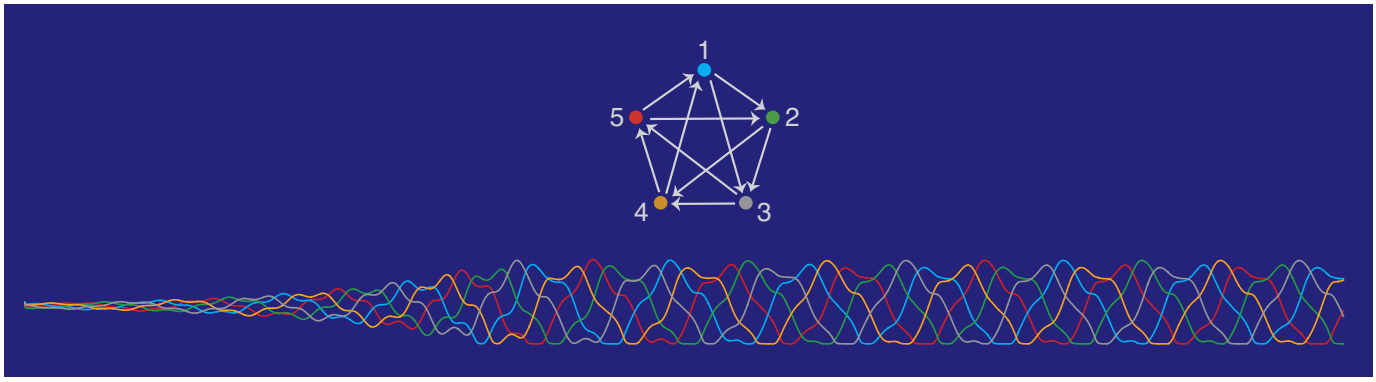
Graph Rules for Recurrent Neural Network Dynamics



Carina Curto and Katherine Morrison

1. Introduction

Neurons in the brain are constantly flickering with activity, which can be spontaneous or in response to stimuli [LBH09]. Because of positive feedback loops and the potential for runaway excitation, real neural networks often possess an abundance of inhibition that serves to shape and stabilize the dynamics [YMSL05, KAY14]. The excitatory neurons in such networks exhibit intricate patterns of connectivity, whose structure controls the allowed patterns of activity. A central question in neuroscience is thus: how does network connectivity shape dynamics?

For a given model, this question becomes a mathematical challenge. The goal is to develop a theory that directly relates properties of a nonlinear dynamical system to its underlying graph. Such a theory can provide insights and hypotheses about how network connectivity constrains activity in real brains. It also opens up new possibilities for modeling neural phenomena in a mathematically tractable way.

Carina Curto is a professor of mathematics at Penn State University. Her email address is ccurto@psu.edu.

Carina Curto was supported by NIH R01 EB022862, NIH R01 NS120581, NSF DMS-1951165, and a Simons Fellowship.

Katherine Morrison is an associate professor of mathematics at the University of Northern Colorado. Her email address is katherine.morrison@unco.edu.

Katherine Morrison was supported by NIH R01 EB022862 and NSF DMS-1951599.

Communicated by Notices *Associate Editor Emilie Purvine.*

DOI: https://doi.org/10.1090/noti2661

Here we describe a class of inhibition-dominated neural networks corresponding to directed graphs, and introduce some of the theory that has been developed to study them. The heart of the theory is a set of parameter-independent *graph rules* that enables us to directly predict features of the dynamics from combinatorial properties of the graph. Specifically, graph rules allow us to constrain, and in some cases fully determine, the collection of stable and unstable fixed points of a network based solely on graph structure.

Stable fixed points are themselves static attractors of the network, and have long been used as a model of stored memory patterns [Hop82]. In contrast, unstable fixed points have been shown to play an important role in shaping *dynamic* (nonstatic) attractors, such as limit cycles [PMMC22]. By understanding the fixed points of simple networks, and how they relate to the underlying architecture, we can gain valuable insight into the high-dimensional nonlinear dynamics of neurons in the brain.

For more complex architectures, built from smaller component subgraphs, we present a series of *gluing rules* that allow us to determine all fixed points of the network by gluing together those of the components. These gluing rules are reminiscent of sheaf-theoretic constructions, with fixed points playing the role of sections over subnetworks.

First, we review some basics of recurrent neural networks and a bit of historical context.

Basic network setup. A *recurrent neural network* is a directed graph G together with a prescription for the dynamics on the vertices, which represent neurons (see Figure 1A). To each vertex i we associate a function $x_i(t)$ that tracks the activity level of neuron i as it evolves in time. To

each ordered pair of vertices (i, j) we assign a weight, W_{ij}, governing the strength of the influence of neuron j on neuron i. In principle, there can be a nonzero weight between any two nodes, with the graph G providing constraints on the allowed values W_{ij}, depending on the specifics of the model.

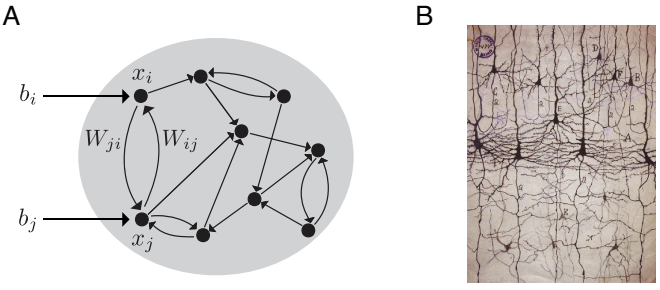


Figure 1. (A) Recurrent network setup. (B) A Ramón y Cajal drawing of real cortical neurons.

The dynamics often take the form of a system of ODEs, called a *firing rate model* [DA01]:

$$\tau_i \frac{dx_i}{dt} = -x_i + \varphi\left(\sum_{j=1}^{n} W_{ij}x_j + b_i\right), \tag{1}$$

$$= -x_i + \varphi(y_i), \tag{2}$$

for $i = 1, \ldots, n$. The various terms in the equation are illustrated in Figure 1, and can be thought of as follows:

- $x_i = x_i(t)$ is the firing rate of a single neuron i (or the average activity of a subpopulation of neurons);
- τ_i is the "leak" timescale, governing how quickly a neuron's activity exponentially decays to zero in the absence of external or recurrent input;
- W is a real-valued matrix of synaptic interaction strengths, with W_{ij} representing the strength of the connection from neuron j to neuron i;
- $b_i = b_i(t)$ is a real-valued external input to neuron i that may or may not vary with time;
- $y_i = y_i(t) = \sum_{j=1}^{n} W_{ij}x_j(t) + b_i(t)$ is the total input to neuron i as a function of time; and
- $\varphi : \mathbb{R} \to \mathbb{R}$ is a nonlinear, but typically monotone increasing function.

