Intrinsic Computation in Nanoscale Arrays:

Spatiotemporal Considerations for the Computational Mechanics of Bistable Nonlinear Elements with Nearest Neighbor Coupling

PAUL RIECHERS

Department of Electrical and Computer Engineering
University of California at Davis,
Davis, CA 95616
pmriechers@ucdavis.edu

This project aims to develop useful theoretical and computational tools for the discovery and analysis of information processing rules that emerge from the nonlinear interaction of simple spatially extended bistable elements with nearest-neighbor coupling. Although I have developed this project with nanoelectronic components in mind, I have tried to create a more general framework to understand how computational mechanics can be applied to simple processes with spatiotemporal extent. Several example systems, including some elementary cellular automata (CA), are discussed for concreteness.

Introduction

This paper summarizes my initial efforts to formulate a systematic approach to discover the intrinsic computational abilities of simple arrangements of interacting bistable elements in a physical nanoelectronic array. Possible components of interest, like capacitively coupled single-electron tunneling junctions or exchange-coupled superparamagnetic nanoparticles would have a very simple physical structure as well as a very simple coupling mechanism. Since the underlying dynamics are also rather simple, the computational power of such a system depends on the rules of information processing that emerge from the nonlinear collaboration among the cells in the array. However, there is currently no straightforward way to engineer such collaboration.

The section on *Continuous Dynamical Computing* in Appendix H of Ref. [2] suggests that computational mechanics is an appropriate tool for a more systematic study of the attainable behavior in such arrays. Specifically, computational mechanics can detect what a system is intrinsically computing. At least for a small functional subset, or *supercell*, of an array, the interpretation of function might be tractable.

After giving a background on a few of the dynamical systems that can be addressed with the spatiotemporal computational-mechanical analysis developed in this project, I will discuss a selection of methods and results. Most of the methods have been an inspired combination of concepts from information theory [11] and computational mechanics [2]. Although some work has successfully used computation mechanics to discover structure in spatially extended systems (see e.g. Ref. [6],[7]), some of the methods seem unnatural since they were adapted from time series analysis, and a clear theoretical foundation is still needed to advance this promising application of computational mechanics.

I have approached the challenge of spatiotemporal generalization in several steps. First, I show how strictly-spatial information should be described, and how we can visualize its evolution in time. Next, I consider the evolution of strictly-temporal information about structure, more as a time series analysis. Finally, I discuss some preliminary thoughts and results regarding the structure of information in a process' evolving spacetime probability distribution. Although I have not yet come up with a completely cohesive framework, I have tried to show why my approach is appropriate for finding structure in spatiotemporal processes, and have tried to discuss many remaining items of inconsistency.

Background of Relevant Dynamical Systems

Although a wide variety of physical mechanisms with rich continuous dynamics can mediate the interaction among components in nanoscale arrays, we can often discretize these systems in meaningful ways to define a finite-state, discrete-time process. Specifically, many systems of interest can be abstracted to a set of bistable elements with nearest-neighbor coupling. Furthermore, if oscillations have a characteristic time scale (and possibly even synchronize or occur in short time windows), the process has a natural discrete time step. In such cases, initially disparate physical processes can be partitioned in a meaningfully similar way. Through this partitioning, these different systems become amenable to a common set of analysis and design tools, while they maintain the intrinsic computational power of their unique continuous dynamic.

There are many examples of systems, from theoretical to physical to electrical, that fall under the proposed common framework. Simple and well-studied systems with spatially extended bistable elements include Cellular Automata (CA) and Ising spin lattices. Standard CNN[1] allows for more complicated neighboring interactions, but is still implemented through uniform coupling in the sense that each cell uses the same template that defines the rule for interaction [3]. A truly nanoscale array could use Coulomb blockade to create bistable single-electron tunneling phase states in small capacitively coupled electrical junctions, as in TPL-CNN. Alternatively, a gated RKKY interaction is being investigated to couple bistable macrospins in an array of superparamagnetic nanoparticles.

Elementary CA consists of a uniform spatial arrangement of indistinguishable automata all following the same rule carried out in discrete time steps. Despite the system's simple description, several rules yield beautifully complex spatiotemporal patterns, and are even capable of universal computation [4].

While CA was not created to model a particular physical phenomenon, the Ising spin lattice was originally presented as a simple model for a uniform spatial arrangement of electron spins that can either align or anti-align with an external magnetic field at finite temperature. Besides the energetic motivation to align with the external field, the antisymmetric requirement of overlapping fermion wavefunctions couples neighboring electrons with an exchange energy term. Although highly simplified, this model of a spin system is still able to show phase transitions at critical temperatures in two-dimensions.

CNN is spatially discrete but computes continuously in time. It has been used for a variety of practical applications, the most common of which is dynamic image processing, and perhaps one of the more unexpected is elementary particle detection.

The quantum mechanical processes of single-electron tunneling and RKKY exchange interaction are both imagined to be implemented in arrays that are nearly spatially discrete. Although the interactions are continuous in time, tunneling events and spin flips can generally be considered discrete time events, since the time scale of the transition is so much shorter than the time between events.

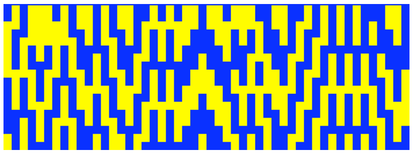
As the following one-dimensional TPL-CNN spacetime diagrams suggest, one-

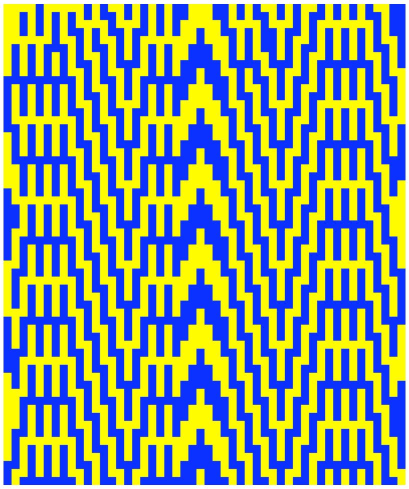
[1] CNN is an evolving acronym that has at different times stood for Cellular Neural Networks, Cellular Nonlinear Networks, and Cellular Nanoscale Networks. Despite the most recent name, commercially available CNN processors are implemented through an elaborate CMOS structure that is far from nanoscale implementation.

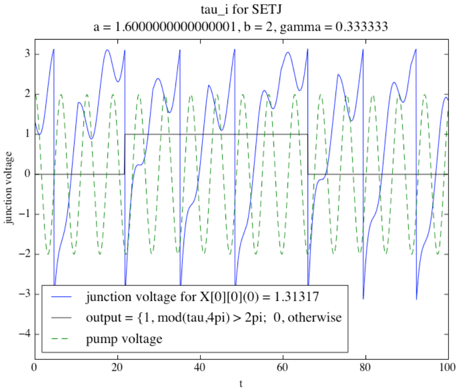
dimensional single-electron tunneling is capable of producing behavior that could be captured by fairly simple rules:

Figure 1

1-D TPL-CNN spacetime diagram with uniform bistable bias, uniform coupling, and random initial conditions. The horizontal axis represents the spatial dimension. Time increases as you go down the diagram. The first row, below, is the first time step. On the right, we see that the one-dimensional array settles down to a periodic 'decision' representing a computation on the initial conditions.






tau_i for SETJ
a = 1.6000000000000001, b = 2, gamma = 0.333333
junction voltage
t
3
2
1
0
−1
−2
−3
−4
0
20
40
60
80
100
junction voltage for X[0][0](0) = 1.31317
output = {1, mod(tau,4pi) > 2pi; 0, otherwise
pump voltage

The above time series shows the actual dynamics of the tunneling phase in the first cell of the array, and shows how the period of the tunneling event determines the binary output.

The first few rows of the spacetime diagram show the transient information processing on a vector of random initial conditions, performed by a row of single-electron tunneling junctions with uniform capacitive coupling and a uniform bias voltage distribution. The uniform capacitive coupling and uniform bias distribution assure that each of the tunneling junctions implements the same intrinsic symmetric rule[2]. Additionally, the uniform voltage bias has been chosen to keep each cell in the bistable mode. Figure 2 shows another example of a TPL CNN spacetime diagram, this time using a graded coupling scheme to implement an asymmetric rule:

[2] Actually, without circular boundary conditions this is not quite accurate. The boundaries have only one nearest neighbor, instead of the normal two, causing slightly different dynamics or rules. This difference propagates through the one-dimensional array like a reflection in a cavity. If we wanted to get a pure rule, we would need an infinitely long array, or an array with circular boundary conditions. However, simulations show that spacetime diagrams from longer arrays under the same voltage bias display similar patterns to that seen in Figure 1. The different behavior at these boundaries is the first hint that boundaries could be useful to enrich the type of information processing from a simple process like single-electron tunneling.

1-D bistable TPL-CNN with graded coupling-capacitance. Unlike the previous spacetime diagrams with uniform coupling, we can see both sides of branching spacetime trees carried to the left over time in this diagram. The first panel shows the initial transient behavior from random initial conditions. Subsequent panels represent chunks of time further in the future.

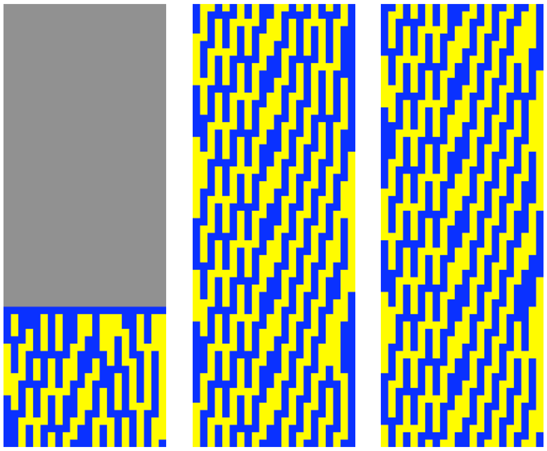


Figure 3 shows an example of exotic boundary conditions—a double-chamber supercell with a single-slit coupling—significantly contributing to the function of the two-dimensional array. This figure exemplifies the importance of boundary conditions in the information processing in a physical array. Boundary conditions can actually *perform* a computation, so we cannot always ignore them, or assume circular boundary conditions as is usually done for CA analysis. I will be guilty of using circular boundary conditions for CA later in this paper, but I will also make some comments about possible ways to account for boundary conditions in computational mechanics analysis, where there presence is important.

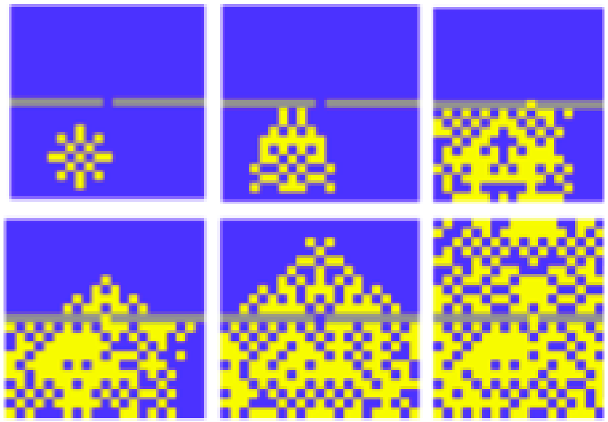


Figure 3

Evolution of phase states in a two-chamber system coupled by a single slit. Gray represents cells that have been removed from the two-dimensional array. Blue and yellow represent binary phase states. The array was initially excited by a small square of nine cells in the left half of the bottom chamber that started in a different phase state than the rest of the array. Besides being a clear example of the importance of boundary conditions, the setup could be interpreted as a single slit experiment for the propagation and interference of nonlinear waves.

Methods and Results

Spatial Block Entropies

The block entropy for a length-L word is given by:

$$H(L) \;=\; H(P(s^L)) \;=\; -\sum_{s^L \in A} P(s^L)\log_2 P(s^L) \qquad , \qquad \text{where} \qquad s^L = \bigcup\nolimits_{i=1}^{L} s_i \,.$$

However, the correct interpretation of the above equation for spatially extended system is not immediately obvious, and in fact requires careful consideration. Most fundamentally, what is L? What constitutes a word in spatially extended systems? What is the size of the corresponding alphabet? What if time is involved? Conversely, what if we are interested in purely spatial properties and want to ignore time?

