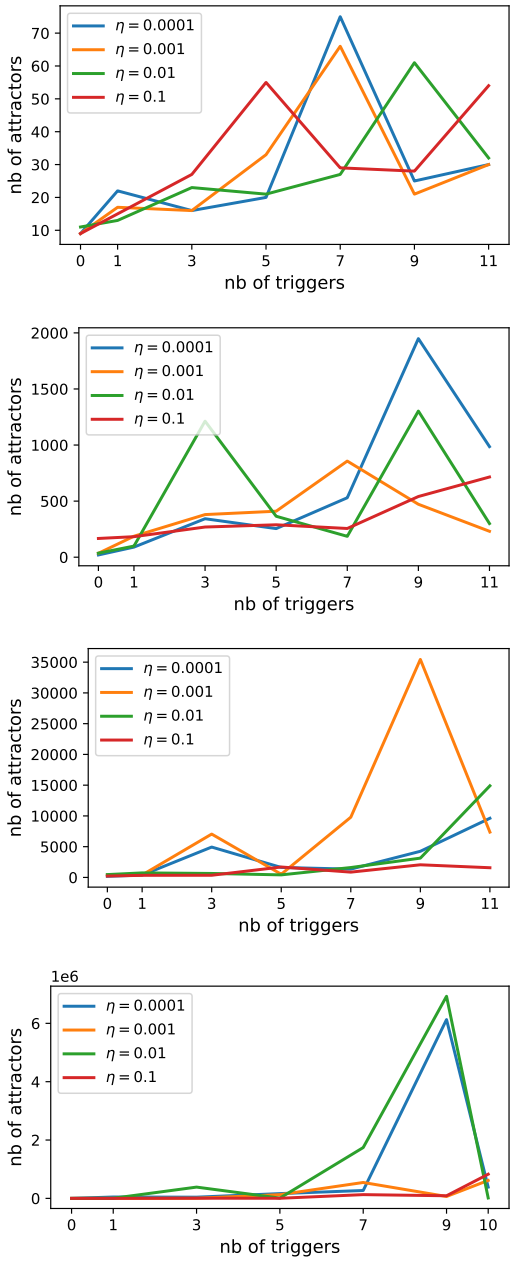

Fig. 3: Maximal number of attractors achieved by Boolean networks with $N \in \{5, 6, 7\}$ internal nodes and by the BGT network (top to bottom), subject to input streams of length 1000 containing $P \in \{1, 3, 5, 7, 9, 10, 11\}$ trigger patterns (horizontal axis).

As expected, the GP mechanism significantly enhances the attractor regime of the network. This effect is clearly illustrated in Figure 5, where the trigger periods (indicated by blue bands) correspond to pronounced increases in the attractor regime. These substantial rises in attractor numbers are generally accompanied by large synaptic changes, quantified as the sum of absolute differences between synaptic weights at successive time steps ($t-1$ and t). The GP mechanism is essential for achieving large attractor regimes, as disabling this process

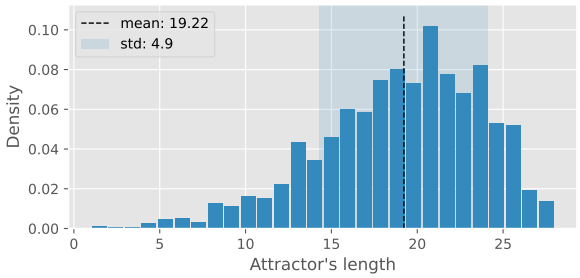

Fig. 4: default

leads to stagnation within very low regimes. In this context, the GP mechanism can be viewed as a form of *learning*.

To quantify this phenomenon on a larger scale, we analyzed all simulations involving networks with 11 trigger patterns (10 for the BGT networks). For each pattern, we measured the difference between the number of attractors two time steps before the pattern and at the end of the pattern. The average of these differences, referred to as the *rise* in attractor regime, is reported in Table I. The results show that the rises increase drastically with network size, reaching an average of 26K attractors for the BGT network.