Of particular importance for this article is the family of *threshold-linear networks* (TLNs). In this case, the nonlinearity is chosen to be the popular threshold-linear (or ReLU) function,

$$\varphi(y) = [y]_+ = \max\{0, y\}.$$

TLNs are common firing rate models that have been used in computational neuroscience for decades [SY12, TSSM97, HSM+00, BF22]. The use of threshold-linear units in neural modeling dates back at least to 1958 [HR58]. In the last 20 years, TLNs have also been shown to be surprisingly tractable mathematically [HSS03, CDI13], [CM16, MDIC16, CGM19, PLACM22], though much of the theory remains underdeveloped. We are especially interested in *competitive* or *inhibition-dominated* TLNs, where the W matrix is nonpositive so the effective interaction between any pair of neurons is inhibitory. In this case, the activity remains bounded despite the lack of saturation in the nonlinearity [MDIC16]. These networks produce complex nonlinear dynamics and can possess a remarkable variety of attractors [MDIC16, PLACM22, PMMC22].

Firing rate models of the form (1) are examples of *recurrent* networks because the W matrix allows for all pairwise interactions, and there is no constraint that the architecture (i.e., the underlying graph G) be feedforward. Unlike deep neural networks, which can be thought of as classifiers implementing a clustering function, recurrent networks are primarily thought of as dynamical systems. And the main purpose of these networks is to model the dynamics of neural activity in the brain. The central question is thus:

Question 1. Given a firing rate model defined by (1) with network parameters (W, b) and underlying graph G, what are the emergent network dynamics? What can we say about the dynamics from knowledge of G alone?

We are particularly interested in understanding the *attractors* of such a network, including both stable fixed points and dynamic attractors such as limit cycles. The attractors are important because they comprise the set of possible asymptotic behaviors of the network in response to different inputs or initial conditions (see Figure 2).

Note that Question 1 is posed for a fixed connectivity matrix W, but of course W can change over time (e.g., as a result of learning or training of the network). Here we restrict ourselves to considering constant W matrices; this allows us to focus on understanding network dynamics on a fast timescale, assuming slowly varying synaptic weights. Understanding the dynamics associated to changing W is an important topic, currently beyond the scope of this work.

Historical interlude: memories as attractors. Attractor neural networks became popular in the 1980s as models of associative memory encoding and retrieval. The best-known example from that era is the Hopfield model [Hop82], originally conceived as a variant on the Ising model from statistical mechanics. In the Hopfield model, the neurons can be in one of two states, $s_i \in \{\pm 1\}$, and the activity evolves according to the discrete time update rule:

$$s_i(t+1) = \operatorname{sgn}\left(\sum_{j=1}^{n} W_{ij}s_j(t) - \theta_i\right).$$

Hopfield's famous 1982 result is that the dynamics are guaranteed to converge to a stable fixed point, provided the interaction matrix W is *symmetric*: that is, $W_{ij} = W_{ji}$

for every $i, j \in \{1, \ldots, n\}$. Specifically, he showed that the "energy" function,

$$E = -\frac{1}{2}\sum_{i,j} W_{ij} s_i s_j + \sum_i \theta_i s_i,$$

decreases along trajectories of the dynamics, and thus acts as a Lyapunov function [Hop82]. The stable fixed points are local minima of the energy landscape (Figure 2A). A stronger, more general convergence result for competitive neural networks was shown in [CG83].

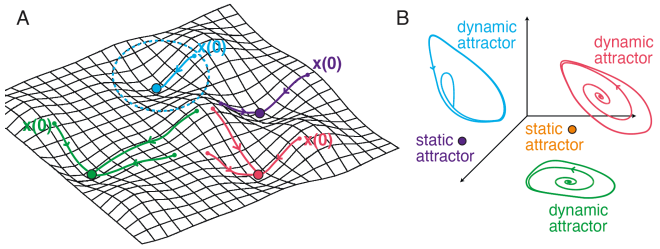


Figure 2. Attractor neural networks. (A) For symmetric Hopfield networks and symmetric inhibitory TLNs, trajectories are guaranteed to converge to stable fixed point attractors. Sample trajectories are shown, with the basin of attraction for the blue stable fixed point outlined in blue. (B) For asymmetric TLNs, dynamic attractors can coexist with (static) stable fixed point attractors.

These fixed points are the only attractors of the network, and they represent the set of memories encoded in the network. Hopfield networks perform a kind of *pattern completion*: given an initial condition $s(0)$, the activity evolves until it converges to one of multiple stored patterns in the network. If, for example, the individual neurons store black and white pixel values, this process could input a corrupted image and recover the original image, provided it has previously been stored as a stable fixed point in the network by appropriately selecting the weights of the W matrix. The novelty at the time was the nonlinear phenomenon of multistability: namely, that the network could encode many such stable equilibria and thus maintain an entire catalogue of stored memory patterns. The key to Hopfield's convergence result was the requirement that W be a symmetric interaction matrix. Although this was known to be an unrealistic assumption for real (biological) neural networks, it was considered a tolerable price to pay for guaranteed convergence. One did not want an associative memory network that wandered the state space indefinitely without ever recalling a definite pattern.

Twenty years later, Hahnloser, Seung, and others followed up and proved a similar convergence result in the case of symmetric threshold-linear networks [HSS03]. More results on the collections of stable fixed points that can be simultaneously encoded in a symmetric TLN can be found in [CDI13, CM16], including some unexpected connections to Cayley–Menger determinants and classical distance geometry.

In all of this work, stable fixed points have served as the model for encoded memories. Indeed, these are the only types of attractors that arise for symmetric Hopfield networks or symmetric TLNs. Whether or not guaranteed convergence to stable fixed points is desirable, however, is a matter of perspective. For a network whose job it is to perform pattern completion or classification for static images (or codewords), as in the classical Hopfield model, this is exactly what one wants. But it is also important to consider memories that are temporal in nature, such as sequences and other dynamic patterns of activity. Sequential activity, as observed in central pattern generator circuits (CPGs) and spontaneous activity in hippocampus and cortex, is more naturally modeled by dynamic attractors such as limit cycles. This requires shifting attention to the *asymmetric* case, in order to be able to encode attractors that are not stable fixed points (Figure 2B).

Beyond stable fixed points. When the symmetry assumption is removed, TLNs can support a rich variety of dynamic attractors such as limit cycles, quasiperiodic attractors, and even strange (chaotic) attractors. Indeed, this richness can already be observed in a special class of TLNs called combinatorial threshold-linear networks (CTLNs), introduced in Section 3. These networks are defined from directed graphs, and the dynamics are almost entirely determined by the graph structure. A striking feature of CTLNs is that the dynamics are shaped not only by the stable fixed points, but also the *unstable* fixed points. In particular, we have observed a direct correspondence between certain types of unstable fixed points and dynamic attractors (see Figure 3). This is reviewed in Section 4.

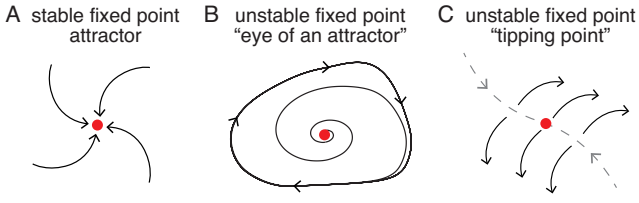


Figure 3. Stable and unstable fixed points. (A) Stable fixed points are attractors of the network. (B-C) Unstable fixed points are not themselves attractors, but certain unstable fixed points seem to correspond to dynamic attractors (B), while others function solely as tipping points between multiple attractors (C).