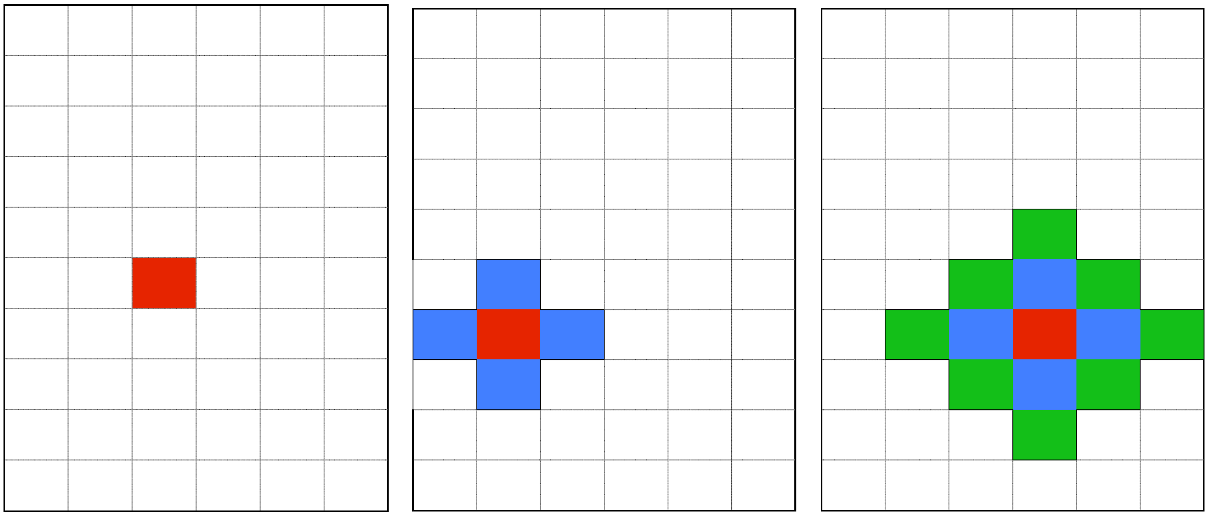
I propose that a word of length L should represent a measurement from the system that includes L layers of causal information. In different spacetime dimensions the shape of the onion of influence might change, but the following formalism, experimentation, and results represents the consequences of adhering to this assumption.

The Strictly Spatial Onion

When first considering the above questions, it is important to note that different geometric spaces induce different types of alphabets. Specifically, the type of coupling between neighbors determines the geometry of the causal onion, and thus the appropriate norm to map from the physical lattice to L-space. For example, as shown in Figure 4, cells in a two dimensional array with four nearest neighbors induces a taxicab geometry, which uses the L^1 norm,

$\| x \|_1 \equiv \sum_{i=1}^{D} | x_i |$ (which is $| x_{L_x} | + | x_{L_y} |$ for the D = 2 strictly spatial case), with discrete lattice points, such that the L from the block entropy becomes $L = 1 + \| x \|_1$. The extra '1' is due to the causal layer of self-influence. Alternatively, we would expect that an inter-component interaction mediated through vacuum would make L-space exactly coincide with real-space. In such cases (for example, communication via electromagnetic radiation through vacuum) our L-space would use the more familiar L^2, or Euclidean, norm. As a final example, an array of cells with eight nearest neighbors, in either two or three dimensions maps to L-space through the maximum, or Chebyshev, norm, L^{∞}, such that $\| x \|_{\infty} \equiv \max\{| x_1 |,\ldots,| x_D |\}$ (which is $\max\{| x_{L_x} |,| x_{L_y} |\}$ for the D = 2, strictly-spatial case). It is interesting to make the analogy that the taxicab geometry relates to a simple cubic lattice, whereas the eight nearest neighbor case relates to a body centered cubic lattice. A D-dimensional lattice with coupling through the face of the D-dimensional cube induces an L^1 norm, while coupling through the corners of the D-cube requires the L^{∞} norm. In one-dimension, all of the L^p norms are equivalent, so it is good that we have worked out the correct framework in higher dimensions.

Figure 4



(a) L = 1 (b) L = 2 (c) L = 3

The above length-L templates show how real-space maps to L-space for a 2-D array of elements (contained by the grid) with four nearest neighbors. Each of these L-templates can be scanned across the entire array to find the corresponding $P(s^L)$ of the block-entropy equation in the spatial-only regime.

Let us now relate our L-space observations back to the interpretation of block entropy:

$$H(L) = H(P(s^L)) = -\sum_{s^L \in A} P(s^L) \log_2 P(s^L) \quad , \quad \text{where} \quad s^L = \bigcup_{i=1}^{L} s_i$$

The strictly spatial sphere of influence for an array with n internal states per cell, D spatial dimensions, and L^1-norm coupling has $s_1 \in \{0_{L=1}, 1_{L=1}\}$, but $s_2 \in \{0_{L=2}, 1_{L=2}, 2_{L=2}, \ldots, (2^4-1)_{L=2}\}$. The subscript on the letters is a reminder that the meaning of each symbol is unique. For example, a '0' in the set A(L=1) is different than the meaning of a '0' in the set A(L=2). In general the size of the alphabet (which is really more of a vocabulary) for length-L words is

$$length\{A(L_{L^1-coupling})\} \quad = \quad \prod_{k=0}^{L-1}\left\{n^{(\delta_{k,0}+2^D k^{(D-1)})}\right\},$$

at least in one and two dimensions, although I believe my formula is more general. The N_C nearest neighbors are included implicitly in this formula since $N_C = 2^D$ for L^1-norm coupling. Similarly, for a D-dimensional array with n internal states per cell and L^∞-norm coupling, $s_1 \in \{0,1\}$, but $s_2 \in \{0,1,2,\ldots,2^8-1\}$, and in general,

$$length\{A(L_{L^\infty-coupling})\} \quad = \quad \prod_{k=0}^{L-1}\left\{n^{(\delta_{k,0}+(2k+1)^D-|2k-1|^D)}\right\},$$

at least in one through three dimensions. The N_C nearest neighbors are included implicitly in this formula since $N_C = 3^D - 1$, for L^∞-norm coupling.

To further our discussion, we will define the *neighborhood population* ξ, at a particular L for any array as the number of cells enclosed in a volume of radius L. For a particular array, we should be able to define some relationship ξ(L), although ξ need not be linear in L. This relationship is further complicated by boundary conditions, which we will consider in more detail in another section. For now, let us assume that the relationship ξ(L) is known. Since there is a finite number of cells, ξ_{max}, in any physical array, we consider L_{max} to be the smallest L such that $\xi(L) = \xi_{max}$:

$$L_{\max} \equiv \min\{L \quad \ni \quad \xi(L) = \xi_{\max}\}$$

In terms of the neighborhood population, we can reformulate the alphabet length more generally as:

$$length\{A(L)\} \quad = \quad \prod_{k=1}^{L}\left\{n^{(\xi(k)-\xi(k-1))}\right\} = \prod_{k=1}^{L}\left\{n^{\xi(k)}n^{-\xi(k-1)}\right\} = n^{\xi(L)}.$$

To really understand the meaning and effect of probabilities in the block entropy, we should compare the number of possible L-words with the actual number of L-words that will be expressed at a given time for any finite (i.e. physically real) array. A truly random set of events with infinitely many words has an entropy equal to the log of the length-L alphabet:

$$H_{ideally-random}(L) = -\sum_{s^L \in A} P(s^L)\log_2 P(s^L) = \log_2(length\{A(L)\}) = \xi(L)\log_2(n)$$

However, a physical system with only $\xi_{\max}$ expressible variables has quantized probabilities: the minimum nonzero probability expressible by the system at any one time is $\Delta P = 1/\xi_{\max}$. Also, for any L, the number of L-words expressed at a given time will equal the number of cells in the array, $\xi_{\max}$. So the closest that a finite physical system comes to a random configuration of L discrete variables is approximately described by the entropy:

$$H_{physically-random}(L) \approx \min\{\log_2(length\{A(L)\}), \log_2(\xi_{\max})\} = \min\{\xi(L)\log_2(n), \quad \log_2(\xi_{\max})\}$$

although this is not quite correct since $\xi_{\max}$ is not in general divisible by *length{A(L)},* so that for small $\xi_{\max}$, this approximation will be poor when *length{A(L)}* $< \xi_{\max}$.

Now, we consider the reciprocal perspective on randomness to show that our quantized probabilities not only limit randomness, but also make non-random processes produce configurations that appear as random as possible when the would-have-been predictable structure has spatial frequency comparable to the lattice size. We see that a configuration of $\xi_{\max}$ cells has a maximum block-entropy, the *saturation entropy*:

$$H_{sat} = \log_2(\xi_{\max})$$

independent of L, for all L such that $\xi(L)\log_2(n) > \log_2(\xi_{\max})$.

Aware of the features of finite probability resolution in a strictly-spatial system, we are now in a position to consider functions of H(L). Although we will revisit the utility of these formulations, the typical entropy rate for finite L, which we will call the *effective entropy rate* at L, can still be defined as

$$h_\mu(L) \equiv H(L) - H(L-1)$$

and the excess entropy for finite L, which we will call the *effective excess entropy* at L, can be defined as

$$\mathbf{E}(L) \equiv \sum_{L'=1}^{L} \{h_\mu(L') - h_\mu(L)\}.$$

However, we can rewrite $\mathbf{E}(L)$ as

$$\mathbf{E}(L) = -Lh_\mu(L) + \sum_{L'=1}^{L}\{h_\mu(L')\} = -Lh_\mu(L) + \sum_{L'=1}^{L}\{H(L') - H(L'-1)\},$$

where $\sum_{L'=1}^{L}\{H(L') - H(L'-1)\} = [H(1) - H(0)] + [H(2) - H(1)] + \ldots + [H(L) - H(L-1)] = H(L)$

So,

$\mathbf{E}(\mathrm{L}) = H(L) - Lh_{\mu}(L)$.

However, for any physical array, the maximum entropy of a spatial configuration, $\mathrm{H}_{\mathrm{sat}}$, (severely?) bounds the *global entropy rate*

$\mathrm{h}_{\mu} \equiv \mathrm{h}_{\mu}(\mathrm{L}_{\max})$

and imposes a limit on the *global excess entropy*

$\mathbf{E} \equiv \mathbf{E}(\mathrm{L}_{\max})$.

We start considering these constraints with the simple *y = mx + b* of the block entropy diagram by rearranging and substituting the three equations above:

$\mathrm{H}(\mathrm{L}_{\max}) = \mathrm{L}_{\max}\, \mathrm{h}_{\mu}(\mathrm{L}_{\max}) + \mathbf{E}(\mathrm{L}_{\max}) = \mathrm{L}_{\max}\, \mathrm{h}_{\mu} + \mathbf{E} \le \mathrm{H}_{\mathrm{sat}}$

Since $\mathbf{E}$ and h_{μ} are assumed to be nonnegative, we have the weaker constraints that

$\mathbf{E} \le H_{sat} = \log_2(\xi_{\max})$

and

$$h_{\mu} \le \frac{H_{sat}}{L_{\max}} = \frac{\log_2(\xi_{\max})}{L_{\max}(\xi_{\max})}$$

The constraint on entropy rate here is very interesting, since the dimension and coupling of the physical array primarily determines the relationship between $\mathrm{L}_{\max}$ and $\xi_{\max}$. This very general result is thus roughly a statement about how the entropy rate of physical configurations scale with physical dimension. The result is especially insightful if we consider the limit of the number of array elements, $\xi_{\max} \to \infty$.

For one-dimensional systems, $\mathrm{L}_{\max}(\xi_{\max}) \propto \xi_{\max}$, so

$$\lim_{\xi_{\max}\to\infty}\left(h_{\mu,D=1}\right) \le \lim_{\xi_{\max}\to\infty}\left\{\frac{\log_2(\xi_{\max})}{L_{\max}(\xi_{\max})}\right\} = \lim_{\xi_{\max}\to\infty}\left\{\frac{\log_2(\xi_{\max})}{c_1\xi_{\max}+c_2}\right\} = 0$$

So, for a one-dimensional system, h_{μ} quickly tends to zero with increasing array size. This result might be related to the nonexistence of phase transitions in most one-dimensional systems.