η	5 nodes	6 nodes	7 nodes	BGT
0.1	4.3 / -3.8	19.1 / -16.7	72.8 / -71.0	10592.4 / -10561.0
0.01	3.1 / -2.9	17.1 / -13.8	244.8 / -228.8	1137.2 / -1121.0
0.001	3.0 / -2.4	14.2 / -12.1	169.4 / -161.2	25831.6 / -25818.1
0.0001	2.7 / -2.3	26.8 / -24.4	157.0 / -96.0	22923.0 / -22907.1

TABLE I: Rises and falls in attractor regimes observed during trigger patterns and following their termination, respectively. Each pair of the form x/y corresponds to the average increase (rise) and decrease (fall) in the number of attractors encountered by the network.

D. Effect of spike-timing-dependent plasticity (STDP)

Between trigger patterns, the network undergoes spike-timing-dependent plasticity (STDP) driven by random background inputs (see Section III-C). STDP is a local mechanism that adjusts synaptic strengths by reinforcing or weakening connections based on the timing of pre- and post-synaptic spikes. This mechanism is commonly associated with learning, as it refines neural circuits to encode patterns of significant input activity.

The STDP mechanism drives the network abruptly into low attractor regimes. This effect is illustrated in Figure 5, where the ends of trigger periods – marked by the reactivation of STDP – coincide with sudden drops in the attractor regime. STDP typically induces small but targeted synaptic changes that can disrupt the attractor structure of the network. We hypothesize that sustaining a high attractor regime incurs a cost, which STDP over random background inputs minimizes. Consequently, STDP in this context can be interpreted as a form of *memory recalibration* allowing the network to adapt and enable new learning.. It is important to note that