Despite exhibiting complex, high-dimensional, nonlinear dynamics, recent work has shown that TLNs—and especially CTLNs—are surprisingly tractable mathematically. Motivated by the relationship between fixed points and attractors, a great deal of progress has been made on the problem of relating fixed point structure to network

architecture. In the case of CTLNs, this has resulted in a series of *graph rules*: theorems that allow us to rule in and rule out potential fixed points based purely on the structure of the underlying graph [CGM19, PLACM22]. In Section 5, we give a novel exposition of graph rules, and introduce several *elementary graph rules* from which the others can be derived.

Inhibition-dominated TLNs and CTLNs also display a remarkable degree of modularity. Namely, attractors associated to smaller networks can be embedded in larger ones with minimal distortion [PMMC22]. This is likely a consequence of the high levels of background inhibition: it serves to stabilize and preserve local properties of the dynamics. These networks also exhibit a kind of compositionality, wherein fixed points and attractors of subnetworks can be effectively "glued" together into fixed points and attractors of a larger network. These local-to-global relationships are given by a series of theorems we call *gluing rules*, given in Section 6.

2. TLNs and Hyperplane Arrangements

For firing rate models with threshold-nonlinearity $\varphi(y) = [y]_+ = \max\{0, y\}$, the network equations (1) become

$$\frac{dx_i}{dt} = -x_i + \left[\sum_{j=1}^{n} W_{ij}x_j + b_i\right]_+ = -x_i + [y_i]_+, \qquad (3)$$

for $i = 1, \ldots, n$. We also assume $W_{ii} = 0$ for each i. Note that the leak timescales have been set to $\tau_i = 1$ for all i. We thus measure time in units of this timescale.

For constant W matrix and input vector b, the equations

$$y_i = \sum_{j=1}^{n} W_{ij}x_j + b_i = 0,$$

define a hyperplane arrangement $\mathcal{H} = \mathcal{H}(W, b) = \{H_1, \ldots, H_n\}$ in $\mathbb{R}^n$. The i-th hyperplane H_i is defined by $y_i = \vec{n}_i \cdot x + b_i = 0$, with normal vector $\vec{n}_i = (W_{i1}, \ldots, W_{in})$, population activity vector $x = (x_1, \ldots, x_n)$, and affine shift b_i. If $W_{ij} \neq 0$, then H_i intersects the j-th coordinate axis at the point $x_j = -b_i/W_{ij}$. H_i is parallel to the i-th axis.

The hyperplanes $\mathcal{H}$ partition the positive orthant $\mathbb{R}^n_{\geq 0}$ into chambers. Within the interior of each chamber, each point x is on the plus or minus side of each hyperplane H_i. The equations thus reduce to a linear system of ODEs, with either $dx_i/dt = -x_i$ or $dx_i/dt = -x_i + y_i = -x_i + \sum_{j=1}^{n} W_{ij}x_j + b_i$ for each i. In particular, TLNs are piecewise-linear dynamical systems with a different linear system governing the dynamics in each chamber.

A *fixed point* of a TLN (3) is a point $x^* \in \mathbb{R}^n$ that satisfies $dx_i/dt|_{x=x^*} = 0$ for each $i \in \{1, \ldots, n\}$. In particular, we must have

$$x_i^* = [y_i^*]_+ \text{ for all } i = 1, \ldots, n, \qquad (4)$$

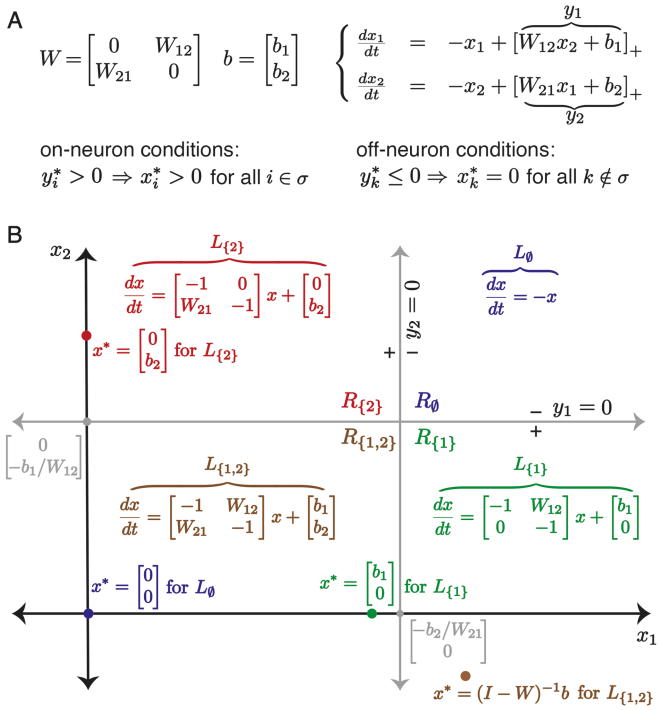


Figure 4. TLNs as a patchwork of linear systems. (A) The connectivity matrix W, input b, and differential equations for a TLN with $n = 2$ neurons. (B) The state space is divided into chambers (regions) R_σ, each having dynamics governed by a different linear system L_σ. The chambers are defined by the hyperplanes $\{H_i\}_{i=1,2}$, with H_i defined by $y_i = 0$ (gray lines).

where y_i^* is y_i evaluated at the fixed point. We typically assume a nondegeneracy condition on (W, b) [CGM19], which guarantees that each linear system is nondegenerate and has a single fixed point. This fixed point may or may not lie within the chamber where its corresponding linear system applies. The fixed points of the TLN are precisely the fixed points of the linear systems that lie within their respective chambers.

Figure 4 illustrates the hyperplanes and chambers for a TLN with $n = 2$. Each chamber, denoted as a region R_σ, has its own linear system of ODEs, L_σ, for $\sigma = \emptyset, \{1\}, \{2\}$, or $\{1, 2\}$. The fixed points corresponding to each linear system are denoted by x^*, in matching color. Note that only chamber $R_{\{2\}}$ contains its own fixed point (in red). This fixed point, $x^* = [0, b_2]^T$, is thus the only fixed point of the TLN.

Figure 5 shows an example of a TLN on $n = 3$ neurons. The W matrix is constructed from a 3-cycle graph and $b_i = \theta = 1$ for each i. The dynamics fall into a limit cycle where the neurons fire in a repeating sequence that follows the arrows of the graph. This time, the TLN equations define a hyperplane arrangement in $\mathbb{R}^3$, again with each hyperplane H_i defined by $y_i = 0$ (Figure 5C). An initial condition near the unstable fixed point in the all + chamber (where $y_i > 0$ for each i) spirals out and converges