In contrast, $\mathrm{L}_{\max}(\xi_{\max}) \propto \xi_{\max}^{1/2}$ for two-dimensional systems, and

$\mathrm{L}_{\max}(\xi_{\max}) \propto \xi_{\max}^{1/\mathrm{D}}$

for a D-dimensional spatial configuration. For example, for a two-dimensional m×m array, $\xi_{\max}$ = m×m = m^2, but $L_{\max}$ = 2m for L^1 coupling while $L_{\max}$ = m or L^∞ coupling. Either way, $\xi_{\max} \propto L_{\max}^2 \rightarrow L_{\max}(\xi_{\max}) \propto \xi_{\max}^{½}$ for the two-dimensional array. More generally, for an array of D physical dimensions,

$$\lim_{\xi_{\max}\to\infty}\left(h_\mu\right) \leq \lim_{\xi_{\max}\to\infty}\left\{\frac{\log_2(\xi_{\max})}{L_{\max}(\xi_{\max})}\right\} = \lim_{\xi_{\max}\to\infty}\left\{\frac{\log_2(\xi_{\max})}{c_1\xi_{\max}{}^{1/D}+c_2}\right\} = 0$$

so entropy rates of strictly-spatial configurations tend to zero as $\xi_{\max} \rightarrow \infty$ in all dimensions, although the scaling rate of the inequality differ.

In contention with these results, one might argue that we must average over the spatial configuration of the array at different times to attain the true probability distribution. It seems that once we introduce probability measurements in time, we can beat our original probability resolution of $\Delta P = 1/\xi$. After some integer number of discrete time measurements has elapsed, the probability resolution for our configuration is now $\Delta P = 1/\xi_{\max}\Delta t$. However, in a time dependent process, there has also been a far greater increase in time dependent probabilities, so we are really just embedding our problems in a new level of sophistication. Additionally, we are finding a probability distribution to a different question (although it is a question that we are likely interested in). Yet, if we continually recreate the initial conditions of interest, we can begin to assimilate the correct time-dependent probability distribution, similar to the method of spike-triggered averaging in neuroscience [10]. The point is that for a deterministic system, the current configuration of the array is the correct probability distribution for L-blocks in the configuration, and the corresponding strictly-spatial block entropies lead to an entropy rate of the configuration that vanishes with increasing array size. In fact, the only way to have a finite entropy rate is to have a finite array.

While the global entropy measures may have limited utility, the effective entropy measures for a configuration of states in a physical array could actually be extremely insightful. Persistent effective entropy rates and effective excess entropies can describe the various types of patterns and computations interacting at different length scales.

We will see just a glimpse of the utility of viewing the time evolution of effective entropy rates and effective excess entropies for strictly-spatial configurations in the following section.

Time Evolution of the Strictly Spatial Onion

In this section, we will investigate how a one-dimensional elementary cellular automata (CA) evolves in time. Specifically, we will focus our examples on universal rule 110. However, we will also mention a few other example systems for contrast.

For a one-dimensional system of bistable elements with nearest neighbor coupling, the alphabet size at a given integer L > 0 is

$$length\{A(L)\}_{bistable,NN-coupling} = 2\prod_{k=1}^{L-1}\left\{2^{2^{1}k^{0}}\right\} = 2\left(2^{2(L-1)}\right) = 2^{2L-1}$$

from the above definition. Because the array is one-dimensional (with circular boundary conditions), $2L_{max} - 1 = \xi_{max}$, so

$$L_{max} = (\xi_{max} + 1) / 2$$

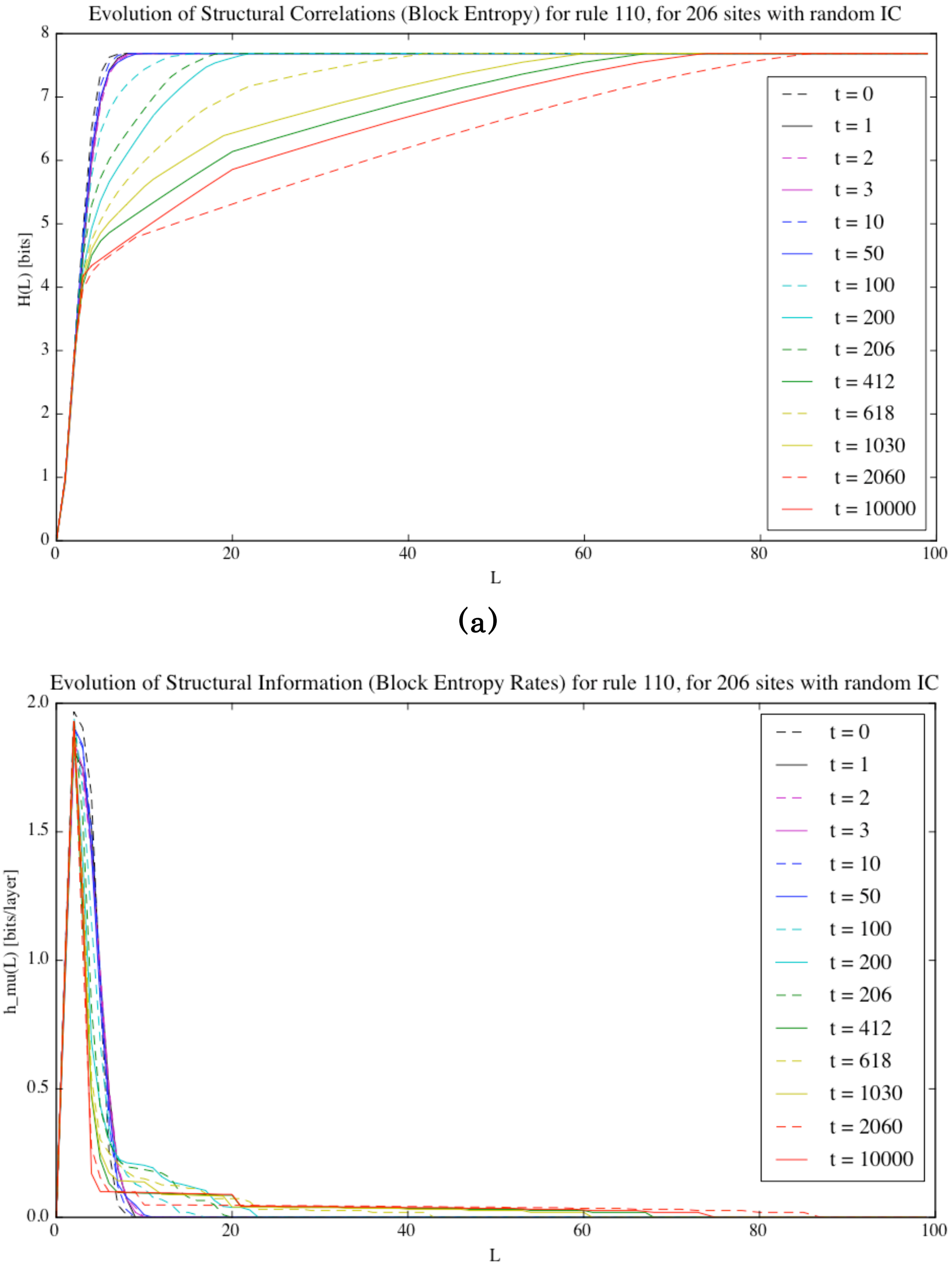
The maximum global entropy rate for the configuration at any one time is thus bounded by

$$h_{\mu} \leq \frac{\log_2(\xi_{max})}{L_{max}(\xi_{max})} = \frac{2\log_2(\xi_{max})}{(\xi_{max}+1)}$$

while the global excess entropy is bounded by

$$\mathbf{E} \leq \log_2(\xi_{max}).$$

These global limitations will be apparent in the examples, but our interest will most likely yield to the time evolution of the effective entropy rates and effective excess entropy as the underlying dynamics generating the spatial configuration give different structure at different scales.

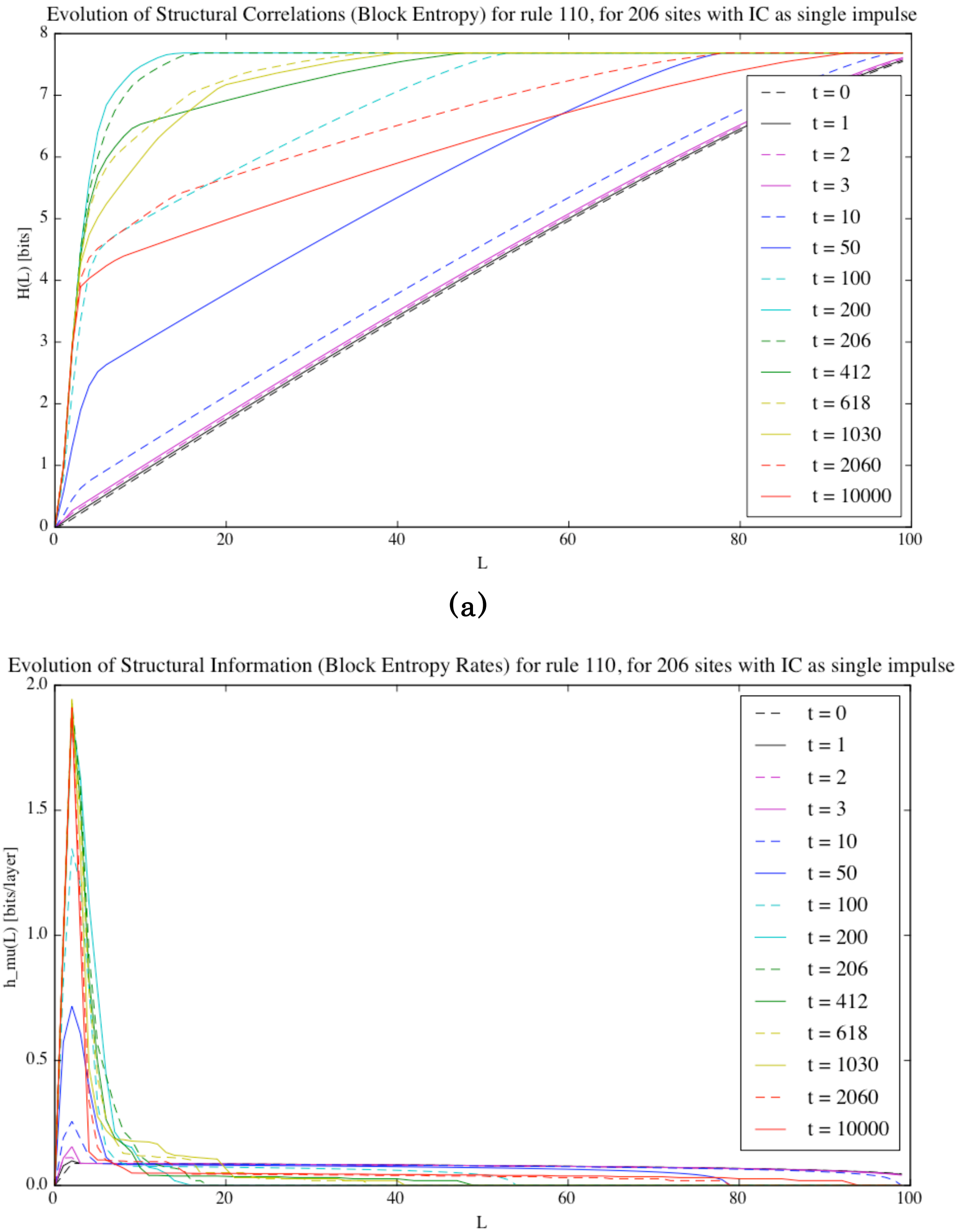


(a)

(b)

Figure 5

Figure 5 shows the evolution of strictly-spatial block entropies and entropy rates from random initial conditions for elementary CA rule 110.



(a)

(b)

Figure 6

Figure 6 shows the evolution of strictly-spatial block entropies and entropy rates from random initial conditions for elementary CA rule 110. The saturation L decreases and again increasing after t ≈ nSites = 206. We are really just watching the process become disordered (until about t =50) as apparent in Figure 7, and then finally settling down to the H(L) plot characteristic of CA 110.