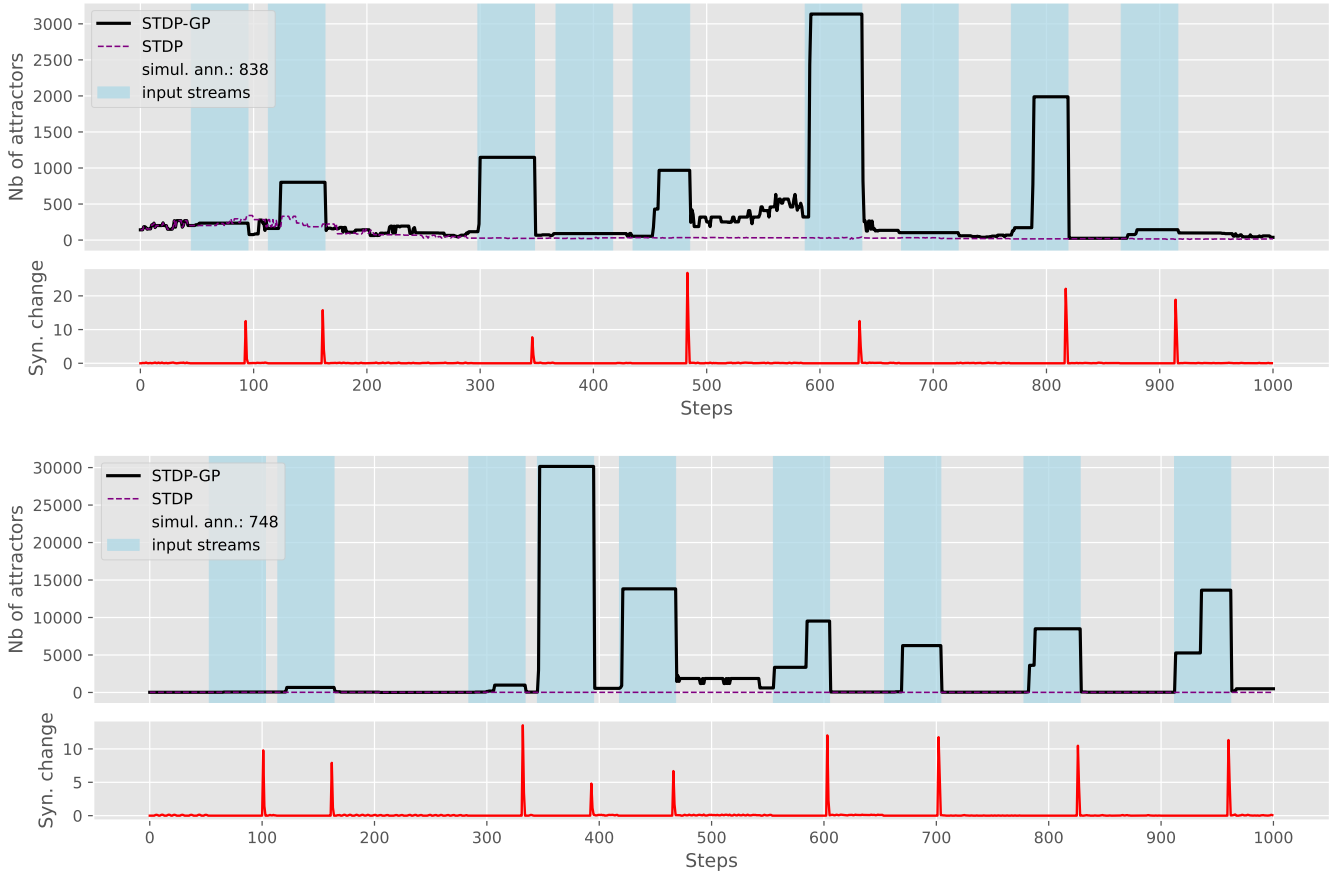

Fig. 5: Simulations of Boolean networks subject to random background activity and trigger patterns (indicated by blue bands). Top: a random network with 7 nodes. Bottom: the BGT network. The networks undergo STDP during background activity and GP during trigger patterns. Synaptic changes induced by these plasticity mechanisms are shown (red trace). At each time step, the number of attractors is computed, and the evolution of the attractor regime is displayed (black trace). For comparison, the attractor regime evolution without GP is shown (dotted trace).

the behavior of STDP differs significantly when applied to crafted, as opposed to random, input streams, as detailed in Section IV-E.

To better understand this phenomenon, we analyzed simulations involving networks with 11 trigger patterns (10 for the BGT networks). For each pattern, we measured the difference between the number of attractors attained during the pattern and the number of attractors two time steps after the pattern ends. The average of these differences, referred to as the *drop* in attractor regime, is reported in Table I. The results reveal that the intensity of the drops mirrors that of the rises, with both increasing significantly as the network size grows.

E. Stability of attractor regimes

Consider a network operating in a high attractor regime. If the synaptic weights of the network were to be frozen, then the network would necessarily maintain its high attractor regime over time. Specifically, the frozen synaptic configuration would correspond to a unique automaton, whose cycles represent the attractor regime, as discussed in Section III-B. However, in biological neural networks, synaptic weights are not abruptly frozen after learning; rather, they are continuously shaped by ongoing network activity. Furthermore, STDP during background activity tends to drastically reduce the attractor regime, as shown in Section IV-D. This raises a natural question: Can neural networks maintain high attractor regimes over time despite the continuous influence of activity-dependent plasticity?

Notably, we demonstrate that a network exposed to specific input streams (rather than random background activity) can be effectively stabilized in high attractor regimes over extended periods, even in the presence of STDP. Specifically, we consider a synaptic configuration of the BGT network associated with a regime R containing 1889 attractors. We then compute a specific input stream that drives the network through attractors of R, as long as no STDP is enabled. Afterward, we re-enable the STDP mechanism and track the evolution of the attractor regime in response to this specific input stream.

The results of this simulation, presented in Figure 6, demonstrate that the network is capable of maintaining its attractor