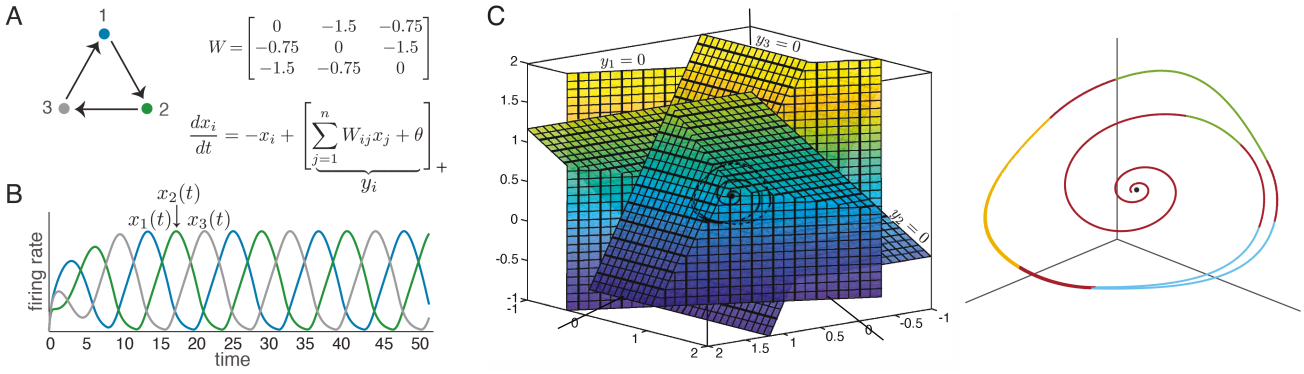


Figure 5. A network on $n = 3$ neurons, its hyperplane arrangement, and limit cycle. (A) A TLN whose connectivity matrix W is dictated by a 3-cycle graph, together with the TLN equations. (B) The TLN from A produces firing rate activity in a periodic sequence. (C) (Left) The hyperplane arrangement defined by the equations $y_i = 0$, with a trajectory initialized near the fixed point shown in black. (Right) A close-up of the trajectory, spiraling out from the unstable fixed point and falling into a limit cycle. Different colors correspond to different chambers of the hyperplane arrangement through which the trajectory passes.

to a limit cycle that passes through four distinct chambers. Note that the threshold nonlinearity is critical for the model to produce nonlinear behavior such as limit cycles; without it, the system would be linear. It is, nonetheless, nontrivial to prove that the limit cycle shown in Figure 5 exists. A recent proof was given for a special family of TLNs constructed from any k-cycle graph [BCRR21].

The set of all fixed points $\mathrm{FP}(W, b)$. A central object that is useful for understanding the dynamics of TLNs is the collection of *all* fixed points of the network, both stable and unstable. The *support* of a fixed point $x^* \in \mathbb{R}^n$ is the subset of active neurons,

$$\operatorname{supp} x^* \stackrel{\text{def}}{=} \{i \mid x_i^* > 0\}.$$

Our nondegeneracy condition (that is generically satisfied) guarantees we can have at most one fixed point per chamber of the hyperplane arrangement $\mathcal{H}(W, b)$, and thus at most one fixed point per support. We can thus label all the fixed points of a given network by their supports:

$$\mathrm{FP}(W, b) \stackrel{\text{def}}{=} \{\sigma \subseteq [n] \mid \sigma = \operatorname{supp} x^*, \text{ for some fixed pt } x^* \text{ of the associated TLN}\},$$

where $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$. For each support $\sigma \in \mathrm{FP}(W, b)$, the fixed point itself is easily recovered. Outside the support, $x_i^* = 0$ for all $i \notin \sigma$. Within the support, x^* is given by:

$$x_\sigma^* = (I - W_\sigma)^{-1} b_\sigma.$$

Here x_σ^* and b_σ are the column vectors obtained by restricting x^* and b to the indices in σ, and W_σ is the induced principal submatrix obtained by restricting rows and columns of W to σ.

From (4), we see that a fixed point with $\operatorname{supp} x^* = \sigma$ must satisfy the "on-neuron" conditions, $y_i^* > 0$ for all $i \in \sigma$, as well as the "off-neuron" conditions, $y_k^* \leq 0$ for all $k \notin \sigma$, to ensure that $x_i^* > 0$ for each $i \in \sigma$ and $x_k^* = 0$ for each $k \notin \sigma$. Equivalently, these conditions guarantee that the fixed point x^* of L_σ lies inside its corresponding chamber, R_σ. Note that for such a fixed point, the values x_i^* for $i \in \sigma$ depend only on the restricted subnetwork (W_σ, b_σ). Therefore, the on-neuron conditions for x^* in (W, b) are satisfied if and only if they hold in (W_σ, b_σ). Since the off-neuron conditions are trivially satisfied in (W_σ, b_σ), it follows that $\sigma \in \mathrm{FP}(W_\sigma, b_\sigma)$ is a necessary condition for $\sigma \in \mathrm{FP}(W, b)$. It is not, however, sufficient, as the off-neuron conditions may fail in the larger network.

Conveniently, the off-neuron conditions are independent and can be checked one neuron at a time. Thus,

$$\sigma \in \mathrm{FP}(W, b) \Leftrightarrow \sigma \in \mathrm{FP}(W_{\sigma\cup k}, b_{\sigma\cup k}) \text{ for all } k \notin \sigma.$$

When $\sigma \in \mathrm{FP}(W_\sigma, b_\sigma)$ satisfies all the off-neuron conditions, so that $\sigma \in \mathrm{FP}(W, b)$, we say that σ *survives* to the larger network; otherwise, we say σ *dies*.

The fixed point corresponding to $\sigma \in \mathrm{FP}(W, b)$ is *stable* if and only if all eigenvalues of $-I + W_\sigma$ have negative real part. For competitive (or inhibition-dominated) TLNs, all fixed points—whether stable or unstable—have a stable manifold. This is because competitive TLNs have $W_{ij} \leq 0$ for all $i, j \in [n]$. Applying the Perron–Frobenius theorem to $-I + W_\sigma$, we see that the largest magnitude eigenvalue is guaranteed to be real and negative. The corresponding eigenvector provides an attracting direction into the fixed point. Combining this observation with the nondegeneracy condition reveals that the unstable fixed points are all hyperbolic (i.e., saddle points).

3. Combinatorial Threshold-Linear Networks

Combinatorial threshold-linear networks (CTLNs) are a special case of competitive (or inhibition-dominated) TLNs, with the same threshold nonlinearity, that were first introduced in [MDIC16]. What makes CTLNs special is that we restrict to having only two values for the connection strengths W_{ij}, for $i \neq j$. These are obtained as follows from

a directed graph G, where $j \to i$ indicates that there is an edge from j to i and $j \not\to i$ indicates that there is no such edge:

$$W_{ij} = \begin{cases} 0 & \text{if } i = j, \\ -1+\varepsilon & \text{if } j \to i \text{ in } G, \\ -1-\delta & \text{if } j \not\to i \text{ in } G. \end{cases} \tag{5}$$

Additionally, CTLNs typically have a constant external input $b_i = \theta$ for all i in order to ensure the dynamics are internally generated rather than inherited from a changing or spatially heterogeneous input.

A CTLN is thus completely specified by the choice of a graph G, together with three real parameters: ε, δ, and θ. We additionally require that $\delta > 0$, $\theta > 0$, and $0 < \varepsilon < \frac{\delta}{\delta+1}$. When these conditions are met, we say the parameters are within the *legal range*. Note that the upper bound on ε implies $\varepsilon < 1$, and so the W matrix is always effectively inhibitory. For fixed parameters, only the graph G varies between networks. The network in Figure 5 is a CTLN with the *standard parameters* $\varepsilon = 0.25$, $\delta = 0.5$, and $\theta = 1$.