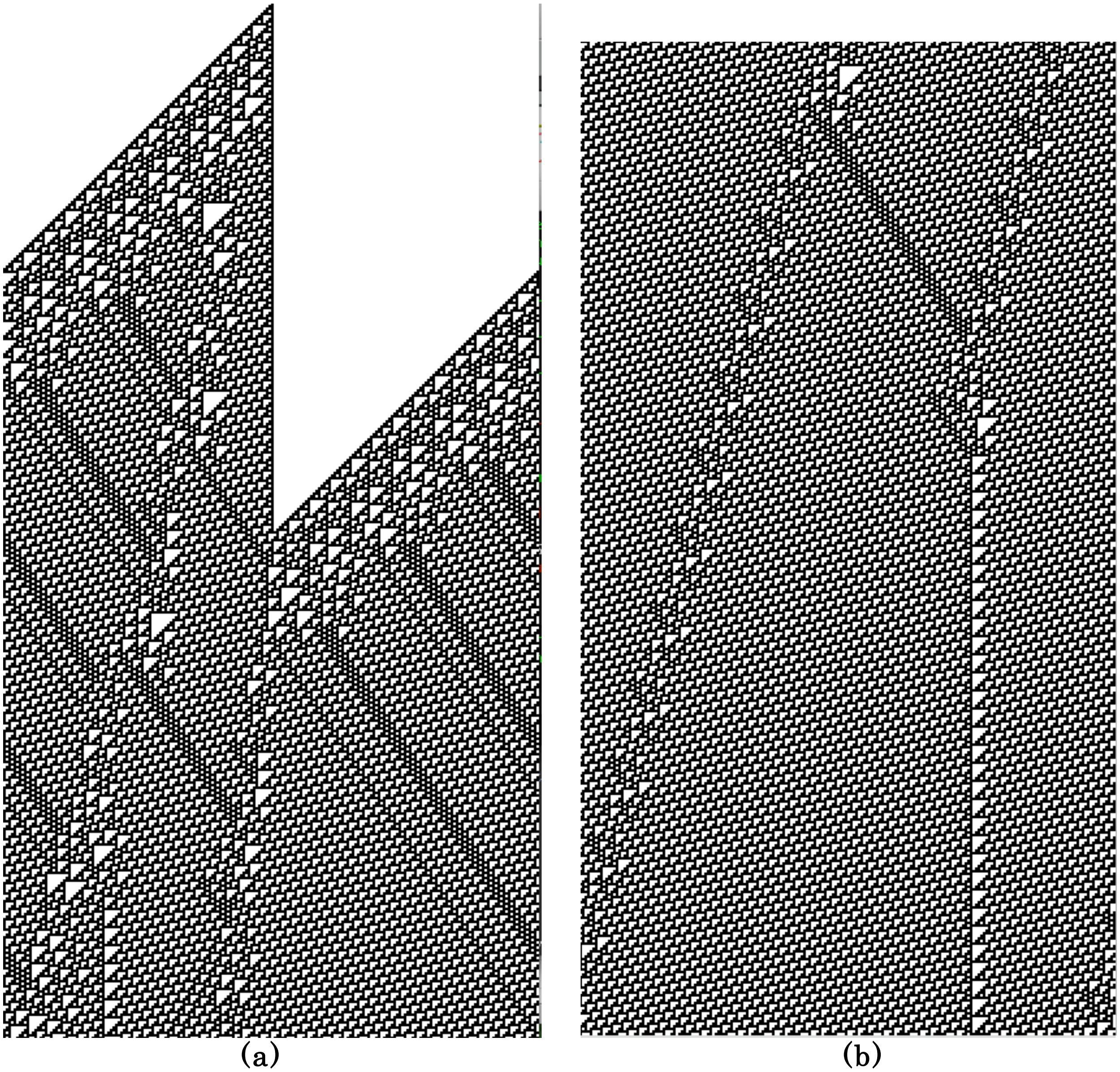
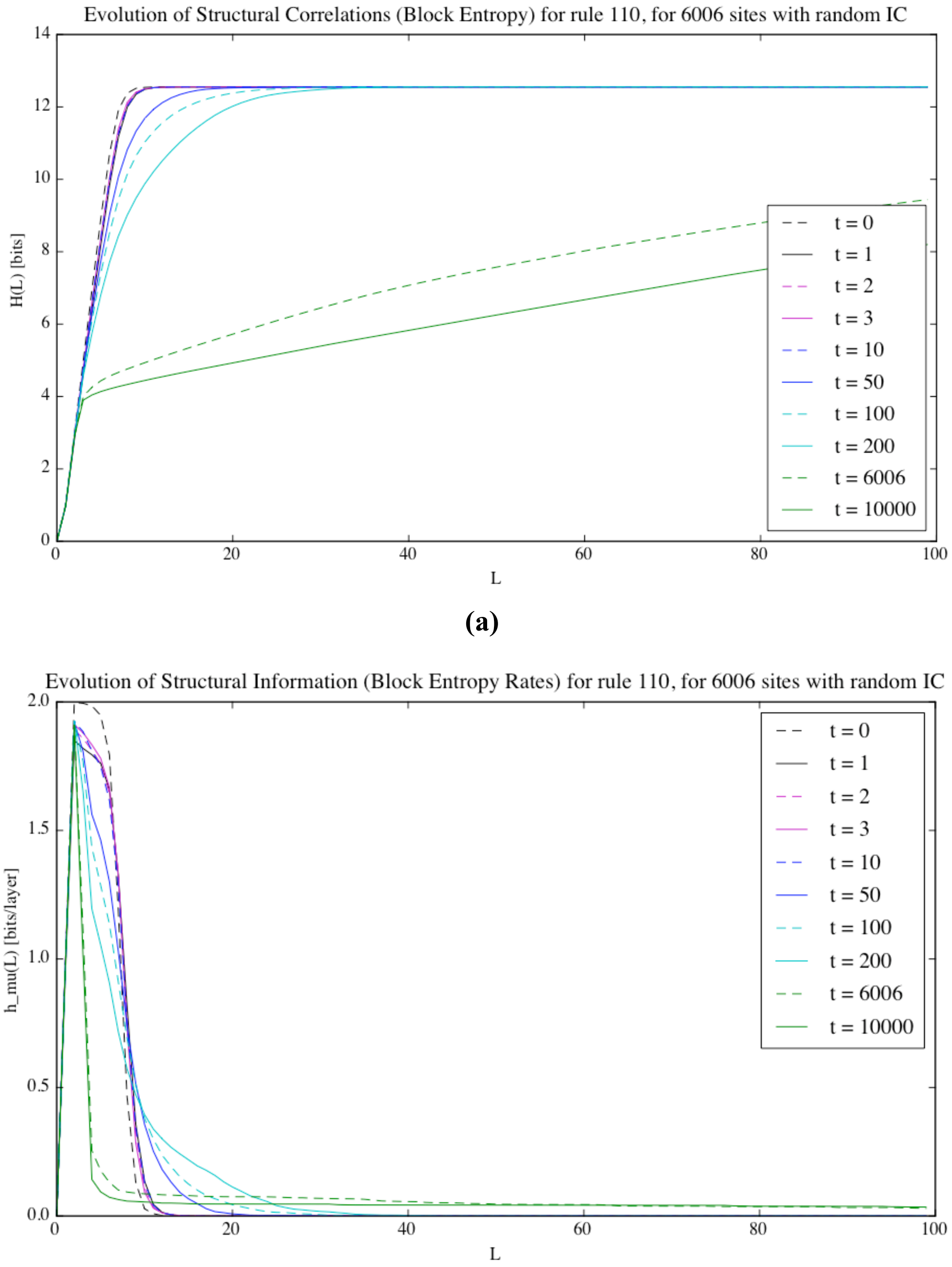


Figure 7

Above: A 206-site Rule 110 spacetime excerpts for (a) the beginning of the evolution and then (b) up to t = 2000, corresponding to Figures 5 and 6

How do we know that this curve (Figure 6, at ~t = 10000) is characteristic of 110? Figure 5 and Figure 7 can be used to argue heuristically. The first shows that we approach approximately the same line from random ICs. The second shows that we approach the same line even for a much larger (and thus more trustworthy) number of cells. The similarity of the steady state H(L) curves in Figures 5, 6, and 7 (at least in the range of L ≤ 40) suggests that this H(L) line is really a distinct feature of Rule 110.


Evolution of Structural Correlations (Block Entropy) for rule 110, for 6006 sites with random IC

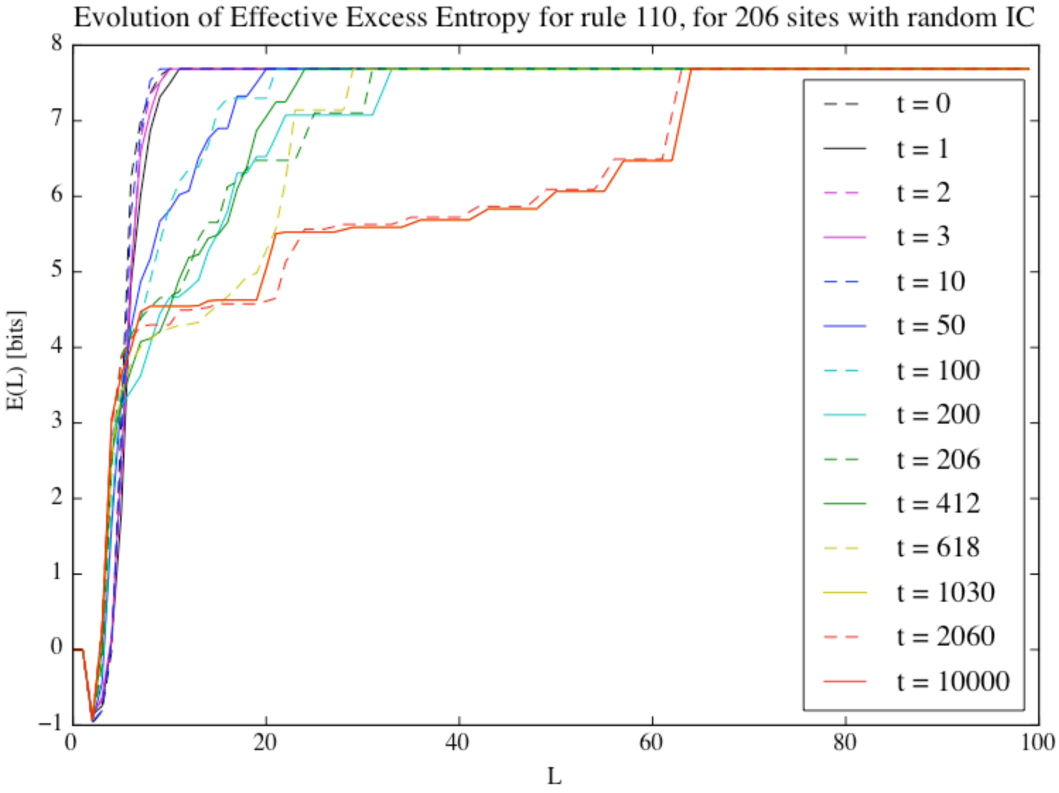
(a)

Evolution of Structural Information (Block Entropy Rates) for rule 110, for 6006 sites with random IC

(b)