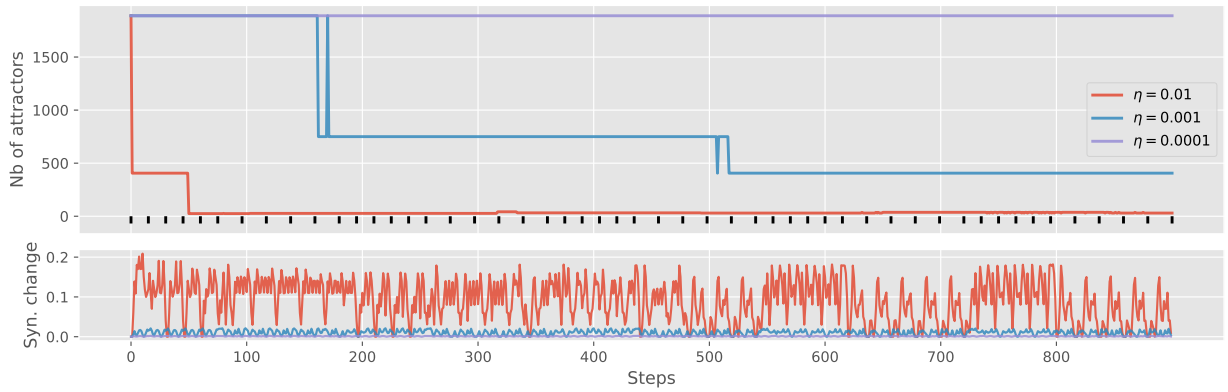

Fig. 6: Evolution of the attractor regime in the BGT network under a specific input stream and different STDP learning rates. (Top) The network begins in a high attractor regime of 1889 attractors and receives crafted input streams (black ticks) that would confine it to this regime in the absence of STDP. The evolution of the attractor regime is shown for three STDP learning rates ($\eta = 0.0001$, $\eta = 0.001$, $\eta = 0.01$). (Bottom) Magnitude of synaptic changes induced by STDP over time.

regime over time, depending on the learning rate η. For a very small learning rate ($\eta = 0.0001$), the network retains its high attractor regime indefinitely. We hypothesize that the synaptic changes induced by this minimal learning rate are too small to alter the dynamical automaton associated with the network, thereby preserving its attractor regime. At an intermediate learning rate ($\eta = 0.001$), the network initially sustains its high attractor regime but gradually transitions to lower regimes, displaying a plateauing effect. In contrast, for a large learning rate ($\eta = 0.01$), the network rapidly collapses into low attractor regimes.

These observations suggest that the magnitude of synaptic changes directly influences the stability of the dynamical automaton underlying the network. Small synaptic changes are associated with stability and preservation of the attractor regime, while large changes lead to its disruption. Overall, this mechanism may be interpreted as a form of *progressive memory loss* in the absence of memory reactivation.

V. Conclusion

This study investigates the dynamics of attractor regimes in Boolean recurrent neural networks, focusing on the role of synaptic plasticity mechanisms – Spike-Timing-Dependent Plasticity (STDP) and Global Plasticity (GP). Our results reveal that Boolean networks are capable of attaining a huge number of attractors, which increases with the network's size.

Spike-Timing-Dependent Plasticity (STDP) has been shown to effectively learn individual patterns in sequence, by locally adjusting synaptic weights based on the precise timing of pre- and post-synaptic spikes. This mechanism allows the network to fine-tune its representation of specific patterns. However, our findings demonstrate that STDP alone is insufficient to drive a substantial expansion of the network's attractor regime. The local, pattern-specific adjustments induced by STDP typically result in either the maintenance or abrupt degradation of the attractor space, rather than its expansion.

In contrast, the expansion of attractor regimes requires global synaptic modifications, implemented by our Global Plasticity (GP) mechanism. By operating at the network level, GP allows the exploration of a broader range of synaptic configurations with the goal of expanding the network's attractor regime. These global modifications facilitate the simultaneous creation of many attractors, suggesting that GP is a crucial mechanism for enlarging the network's attractor landscape. This process can be viewed as a form of learning, where GP creates additional memory space to store more information.

Our results further indicate that the stabilization of attractor regimes requires the network to follow specific input patterns that reinforce the desired attractor states. In contrast, exposure to random or unsupervised activity leads to the massive degradation of memory representations. These considerations underscore the importance of controlled input streams for stabilizing high attractor states and emphasize the sensitive interplay between synaptic plasticity and external input. We interpret the phenomenon of progressive degradation in attractor regimes, which occurs through successive stabilization of regimes in plateaus, as a form of progressive memory loss in the absence of reactivation of learned patterns.