We interpret a CTLN as modeling a network of n excitatory neurons, whose net interactions are effectively inhibitory due to a strong global inhibition (Figure 6). When $j \not\to i$, we say j *strongly inhibits* i; when $j \to i$, we say j *weakly inhibits* i. The weak inhibition is thought of as the sum of an excitatory synaptic connection and the background inhibition. Note that because $-1-\delta < -1 < -1+\varepsilon$, when $j \not\to i$, neuron j inhibits i *more* than it inhibits itself via its leak term; when $j \to i$, neuron j inhibits i *less* than it inhibits itself. These differences in inhibition strength cause the activity to follow the arrows of the graph.

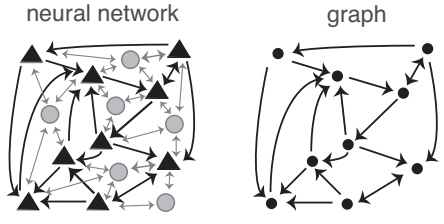


Figure 6. CTLNs. A neural network with excitatory pyramidal neurons (triangles) and a background network of inhibitory interneurons (gray circles) that produces a global inhibition. The corresponding graph (right) retains only the excitatory neurons and their connections.

The set of fixed point supports of a CTLN with graph G is denoted as:

$$\mathrm{FP}(G,\varepsilon,\delta) \stackrel{\text{def}}{=} \{\sigma \subseteq [n] \mid \sigma = \operatorname{supp} x^* \text{ for some fixed pt } x^* \text{ of the associated CTLN}\}.$$

$\mathrm{FP}(G,\varepsilon,\delta)$ is precisely $\mathrm{FP}(W,b)$, where W and b are specified by a CTLN with graph G and parameters ε and δ. Note that $\mathrm{FP}(G,\varepsilon,\delta)$ is independent of θ, provided θ is constant across neurons as in a CTLN. It is also frequently independent of ε and δ. For this reason we often refer to it as $\mathrm{FP}(G)$, especially when a fixed choice of ε and δ is understood.

The legal range condition, $\varepsilon < \frac{\delta}{\delta+1}$, is motivated by a theorem in [MDIC16]. It ensures that single directed edges $i \to j$ are not allowed to support stable fixed points $\{i,j\} \in \mathrm{FP}(G,\varepsilon,\delta)$. This allows us to prove the following theorem connecting a certain graph structure to the absence of stable fixed points. Note that a graph is *oriented* if for any pair of nodes, $i \to j$ implies $j \not\to i$ (i.e., there are no bidirectional edges). A *sink* is a node with no outgoing edges.

Theorem 3.1 ([MDIC16, Theorem 2.4]). *Let G be an oriented graph with no sinks. Then for any parameters $\varepsilon, \delta, \theta$ in the legal range, the associated CTLN has no stable fixed points. Moreover, the activity is bounded.*

The graph in Figure 5A is an oriented graph with no sinks. It has a single fixed point, $\mathrm{FP}(G) = \{123\}$, irrespective of the parameters (note that we use "123" as shorthand for the set $\{1,2,3\}$). This fixed point is unstable and the dynamics converge to a limit cycle (Figure 5C).

Even when there are no stable fixed points, the dynamics of a CTLN are always bounded [MDIC16]. In the limit as $t \to \infty$, we can bound the total population activity as a function of the parameters ε, δ, and θ:

$$\frac{\theta}{1+\delta} \le \sum_{i=1}^{n} x_i \le \frac{\theta}{1-\varepsilon}. \tag{6}$$

In simulations, we observe a rapid convergence to this regime.

Figure 7 depicts four solutions for the same CTLN on $n = 100$ neurons. The graph G was generated as a directed Erdos–Renyi random graph with edge probability $p = 0.2$; note that it is *not* an oriented graph. Since the network is deterministic, the only difference between simulations is the initial conditions. While panel A appears to show chaotic activity, the solutions in panels B, C, and D all settle into a fixed point or a limit cycle within the allotted time frame. The long transient of panel B is especially striking: around $t = 200$, the activity appears as though it will fall into the same limit cycle from panel D, but then escapes into another period of chaotic-looking dynamics before abruptly converging to a stable fixed point. In all cases, the total population activity rapidly converges to lie within the bounds given in (6), depicted in gray.

Fun examples. Despite their simplicity, CTLNs display a rich variety of nonlinear dynamics. Even very small networks can exhibit interesting attractors with unexpected properties. Theorem 3.1 tells us that one way to guarantee that a network will produce dynamic—as opposed to