Figure 7

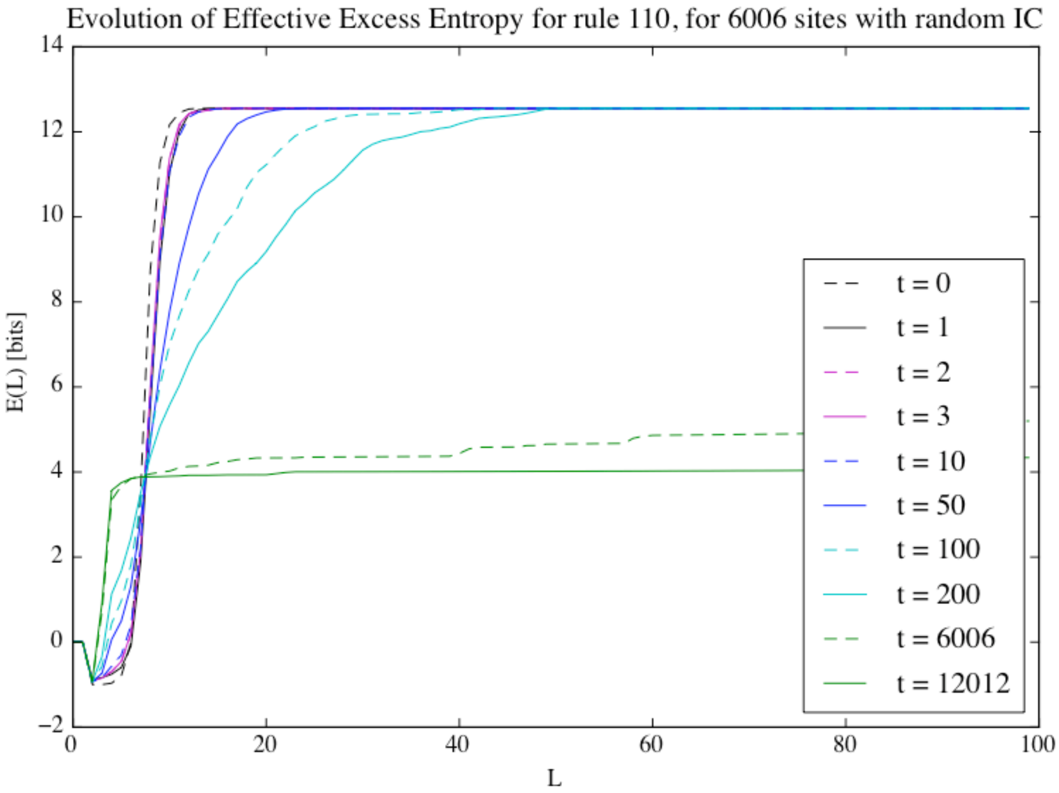
Random initial conditions in Figure 7 give h_μ = 0 bits/layer and **E** = $\log_2(6006)$ = 12.55 bits. h_μ is related to the unpredictability, or information gain, in the system: apparently the entire configuration could be reconstructed from the probabilities given up to L_{sat}. **E** = $\log_2(6006)$ bits suggests that the configuration is fully utilizing its available memory: essentially it is as random as the configuration can allow. Is this to say that the random configuration is not algorithmically compressible? After a few thousand timesteps, the internal process (elementary CA rule 110) has significantly changed the computational-mechanical theoretic properties of the configuration. It appears that $\mathbf{E}(L)$ and $h_\mu(L)$, the L-dependent effective excess entropy and L-dependent effective entropy rate, can describe much more of the process than the simple scalars $\mathbf{E}$ and h_μ. The utility of the effective entropy measures is very apparent in Figure 8.



(a)

Figure 8

Figure 8 shows the effective excess entropy, an the how different length scales will have different apparent memory. Comparing (a) and (b) suggests that the effective excess entropy depends on the number of lattice sites more than the other entropy measures.



(b)

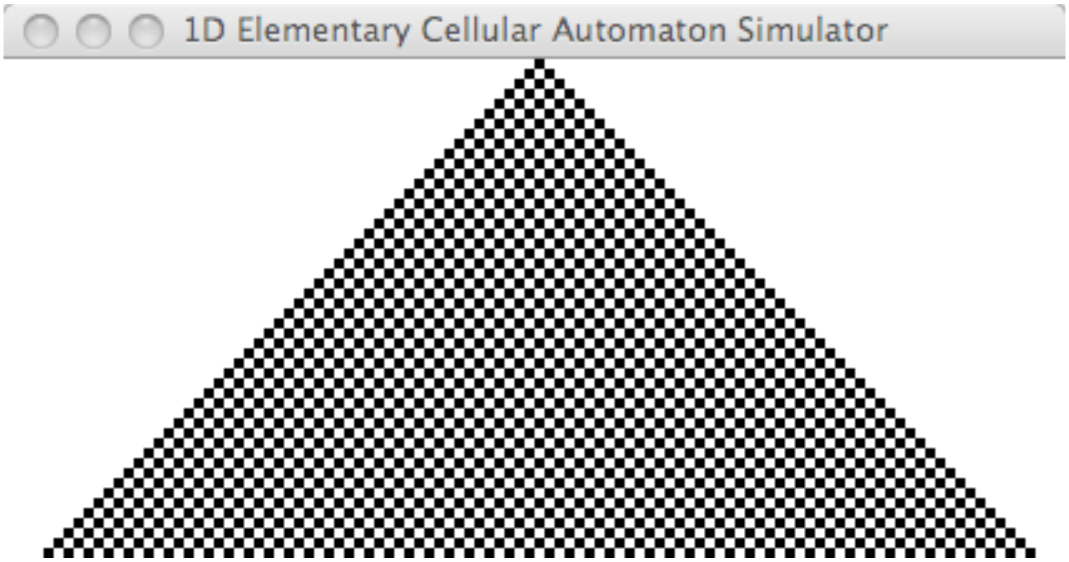
(the last time in 8(b) should read 't = 10000')

Figure 8

We note that for L between 1 and 2, there is a negative effective excess entropy, suggesting some sort of negative memory. Is this meaningful? Is this a true memory sink or just an artifact of an inconsistent alphabet between L=1 and L=2? I believe that this is a feature of the geometry. It reflects the ratio of surface area cells to cells in the volume enclosed by this surface layer. In two dimensions, there will be a different characteristic curve. This could be good: The entropy curves could tell us something about the geometry of our configuration if this is something that we would like to infer. However, if we wanted to avoid this feature, we could plot H(L) vs. ξ(L) to obtain the entropy as a function of cells in the volume considered. Similarly, one might like to plot $\mathbf{E}(L) = H(L) - h_\mu(L)\ L$ as $\mathbf{E}(\xi(L)) = H(\xi(L)) - h_\mu(\xi(L))\ \xi(L)$ to see excess entropy as a function of neighborhood volume.
Meanwhile, neighborhoods from about L=4 up to L=100 all have a persistent apparent memory of between 4 and 5 bits, and a nearly constant entopy rate. We are starting to see that there are characteristic lengths that processes manifest in their physical medium.

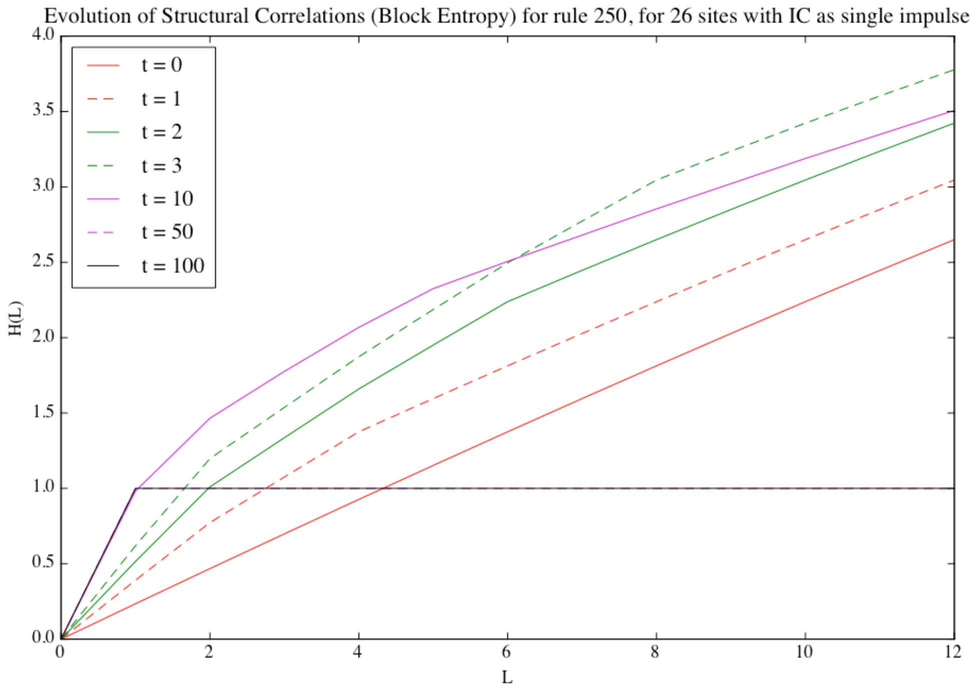
Curves that saturate at different L have different translational symmetries of their probable structure. The L at which a curve reaches $h_\mu(L) = 0$, the *saturation L*, L_{sat}, is roughly the maximum radius of coherent structure. Usually, the curve will be saturating H_{sat}, but not necessarily. As we see in Figure 9, the characteristic curve for Rule 250 saturates at H = 1 bit.

The *friendly neighborhood*, $\xi(L_{sat})$ is the last characteristic length scale that is relevant to the overall structure. Strictly, however, $\xi(L_{sat})$ is only the length (volume = length for 1-D) at which all block-translations are unique, and so we need some more sophisticated analysis to really understand the reason for saturation and the rates of information gain at L less than L_{sat}. Reconstructing an epsilon machine might be an appropriate step to take here.



(a)
Figure 9

Elementary CA Rule 250 (spacetime diagram above) propagates a checkerboard configuration from a single impulse at t_0.



(b)
Figure 9

In Figure 9 (a) and (b), we see that the alternating 010101… sequence, generated by elementary CA rule 250, has H(L) = 1 bit for all L after the steady state time has been reached, so long as the circular boundaries connect a 0 to a 1 (even # of lattice sites).

Because of the variable alphabet size, H(L) is generally a more interesting function of L than we have seen for simple binary alphabets like H(ξ). The strongest constraints on this function is that H(L) must be monotonically increasing. However, H(L) need not be concave down.

We will now take a step back from our long look at the time evolution of the strictly-spatial causal onion, and will consider inferring the information content of a physical array of bistable nearest neighbor elements through a time series of measurements.

The Temporal Onion:

How can we determine the correct general epsilon machine for a bistable element with nearest neighbor coupling? We can start by considering the simple time series of the internal state of the element. If there are two recurrent states, we might expect a three state epsilon machine, with one transient state representing initial ignorance of internal state. We might also expect that measuring a new internal state gives a transition probability. So perhaps we have three states and six transition edges, as shown in Figure 10, below:

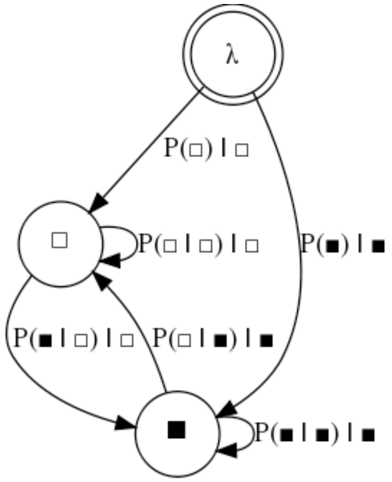


Figure 10

However, given a particular system from the class of systems under consideration, the Hamiltonian (or rule as in the case of CA) nearly explicitly gives us the probability of an element's next state, given the previous state of the element and its nearest neighbors. We thus need to include this information so we are not averaging over relevant information, making the process appear more random than it really is. We can expand the transition probabilities shown in the above figure to include more of the information actually available to the system. For the general epsilon machine for a bistable element with nearest neighbor coupling, we expand transition probabilities to include the effects of nearest neighbors as follows:

$$P(S^n_{t+1} = s \mid S^n_t = s') = \left\langle P(S^n_{t+1} = s \mid S^{n-1} S^n S^{n+1}{}_t = s_j s' s_k) \right\rangle_{s_j, s_k}$$

$$= \sum_{s_j \in A} \sum_{s_k \in A} \left\{ P(S^{n-1}_t = s_j, S^{n+1}_t = s_k) \; * \; P(S^n_{t+1} = s \mid S^{n-1} S^n S^{n+1}{}_t = s_j s' s_k) \right\}$$

where n is the lattice site of the element in question, t is the current discrete time, and A is the set of all symbols in the alphabet. For a single bistable element, this alphabet has only two symbols. Therefore, $P(S^n_{t+1} = s \mid S^n_t = s')$ has been split into four separate pathways, each potentially with unique dynamics and transition probabilities. Distinguishing among these pathways can have consequences for the information theoretic and computational mechanics theoretic quantities of the machine, reflecting an enriched understanding of the properties of the system.

For completeness, we need to also consider the generalization of the transient transition probabilities. The following derivation shows that there is a simple generalization:

$$\begin{aligned} P(S_t^n = s') &= \left\langle P(S_t^n = s' \mid S^{n-1}S^{n+1}{}_t = s_j s_k)\right\rangle_{s_j,s_k} \\ &= \sum_{s_j\in A}\sum_{s_k\in A}\left\{P(S_t^{n-1} = s_j, S_t^{n+1} = s_k) \;*\; P(S_t^n = s' \mid S^{n-1}S^{n+1}{}_t = s_j s_k)\right\} \\ &= \sum_{s_j\in A}\sum_{s_k\in A}\left\{P(S^{n-1}S^nS^{n+1}{}_t = s_j s' s_k)\right\} \end{aligned}$$

where we have used Bayes' Theorem in the last step.