This study is limited to small networks, as the sheer number of attractors in larger networks makes direct computation impractical. A promising direction for future work would involve identifying computable metrics related to the attractor regime, such as algebraic properties of synaptic matrices or other network-level properties. By leveraging these metrics, we could extend the analysis to larger, more biologically-inspired networks, enabling a deeper understanding of attractor dynamics in more complex systems.

Acknowledgment

This study is partially supported by the joint Swiss-French Partenariat Hubert Curien / Germaine de Staël 2024-2025.

REFERENCES

[1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[2] H. Sompolinsky and I. Kanter, "Temporal association in asymmetric neural networks," *Phys. Rev. Lett.*, vol. 57, pp. 2861–2864, Dec 1986.

[3] D. Krotov and H. Sompolinsky, "Dense associative memory for pattern recognition," in *Advances in Neural Information Processing Systems (NeurIPS 2016)*, pp. 1172–1180, 2016.

[4] M. Demircigil, J. Heusel, M. Löwe, S. Upgang, and F. Vermet, "On a model of associative memory with huge storage capacity," *Journal of Statistical Physics*, vol. 168, pp. 288–299, July 2017.

[5] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, V. Greiff, D. Kreil, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter, "Hopfield networks is all you need," *arXiv preprint arXiv:2008.02217*, 2020.

[6] G. E. Hinton and T. J. Sejnowski, "Optimal perceptual inference," *Proceedings of the IEEE conference on Computer Vision*, 1983.

[7] A. Fischer and C. Igel, "Training restricted boltzmann machines: An introduction," *Pattern Recognition*, vol. 47, no. 1, pp. 25–39, 2014.

[8] H. Sompolinsky, A. Crisanti, and H.-J. Sommers, "Chaos in neural networks," *Physical review letters*, vol. 61, no. 3, p. 259, 1988.

[9] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," tech. rep., German National Research Center for Information Technology GMD Technical Report, 2001.

[10] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[11] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical review letters*, vol. 120, no. 2, p. 024102, 2018.

[12] D. Sussillo and L. F. Abbott, "Generating coherent patterns of activity from chaotic neural networks," *Neuron*, vol. 63, no. 4, pp. 544–557, 2009.

[13] R. Ben-Yishai, R. L. Bar-Or, and H. Sompolinsky, "Theory of orientation tuning in visual cortex," *Proceedings of the National Academy of Sciences*, vol. 92, no. 9, pp. 3844–3848, 1995.

[14] A. Samsonovich and B. L. McNaughton, "Path integration and cognitive mapping in a continuous attractor neural network model," *Journal of Neuroscience*, vol. 17, no. 15, pp. 5900–5920, 1997.

[15] R. S. Zemel and M. C. Mozer, "Localist attractor networks," *Neural Computation*, vol. 13, no. 5, pp. 1045–1064, 2001.

[16] H. T. Siegelmann, "Analog-symbolic memory that tracks via reconsolidation," *Physica D: Nonlinear Phenomena*, vol. 237, no. 9, pp. 1207–1214, 2008.

[17] E. T. Rolls, "Attractor networks," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 1, no. 1, pp. 119–134, 2010.

[18] M. Khona and I. R. Fiete, "Attractor and integrator networks in the brain," *Nature Reviews Neuroscience*, vol. 23, pp. 744–766, 2022.

[19] D. O. Hebb, *The organization of behavior : a neuropsychological theory*. John Wiley & Sons Inc., 1949.

[20] V. Braitenberg, "Cell assemblies in the cerebral cortex," in *Theoretical Approaches to Complex Systems* (R. Heim and G. Palm, eds.), (Berlin, Heidelberg), pp. 171–188, Springer, 1978.

[21] G. Palm, *Neural Assemblies: an Alternative Approach to Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 1982.

[22] C. R. Huyck and P. J. Passmore, "A review of cell assemblies," *Biol. Cybern.*, vol. 107, no. 3, pp. 263–288, 2013.

[23] G. Buzsáki, "Neural syntax: cell assemblies, synapsembles, and readers.," *Neuron*, vol. 68, no. 3, pp. 362–385, 2010.