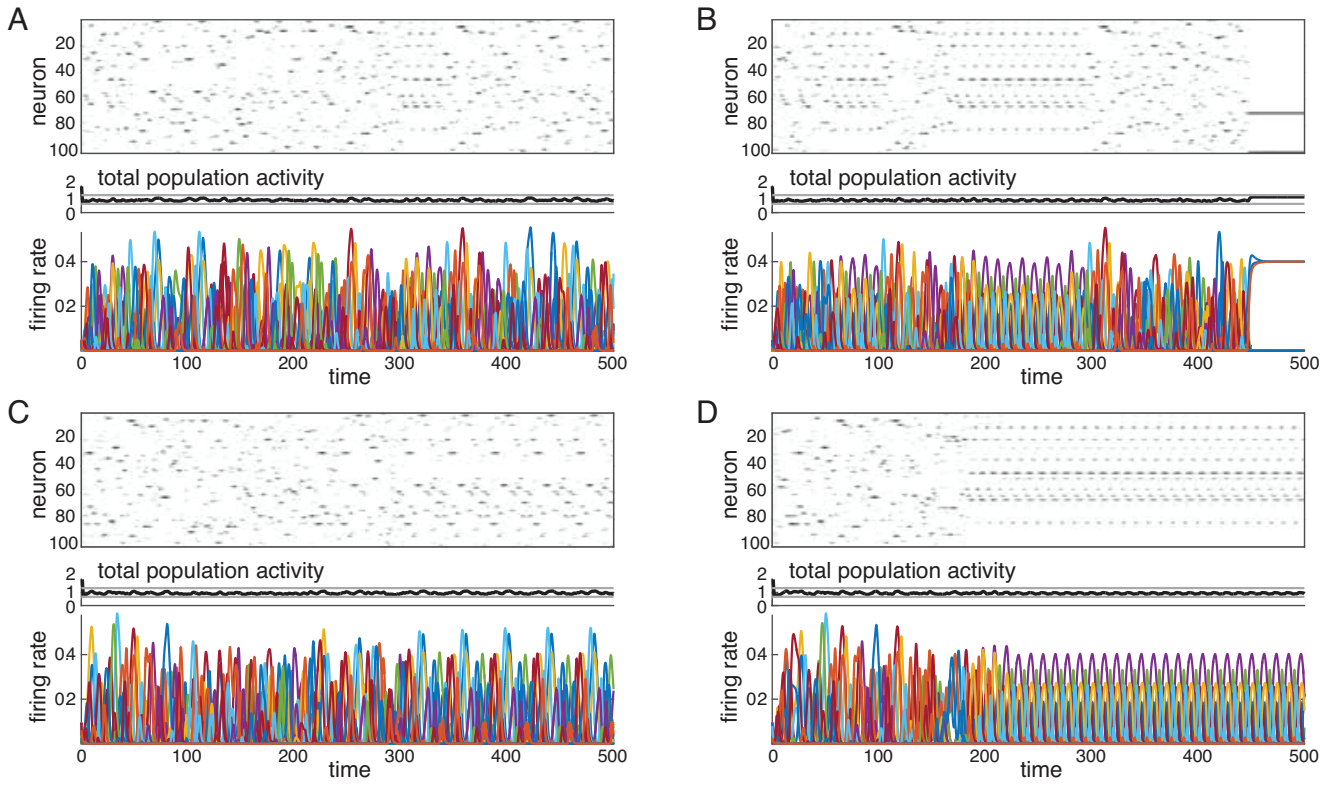


Figure 7. Dynamics of a CTLN network on $n = 100$ neurons. The graph G is a directed Erdos–Renyi random graph with edge probability $p = 0.2$ and no self loops. The CTLN parameters are $\varepsilon = 0.25$, $\delta = 0.5$, and $\theta = 1$. Initial conditions for each neuron, $x_i(0)$, are randomly and independently chosen from the uniform distribution on $[0, 0.1]$. (A-D) Four solutions from the same deterministic network, differing only in the choice of initial conditions. In each panel, the top plot shows the firing rate as a function of time for each neuron in grayscale. The middle plot shows the summed total population activity, $\sum_{i=1}^{n} x_i$, which quickly becomes trapped between the horizontal gray lines—the bounds in equation (6). The bottom plot shows individual rate curves for all 100 neurons, in different colors. (A) The network appears chaotic, with some recurring patterns of activity. (B) The solution initially appears to be chaotic, like the one in A, but eventually converges to a stable fixed point supported on a 3-clique. (C) The solution converges to a limit cycle after $t = 300$. (D) The solution converges to a different limit cycle after $t = 200$. Note that one can observe brief "echoes" of this limit cycle in the transient activity of panel B.

static—attractors is to choose G to be an oriented graph with no sinks. The following examples are of this type.

The Gaudi attractor. Figure 8 shows two solutions to a CTLN for a cyclically symmetric tournament[1] graph on $n = 5$ nodes. For some initial conditions, the solutions converge to a somewhat boring limit cycle with the firing rates $x_1(t), \ldots, x_5(t)$ all peaking in the expected sequence, 12345 (bottom middle). For a different set of initial conditions, however, the solution converges to the beautiful and unusual attractor displayed at the top.

Symmetry and synchrony. Because the pattern of weights in a CTLN is completely determined by the graph G, any symmetry of the graph necessarily translates to a symmetry of the differential equations, and hence of the vector field. It follows that the automorphism group of G also acts on the set of all attractors, which must respect the symmetry. For example, in the cyclically symmetric tournament of Figure 8, both the Gaudi attractor and the "boring" limit cycle below it are invariant under the cyclic permutation (12345): the solution is preserved up to a time translation.

Another way for symmetry to manifest itself in an attractor is via synchrony. The network in Figure 9A depicts a CTLN with a graph on $n = 5$ nodes that has a nontrivial automorphism group C_3, cyclically permuting the nodes 2, 3, and 4. In the corresponding attractor, the neurons 2, 3, 4 perfectly synchronize as the solution settles into the limit cycle. Notice, however, what happens for the network in Figure 9B. In this case, the limit cycle looks very similar to the one in A, with the same synchrony among neurons 2, 3, and 4. However, the graph is missing the $4 \to 5$ edge, and so the graph has no nontrivial automorphisms. We refer to this phenomenon as *surprise symmetry*.

On the flip side, a network with graph symmetry may have multiple attractors that are exchanged by the group action, but do not individually respect the symmetry. This

[1] A tournament *is a directed graph in which every pair of nodes has exactly one (directed) edge between them.*

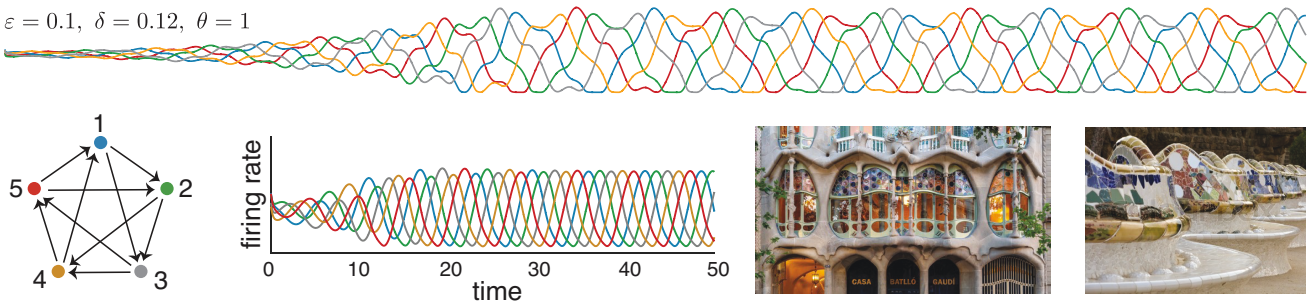


Figure 8. Gaudi attractor. A CTLN for a cyclically symmetric tournament on $n = 5$ nodes produces two distinct attractors, depending on initial conditions. We call the top one the Gaudi attractor because the undulating curves are reminiscent of work by the architect from Barcelona.

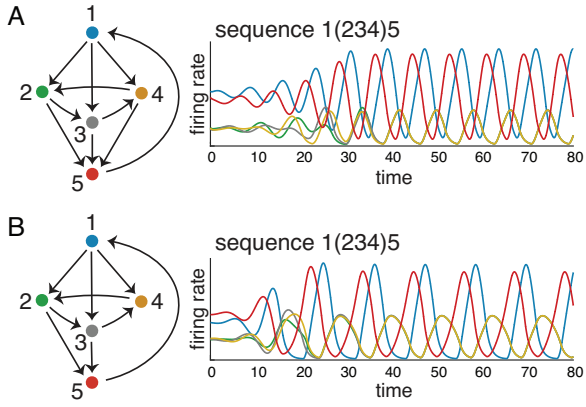


Figure 9. Symmetry and synchrony. (A) A graph with automorphism group C_3 has an attractor where nodes 2, 3, and 4 fire synchronously. (B) The symmetry is broken due to the dropped $4 \to 5$ edge. Nevertheless, the attractor still respects the (234) symmetry with nodes 2, 3, and 4 firing synchronously. Note that both attractors are very similar limit cycles, but the one in B has longer period. (Standard parameters: $\varepsilon = 0.25$, $\delta = 0.5$, $\theta = 1$.)