These results can be summarized with the following visualization of a rather general machine for a bistable *element* with nearest neighbor coupling(note that this is *not* the same as the machine for a *network* of bistable nearest neighbor elements):

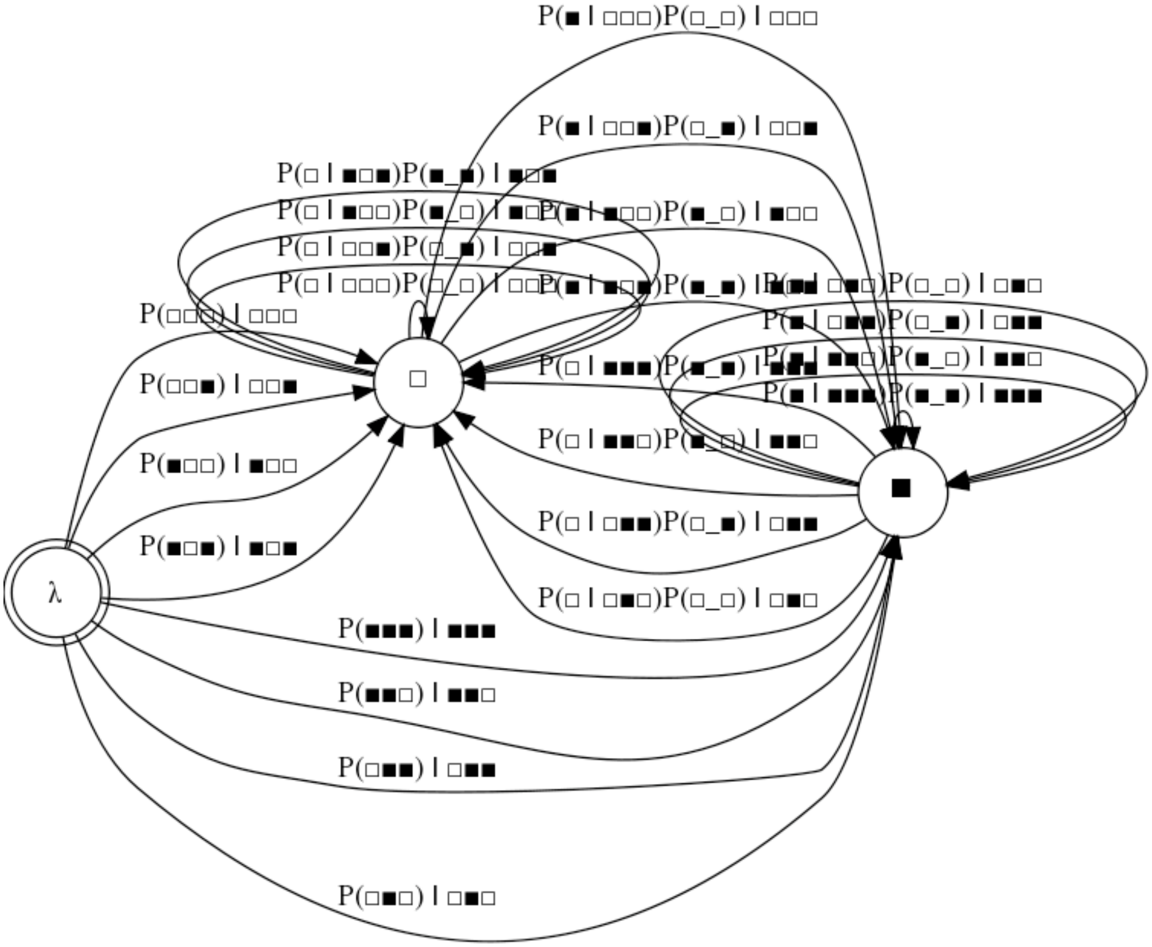


Figure 11

The above black and white squares represent the two bistable states, the time index is removed only to reduce clutter (so the transition probabilities are still, in general, *non-stationary*), and the spatial index is implied by the relative positions of the squares.

For many systems, the fundamental rules of interaction do not change in time, so often $P(S^n_{t+1} = s \mid S^{n-1}S^n\, S^{n+1}{}_t = s_j s' s_k)$ is stationary, although it need not be in general. As an example where the rules could change, a gated RKKY interaction among nearest-neighbor macrospins of superparamagnetic nanoparticles could have the rules changed through a modulation of gate voltage. For most of the example systems described in this paper, $P(S^n_{t+1} = s \mid S^{n-1}S^n\, S^{n+1}{}_t = s_j s' s_k)$ is stationary.

Unlike $P(S^n_{t+1} = s \mid S^{n-1}S^n\, S^{n+1}{}_t = s_j s' s_k)$, $P(S^{n-1}_t = s_j, S^{n+1}_t = s_k)$ will almost always be non-stationary. There are several perspectives to take here. If we are at some time less than t, and we are projecting into the future, then this probability depends non-trivially on each subsequent measurement. However, if we are assuming that we are at some time greater or equal to t already, then $P(S^{n-1}_t = s_j, S^{n+1}_t = s_k) \in \{0,1\}$. Since the state transition occurs in this regime (at time t), we might as well choose a different representation then to get rid of this rather trivial time-dependence. In fact, because of the causal influence of nearest neighbors, each of the eight possible triplet configurations really is a causal state. We are again justified in segmenting the former epsilon machine, as shown formally below. We will proceed by considering $P(S^{n-1}_t = s_j, S^{n+1}_t = s_k) \;*\; P(S^n_{t+1} = s \mid S^{n-1}S^n\, S^{n+1}{}_t = s_j s' s_k)$ in more detail.

First, we consider the separate paths obtained from this transition probability, assuming that it could come from any of the four triplet measurements consistent with $S^n_t = s'$, and show that $P(S^{n-1}_t = s_j, S^{n+1}_t = s_k)$ is only nonzero for one of the previous triplet states:

$$P(S^{n-1}_t = s_j, S^{n+1}_t = s_k) \;=\; \Big\langle P(S^{n-1}_t = s_j,\;\; S^{n+1}_t = s_k \;\mid\; S^{n-1}S^n\, S^{n+1}{}_t = s_1 s_2 s_3)\Big\rangle_{s_1 s_2 s_3}$$

$$= \sum_{s_1 \in A}\sum_{s_2 \in A}\sum_{s_3 \in A}\Big\{P(S^{n-1}S^n S^{n+1}{}_t = s_1 s_2 s_3) \;*\; P(S^{n-1}_t = s_j,\;\; S^{n+1}_t = s_k \;\mid\; S^{n-1}S^n\, S^{n+1}{}_t = s_1 s_2 s_3)\Big\}$$

$$= \sum_{s_1 \in A}\sum_{s_2 \in A}\sum_{s_3 \in A}\Big\{P(S^{n-1}S^n S^{n+1}{}_t = s_1 s_2 s_3) \;*\; \delta_{s_j, s_1}\delta_{s_k, s_3}\Big\}$$

Given that we are in the triplet state, $S^{n-1}S^n S^{n+1}{}_t = s_1 s_2 s_3$, this part of the transition probability normalizes to unity.

Next, we show that $P(S^n_{t+1} = s \mid S^{n-1}S^n\, S^{n+1}{}_t = s_j s' s_k)$ is simply the sum [linear superposition] of separate transition probabilities [-y pathways] among the eight new states:

$$P(S^n_{t+1} = s \mid S^{n-1}S^n\, S^{n+1}{}_t = s_1 s_2 s_3) \;=\; \sum_{s_L \in A}\sum_{s_R \in A}\Big\{P(S^{n-1}S^n S^{n+1}{}_{t+1} = s_L s s_R \;\mid\; S^{n-1}S^n\, S^{n+1}{}_t = s_1 s_2 s_3)\Big\}$$

So, the nodes have split into the more explicit triplets and have taken along their natural transition probabilities and symbols. [Because of the renaming of states, we note that the new transition probabilities are of the form, $P(S^{n-1}S^nS^{n+1}{}_{t+1} = s_L s s_R \;\;|\;\; S^{n-1}S^n\, S^{n+1}{}_t = s_1 s_2 s_3)$. The previous equation serves as a mapping between the alternatively nested representations of the machine.] The new representation of the single-element machine is presented below. Yet our work is not done. As we have become more explicit, we have noticed new dependencies in our transition probabilities. In general this trend can continue, and we are obligated to pursue even the most subtle effects on transition probabilities if we are to claim to have found the epsilon machine, or minimum statistic, for the system.

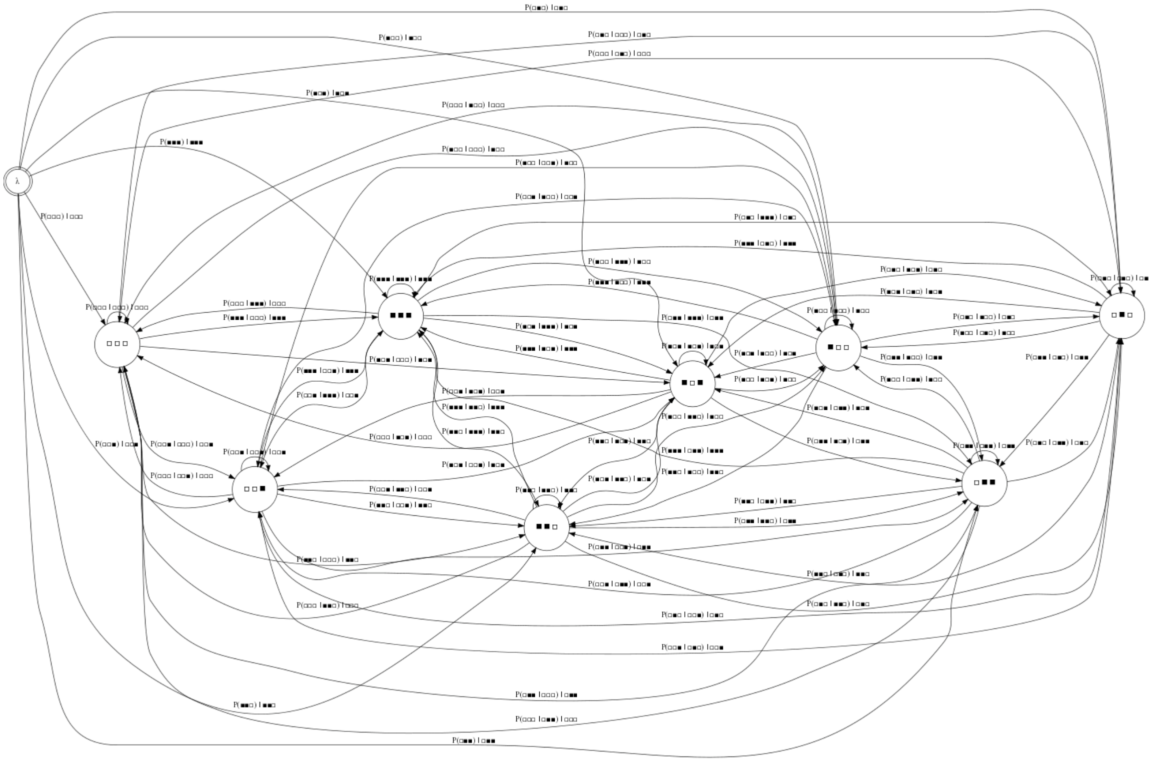


Figure 12

A Waft of the Spacetime Onion

If we track the probability distribution of the entire spacetime lattice, given all previous measurements, as best as possible, then we are truly representing the transition probabilities. Is this necessary? Yes, if we want a minimal statistic.