[24] G. Palm, A. Knoblauch, F. Hauser, and A. Schüz, "Cell assemblies in the cerebral cortex," *Biological Cybernetics*, vol. 108, no. 5, pp. 559–572, 2014.

[25] M. Abeles, *Local Cortical Circuits. An Electrophysiological Study*, vol. 6 of *Studies of Brain Function*. Berlin Heidelberg New York: Springer-Verlag, 1982.

[26] M. Abeles, *Corticonics: Neuronal Circuits of the Cerebral Cortex*. Cambridge University Press, 1st ed., 1991.

[27] M. Diesmann, M.-O. Gewaltig, and A. Aertsen, "Stable propagation of synchronous spiking in cortical neural networks," *Nature*, vol. 402, pp. 529–533, 1999.

[28] M. Abeles, "Time is precious," *Science*, vol. 304, no. 5670, pp. 523–524, 2004.

[29] Y. Ikegaya, G. Aaron, R. Cossart, D. Aronov, I. Lampl, D. Ferster, and R. Yuste, "Synfire chains and cortical songs: Temporal modules of cortical activity," *Science*, vol. 304, no. 5670, pp. 559–564, 2004.

[30] J. Hertz and A. Prügel-Bennett, "Learning synfire chains by self-organization," *Network: Computation in Neural Systems*, vol. 7, pp. 357–363, 1996.

[31] J. Hertz and A. Prügel-Bennett, "Learning synfire chains: Turning noise into signal," *Int. J. Neural Syst.*, vol. 7, no. 4, pp. 445–450, 1996.

[32] D. V. Buonomano, "A learning rule for the emergence of stable dynamics and timing in recurrent networks," *Journal of Neurophysiology*, vol. 94, no. 4, pp. 2275–2283, 2005.

[33] R. Hosaka, O. Araki, and T. Ikeguchi, "Stdp provides the substrate for igniting synfire chains by spatiotemporal input patterns," *Neural computation*, vol. 20, no. 2, pp. 415–435, 2008.

[34] P. Zheng and J. Triesch, "Robust development of synfire chains from multiple plasticity mechanisms," *Front. Comput. Neurosci.*, vol. 8, no. 66, 2014.

[35] J. Cabessa and A. Tchaptchet, "Automata complete computation with hodgkin-huxley neural networks composed of synfire rings," *Neural Networks*, vol. 126, pp. 312–334, 2020.

[36] J. Cabessa and A. E. P. Villa, "An attractor-based complexity measurement for boolean recurrent neural networks," *PLoS ONE*, vol. 9, no. 4, pp. e94204+, 2014.

[37] J. Cabessa and A. E. P. Villa, "A memory-based STDP rule for stable attractor dynamics in boolean recurrent neural networks," in *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pp. 1–8, IEEE, 2019.

[38] L. Susman, N. Brenner, and O. Barak, "Stable memory with unstable synapses," *Nature Communications*, vol. 10, p. 4441, Sep 2019.

[39] J. Cabessa and A. E. P. Villa, "Attractor dynamics of a boolean model of a brain circuit controlled by multiple parameters," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, p. 106318, 10 2018.

[40] I. Tsuda, "Toward an interpretation of dynamic neural activity in terms of chaotic dynamical systems," *Behav. Brain Sci.*, vol. 24, no. 5, pp. 793–847, 2001.

[41] K. Inoue, K. Nakajima, and Y. Kuniyoshi, "Designing spontaneous behavioral switching via chaotic itinerancy," *Science Advances*, vol. 6, no. 46, p. eabb3989, 2020.

[42] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysic*, vol. 5, pp. 115–133, 1943.

[43] S. C. Kleene, "Representation of events in nerve nets and finite automata," in *Automata Studies* (C. Shannon and J. McCarthy, eds.), pp. 3–41, Princeton, NJ: Princeton University Press, 1956.

[44] M. L. Minsky, *Computation: finite and infinite machines*. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1967.

[45] D. B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM J. Comput.*, vol. 4, no. 1, pp. 77–84, 1975.