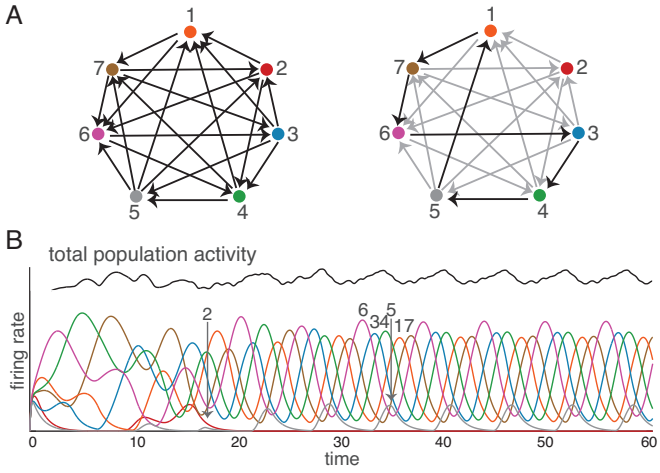


Figure 10. Emergent sequences can be difficult to predict. (A) (Left) The graph of a CTLN that is a tournament on seven nodes. (Right) The same graph, but with the cycle corresponding to the sequential activity highlighted in black. (B) A solution to the CTLN that converges to a limit cycle. This appears to be the only attractor of the network for the standard parameters.

is the more familiar scenario of spontaneous symmetry breaking.

Emergent sequences. One of the most reliable properties of CTLNs is the tendency of neurons to fire in sequence. Although we have seen examples of synchrony, the global inhibition promotes competitive dynamics wherein only one or a few neurons reach their peak firing rates at the same time. The sequences may be intuitive, as in the networks of Figures 8 and 9, following obvious cycles in the graph. However, even for small networks the emergent sequences may be difficult to predict.

The network in Figure 10A has $n = 7$ neurons, and the graph is a tournament with no nontrivial automorphisms. The corresponding CTLN appears to have a single, global attractor, shown in Figure 10B. The neurons in this limit cycle fire in a repeating sequence, 634517, with 5 being the lowest-firing node. This sequence is highlighted in black in the graph, and corresponds to a cycle in the graph. However, it is only one of many cycles in the graph. Why do the dynamics select this sequence and not the others? And why does neuron 2 drop out, while all others persist? This is particularly puzzling given that node 2 has in-degree three, while nodes 3 and 5 have in-degree two.

Indeed, local properties of a network, such as the in- and out-degrees of individual nodes, are insufficient for predicting the participation and ordering of neurons in emergent sequences. Nevertheless, the sequence is fully determined by the structure of G. We just have a limited understanding of how. Recent progress in understanding sequential attractors has relied on special network architectures that are cyclic like the ones in Figure 9 [PLACM22]. Interestingly, although the graph in Figure 10 does not have such an architecture, the induced subgraph generated by the high-firing nodes 1, 3, 4, 6, and 7 is isomorphic to the graph in Figure 8. This graph, as well as the two graphs in Figure 9, have corresponding networks that are in some sense irreducible in their dynamics. These are examples of graphs that we refer to as *core motifs* [PMMC22].

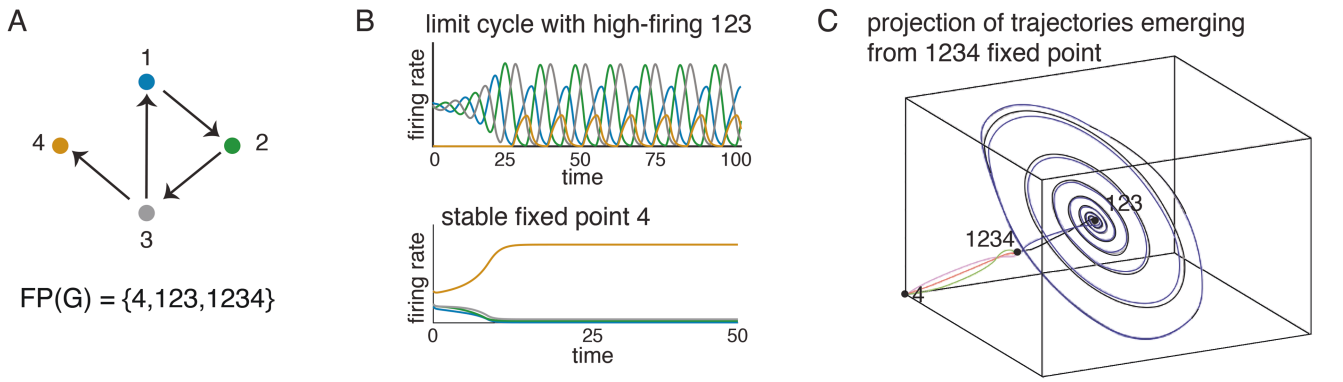


Figure 11. An example CTLN and its attractors. (A) The graph of a CTLN. The fixed point supports are given by $\mathrm{FP}(G) = \{4, 123, 1234\}$, irrespective of parameters $\varepsilon, \delta, \theta$. (B) Solutions to the CTLN in A using the standard parameters $\theta = 1$, $\varepsilon = 0.25$, and $\delta = 0.5$. (Top) The initial condition was chosen as a small perturbation of the fixed point supported on 123. The activity quickly converges to a limit cycle where the high-firing neurons are the ones in the fixed point support. (Bottom) A different initial condition yields a solution that converges to the static attractor corresponding to the stable fixed point on node 4. (C) The three fixed points are depicted in a three-dimensional projection of the four-dimensional state space. Perturbations of the fixed point supported on 1234 produce solutions that either converge to the limit cycle or to the stable fixed point from B.

4. Minimal Fixed Points, Core Motifs, and Attractors

Stable fixed points of a network are of obvious interest because they correspond to static attractors [HSS03, CDI13]. One of the most striking features of CTLNs, however, is the strong connection between *unstable* fixed points and dynamic attractors [PMMC22, PLACM22].

Question 2. For a given CTLN, can we predict the dynamic attractors of the network from its unstable fixed points? Can the unstable fixed points be determined from the structure of the underlying graph G?

Throughout this section, G is a directed graph on n nodes. Subsets $\sigma \subseteq [n]$ are often used to denote both the collection of vertices indexed by σ and the induced subgraph $G|_\sigma$. The corresponding network is assumed to be a CTLN with fixed parameters ε, δ, and θ.

Figure 11 provides an example to illustrate the relationship between unstable fixed points and dynamic attractors. Any CTLN with the graph in panel A has three fixed points, with supports $\mathrm{FP}(G) = \{4, 123, 1234\}$. The collection of fixed point supports can be thought of as a partially ordered set, ordered by inclusion. In our example, 4 and 123 are thus *minimal* fixed point supports, because they are minimal under inclusion. It turns out that the corresponding fixed points each have an associated attractor (Figure 11B). The one supported on 4, a sink in the graph, yields a stable fixed point, while the 123 (unstable) fixed point, whose induced subgraph $G|_{123}$ is a 3-cycle, yields a limit cycle attractor with high-firing neurons 1, 2, and 3. Figure 11C depicts all three fixed points in the state space. Here we can see that the third one, supported on 1234, acts as a "tipping point" on the boundary of two basins of attraction. Initial conditions near this fixed point can yield solutions that converge either to the stable fixed point or the limit cycle.

Not all minimal fixed points have corresponding attractors. In [PMMC22] we saw that the key property of such a $\sigma \in \mathrm{FP}(G)$ is that it be minimal not only in $\mathrm{FP}(G)$ but also in $\mathrm{FP}(G|_\sigma)$, corresponding to the induced subnetwork restricted to the nodes in σ. In other words, σ is the only fixed point in $\mathrm{FP}(G|_\sigma)$. This motivates the definition of core motifs.