We can continue to reduce the apparent randomness in our system by being explicit about the spacetime dependence of the probabilities, as functions of n and t. An intelligent agent sitting at a single lattice site will be capable of reconstructing different amounts of the lattice's discrete-spacetime past and future for different systems. However, more surprisingly, an intelligent agent at a single lattice point may be able to reconstruct the *present* values of the whole lattice with some probability distribution. This eerily EPR-type result deserves some attention! Although information cannot physically travel faster than the system's light speed, the probability of an event could depend on the expected value of another event outside of their mutual light-cones. A more intuitive explanation is possible: given a sufficiently long time series, an agent at a single lattice point can reconstruct a probability distribution for the initial conditions for the whole lattice (assuming a finite lattice; otherwise read 'whole' as 'observable' à la observable universe). The learned rules of local interaction can then be applied to the inferred global probability distribution to construct a probability distribution for the current states over the entire lattice. This spacetime probability distribution should then be evolved to predict the future of the entire lattice at some arbitrary future time. In fact this process of prediction and retrodiction can go back and forth at every timestep to keep the most accurate record of the global spacetime probability distribution, even for probabilities outside of one's light cone. To be general, one must be completely explicit in all possible spacetime contributions to the transition probabilities. However, this extra effort is not warranted for a typical system. Yet the general formalism will help describe the complexity of all possible systems under considerations, and will allow for a classification scheme. If known, the initial conditions (and more generally, the evolution of the word probability distributions) will also change the time-dependent conditional probabilities. An intelligent agent at some lattice point (imagine a scientist sitting inside of a CA cell only able to see his own and directly neighboring cells) could discover these time dependent properties, so they really do belong explicitly in the epsilon machine of the system.

Inferring Machines and Basins for Small Simple Systems

From an engineering perspective, we might be interested in how a probability distribution evolves from a particular initial condition. For example, we could find sets of inputs that result in a desired behavior. Or instead we might want to design a system that carries out a particular function on a predefined set of inputs. The latter situation would be desirable, for example, to implement an image processing task such as edge detection.

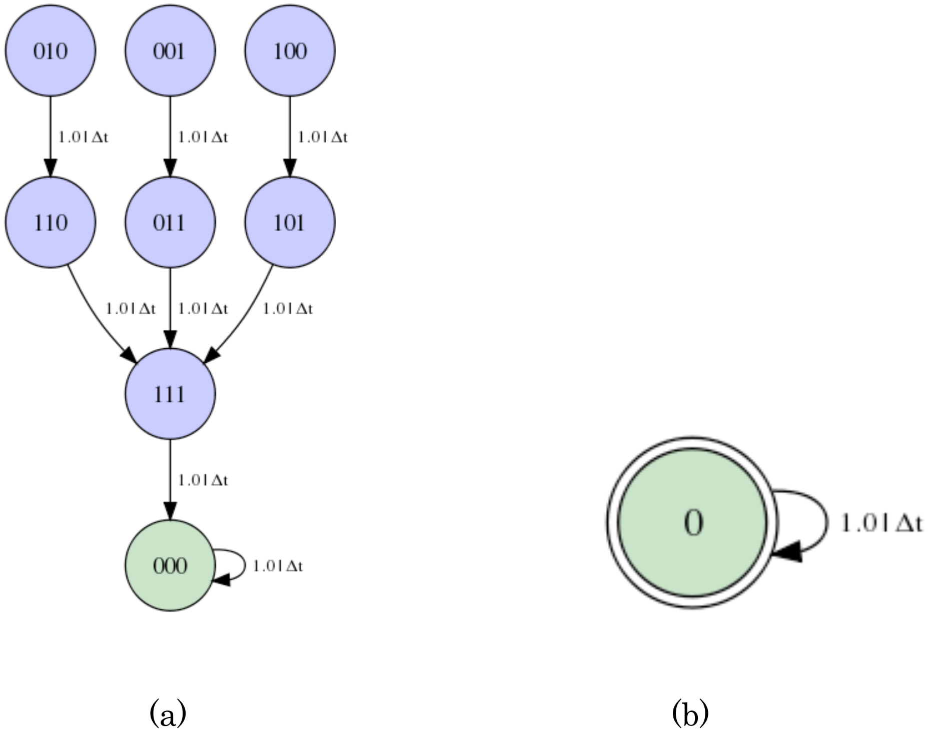


Figure 13

Above are the Markov process (a) and resulting epsilon Machine (b) for a uniform lattice of three circularly bound Rule 110 CA.

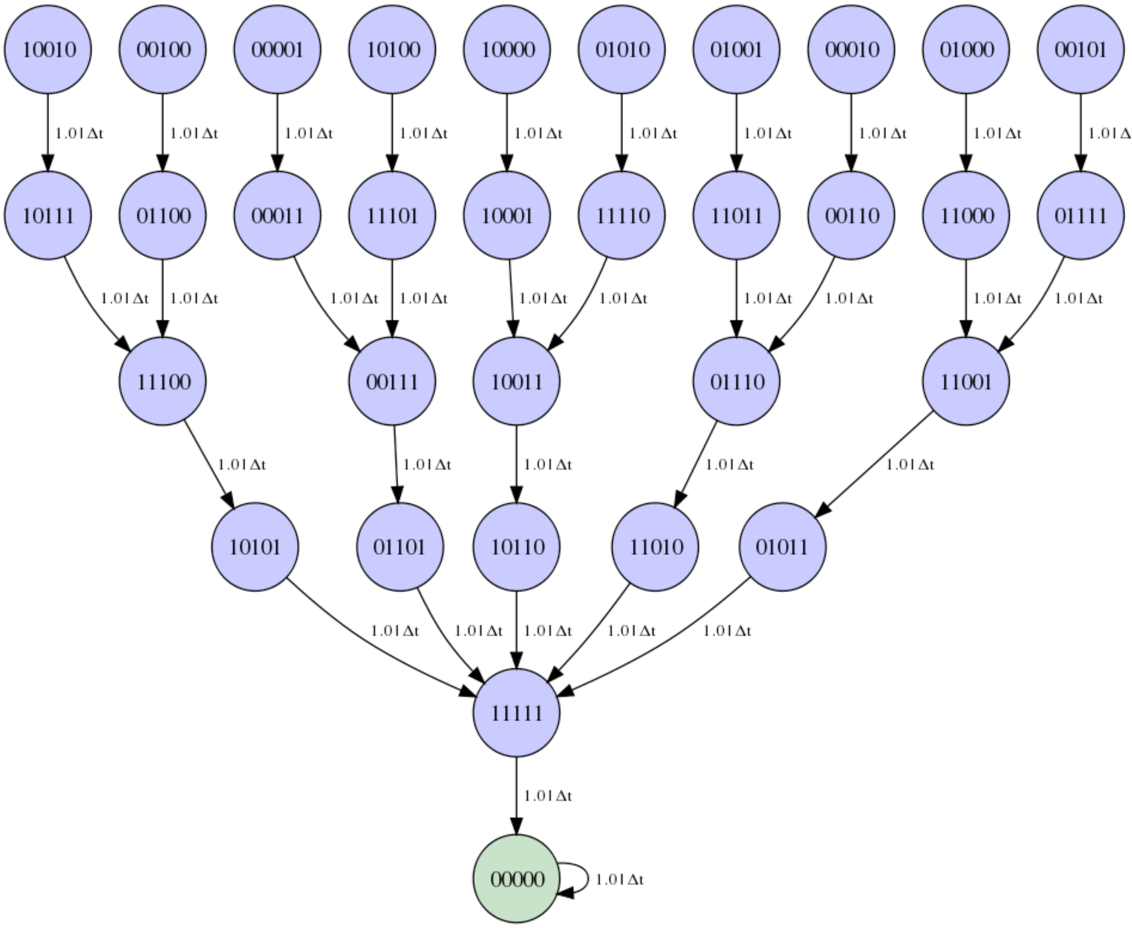


Figure 14(a)

Above, Figure 14 (a) is the Markov process for a uniform lattice of five circularly bound Rule 110 CA. Below, Figure 14 (b) is the resulting epsilon Machine for the process.

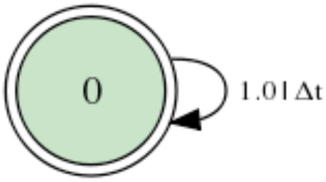


Figure 14 (b)

It seems like the above results are predictable and rather uninteresting. Is there any reason to continue this investigation? The following sequence of four panels is the Markov process for a uniform lattice of ten circularly bound Rule 110 CA, connected from left to right. We see that there are multiple basins of attraction with steady state sequences of different periodicity. The structure in the graph is also rather stunning, although we must be careful to distinguish between graph-layout algorithms and the true underlying geometry.

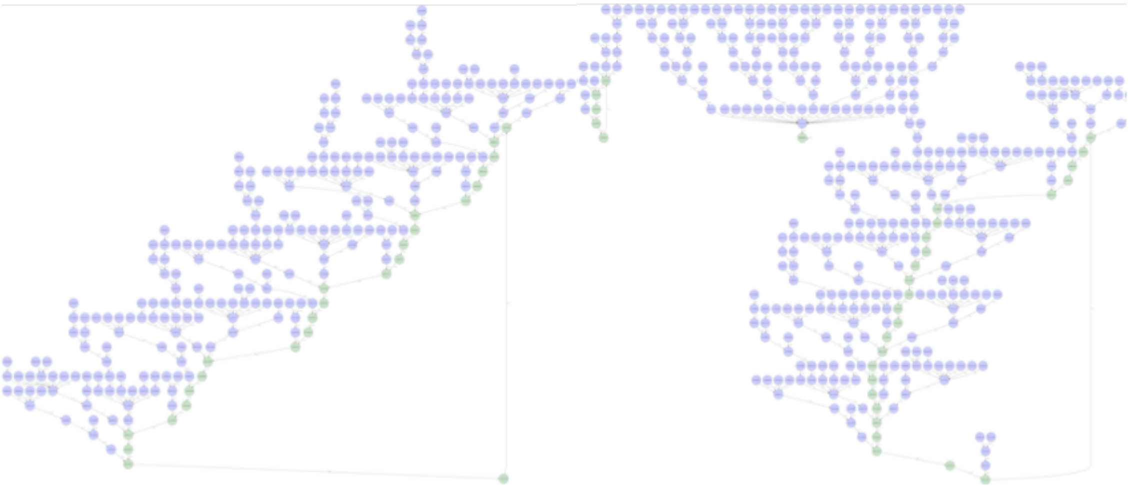


Figure 15 (Left)

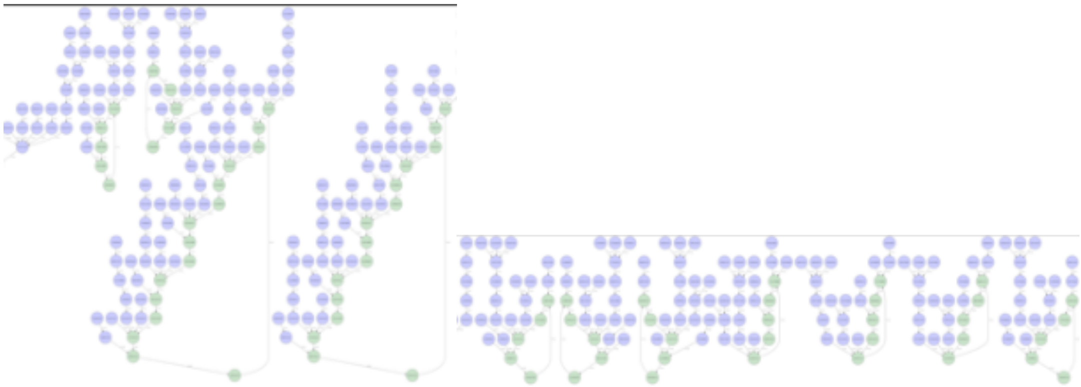
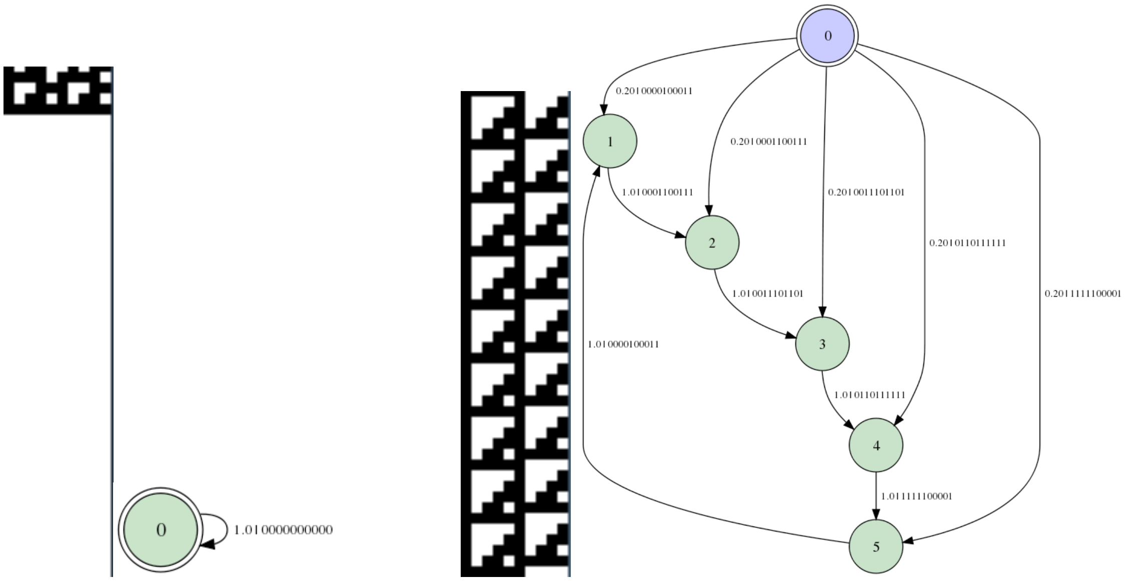


Figure 15 (Right)

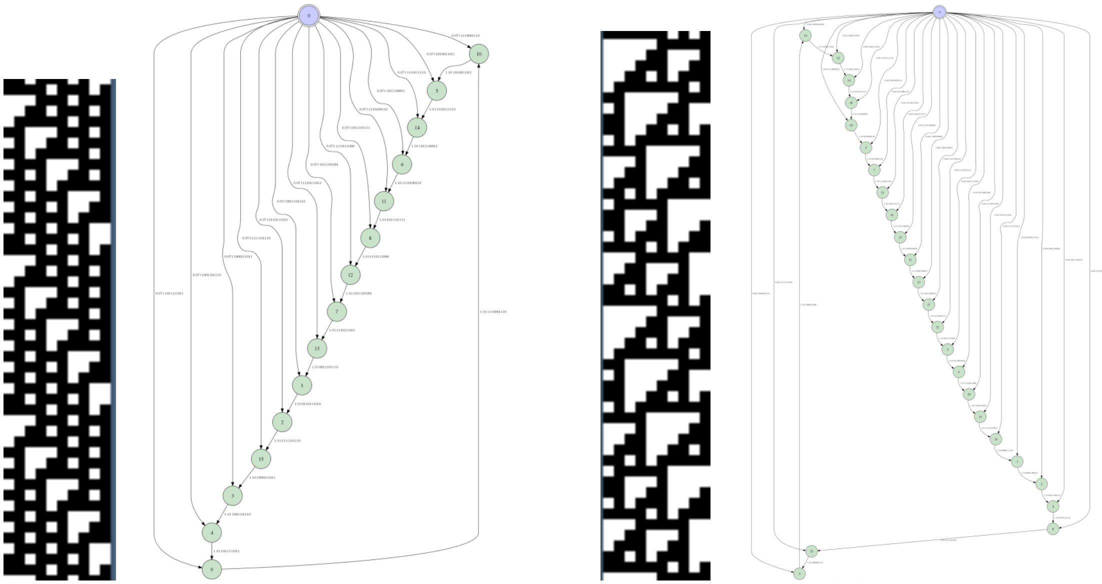
Although the 1024 nodes of the graph make many details hard to see, the green coloring makes recurrent states readily identifiable. In the end, there are only four distinct recurrent processes when considering translational invariance. Much of the structure seems to come from the redundancy of translationally invariant states in this uniform CA.



(a) **Figure 16** (b)

As seen in Figure 16 (a), the all zeros (ten-circularly bounded rule 110 CA with all zero values) state returns to itself with period one.

One can land in the period-five basin by starting with any translation of the sequence of six consecutive ones followed by four consecutive zeros (b).



(c) **Figure 16** (d)

Five consecutive zeros will land you immediately in the period-twenty-five basin, of Figure 16 (c), above.

Finally, perhaps one of the more unique basins, including states with only two zeros separated by three ones, has period-fifteen: Figure 16 (d).

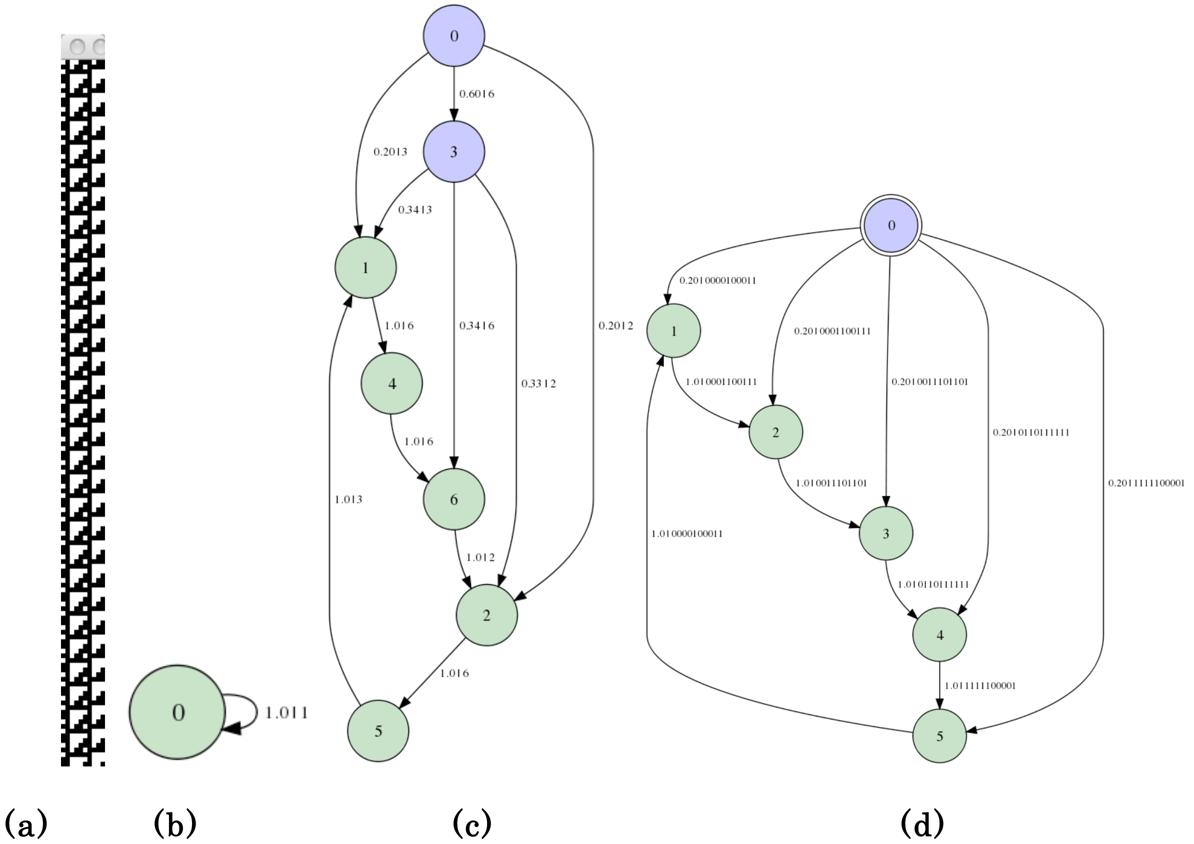


Figure 17

(a) simulation of CA rule 110 with a length-10 lattice and IC 0100111011.
(b) inferred epsilon machine using binary string data of cell[1] (the second cell on the left (all black sequence)).
(c) inferred epsilon machine using nearest neighbor string data of cell[1] (except the states are actually labeled backwards).
(d) actual epsilon machine for the basin of attraction associated with the IC.

Results

In pursuit of a computational mechanics for nanoelectronic arrays, we found ourselves delving head first into the more general but necessary prerequisite question of how computational mechanics can be extended to accomodate spatiotemporal processes. We considered several different incarnations of L, but were always guided by the *causal onion principle* that a word of length L should represent a measurement from the system that includes L layers of causal information. Investigation of the strictly-spatial onion led us to some interesting constraints on typical entropy measures, and a conclusion that effective measures, as functions of L, will usually be more insightful when trying to figure out the information processing capabilities of your process. Exploration of other onions led us to discover an intricate relationship between spatial and temporal probability densities, with the conclusion that a spacetime probability density should be constantly updated through prediction and retrodiction to truly have a minimal statistic for the system. This is not always necessary, but is generally so, for the class of systems considered in this project.

References

[1] Paul M. Riechers and Richard A. Kiehl, “CNN Implemented by Nonlinear Phase Dynamics in Nanoscale Processes,” 2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010).

[2] C. R. Shalizi and J. P. Crutchfield, "Computational Mechanics: Pattern and Prediction, Structure and Simplicity", Journal of Statistical Physics 104 (2001) 819--881.

[3] Leon O Chua. *CNN: A PARADIGM FOR COMPLEXITY.* World Scientific Series on Nonlinear Science, Series A – Vol. 31 , 1998.

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

[5] Moshe Sipper, “Co-evolving Non-Uniform Cellular Automata to Perform Computations,” Physica D, 92:193-208, 1996.

[6] David Feldman. *Computational Mechanics of Classical Spin Systems*. PhD thesis, University of California at Davis, 1998.

[7] David P. Feldman and James P. Crutchfield, “Structural Information in Two-Dimensional Patterns: Entropy Convergence and Excess Entropy,” Santa Fe Institute Working Paper 02-11-065 arxiv.org/abs/cond-mat/0212078

[8] Mark H. Bickhard with Donald T. Campbell, “Emergence,” http://www.lehigh.edu/~mhb0/Emergence27Jul97.pdf

[9] Bertrand Mesot and Christof Teuscher, “Deducing local rules for solving global tasks with random Boolean networks,” Physica D, 211 (2005) 88-106

[10] Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek, *Spikes: Exploring the Neural Code*. The MIT Press. 1999.

[11] Thomas M. Cover and Joy A. Thomas, *Elements of Information Theory.* Second Edition. Wiley-Interscience. New York (2006).

Appendix: Future Directions

This project was primarily motivated by some observations, questions, and frustrations I had while exploring the theoretical information processing capabilities of nanoscale arrays. In Ref. [1], we emphasize the limitations that a uniformly coupled physical array imposes on its computational abilities. My original project goal was to focus on the benefits of nonuniform coupling, but I found that there was much work in just figuring out a consistent framework for computational mechanics in a spatially extended system. As evident by the many loose ends in this paper, this very basic theoretical development is still far from complete. However, through this project, I have started to develop some computational-mechanical tools that could help guide the development of future nanoscale arrays. I will continue to think about the basics, but I am also interested in extending this foundation to address the ambitious questions I set out with.

Most of my considerations so far have been limited to very basic coupling paradigms. Eventually this methodology should be quite easily extendable to a more general class of systems, where for example the bistability and nearest-neighbor conditions are relaxed. I hope that I can develop the spatiotemporal computational mechanical tools further to be able to address my questions about how to design an array to utilize nonuniform coupling. I am interested in manipulating coupling parameters and topology to realize supercells that implement more desirable rules than a simple physical process can exhibit solely with uniform coupling. A potential goal for a functional cluster in an array could be to mimic a CA rule. At a higher hierarchical level, supercells of different CA-like rules could then be used as building blocks in a non-uniformly coupled CA array. Again, at this level, we need a useful set of tools to find useful emergent computational properties of the array. As a simple illustrative example of non-uniform coupling, consider a three-unit CA block with circular boundary conditions. It appears that we can implement any periodic pattern with $T \leq 8$, as long as no word is repeated twice in the sequence. Designing such a trio is not too hard. For example, coupling CA rules, – 108 – 102 – 51 –, in this way yields a three-bit binary counter. Specifically, the new rule on the three bits is: 'Count to the next integer, (mod 8).' However, the aforementioned example is not really exemplary of the desired result of this project, because the trio does not use anything other than local information, and does not exploit the computational powers of wild boundary conditions. Drastically more interesting behavior should be possible for larger non-uniform CA arrays. As a proven example, nearest-neighbor non-uniformly coupled CA have been evolved to outperform the best *possible* nearest-neighbor uniformly coupled CA for the task of deciding whether or not the majority of initial cell values were 1's [5]. The continued development of this project will contribute to the understanding of arrays with a small number of available components yet a large number of coupling constraints.