Definition 4.1. Let G be the graph of a CTLN on n nodes. An induced subgraph $G|_\sigma$ is a *core motif* of the network if $\mathrm{FP}(G|_\sigma) = \{\sigma\}$.

When the graph G is understood, we sometimes refer to σ itself as a core motif if $G|_\sigma$ is one. The associated fixed point is called a *core fixed point*. Core motifs can be thought of as "irreducible" networks because they have a single fixed point of full support. Since the activity is bounded and must converge to an attractor, the attractor can be said to correspond to this fixed point. A larger network that contains $G|_\sigma$ as an induced subgraph may or may not have $\sigma \in \mathrm{FP}(G)$. When the core fixed point does survive, we refer to the embedded $G|_\sigma$ as a *surviving* core motif, and we expect the associated attractor to survive. In Figure 11, the surviving core motifs are $G|_4$ and $G|_{123}$, and they precisely predict the attractors of the network.

The simplest core motifs are cliques. When these survive inside a network G, the corresponding attractor is always a stable fixed point supported on all nodes of the clique. In fact, we conjectured that any stable fixed point for a CTLN must correspond to a maximal clique of G—specifically, a *target-free* clique [CGM19].

Up to size 4, all core motifs are parameter-independent. For size 5, 37 of 45 core motifs are parameter-independent. Figure 12 shows the complete list of all core motifs of

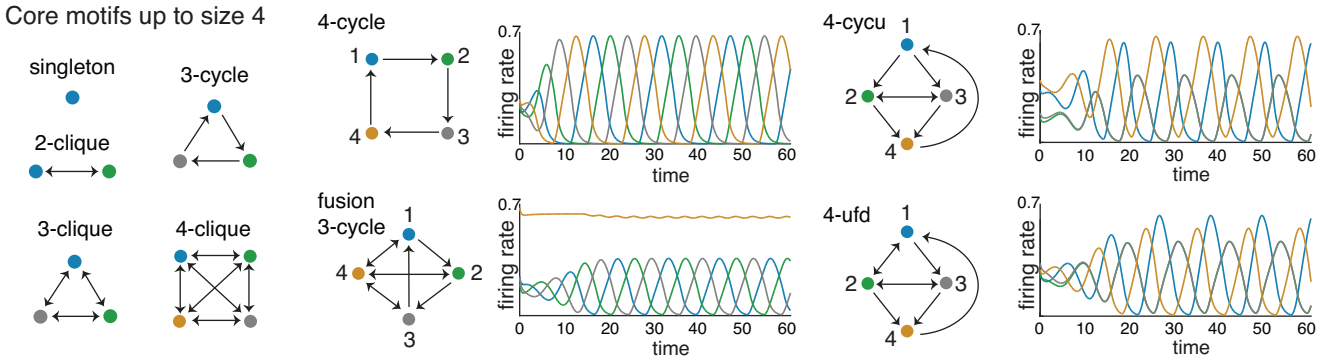


Figure 12. Small core motifs. For each of these graphs, $\mathrm{FP}(G) = \{[n]\}$, where n is the number of nodes. Attractors are shown for CTLNs with the standard parameters $\varepsilon = 0.25$, $\delta = 0.5$, and $\theta = 1$.

size $n \leq 4$, together with some associated attractors. The cliques all correspond to stable fixed points, the simplest type of attractor. The 3-cycle yields the limit cycle attractor in Figure 5, which may be distorted when embedded in a larger network (see Figure 11B). The other core motifs whose fixed points are unstable have dynamic attractors. Note that the 4-cycu graph has a (23) symmetry, and the rate curves for these two neurons are synchronous in the attractor. This synchrony is also evident in the 4-ufd attractor, despite the fact that this graph does not have the (23) symmetry. Perhaps the most interesting attractor, however, is the one for the fusion 3-cycle graph. Here the 123 3-cycle attractor, which does not survive the embedding to the larger graph, appears to "fuse" with the stable fixed point associated to 4 (which also does not survive). The resulting attractor can be thought of as binding together a pair of smaller attractors.

We have performed extensive tests on whether or not core motifs predict attractors in small networks. Specifically, we decomposed all 9608 directed graphs on $n = 5$ nodes into core motif components, and used this to predict the attractors. We found that 1053 of the graphs have surviving core motifs that are not cliques; these graphs were thus expected to support dynamic attractors. The remaining 8555 graphs contain only cliques as surviving core motifs, and were thus expected to have only stable fixed point attractors. Overall, we found that core motifs correctly predicted the set of attractors in 9586 of the 9608 graphs. Of the 22 graphs with mistakes, 19 graphs have a core motif with no corresponding attractor, and 3 graphs have no core motifs for the chosen parameters.[2]

5. Graph Rules

We have seen that CTLNs exhibit a rich variety of nonlinear dynamics, and that the attractors are closely related to the fixed points. This opens up a strategy for linking attractors to the underlying network architecture G via the fixed point supports $\mathrm{FP}(G)$. Our main tools for doing this are *graph rules*.

Throughout this section, we will use greek letters σ, τ, ω to denote subsets of $[n] = \{1, \ldots, n\}$ corresponding to fixed point supports (or potential supports), while latin letters i, j, k, ℓ denote individual nodes/neurons. As before, $G|_\sigma$ denotes the induced subgraph obtained from G by restricting to σ and keeping only edges between vertices of σ. The fixed point supports are:

$$\mathrm{FP}(G) \overset{\text{def}}{=} \{\sigma \subseteq [n] \mid \sigma = \operatorname{supp} x^* \text{ for some fixed pt } x^* \text{ of the associated CTLN}\}.$$

The main question addressed by graph rules is:

Question 3. What can we say about $\mathrm{FP}(G)$ from knowledge of G alone?

For example, consider the graphs in Figure 13. Can we determine from the graph alone which subgraphs will support fixed points? Moreover, can we determine which of those subgraphs are core motifs that will give rise to attractors of the network? We saw in Section 4 (Figure 12) that cycles and cliques are among the small core motifs; can cycles and cliques produce core motifs of any size? Can we identify other graph structures that are relevant for either ruling in or ruling out certain subgraphs as fixed point supports? The rest of Section 5 focuses on addressing these questions.

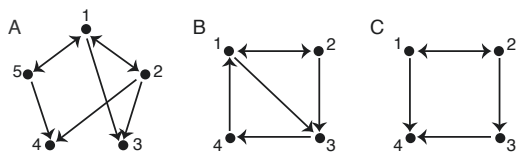


Figure 13. Graphs for which $\mathrm{FP}(G)$ is completely determined by graph rules.

Note that implicit in the above questions is the idea that graph rules are *parameter-independent*: that is, they directly relate the structure of G to $\mathrm{FP}(G)$ via results that are valid

[2] *Classification of CTLNs on n=5 nodes available at* `https://github.com/ccurto/n5-graphs-package`.

for all choices of ε, δ, and θ (provided they lie within the legal range). In order to obtain the most powerful results, we also require that our CTLNs be *nondegenerate*. As has already been noted, nondegeneracy is generically satisfied for TLNs [CGM19]. For CTLNs, it is satisfied irrespective of θ and for almost all legal range choices of ε and δ (i.e., up to a set of measure zero in the two-dimensional parameter space for ε and δ).

5.1. **Examples of graph rules.** We've already seen some graph rules. For example, Theorem 3.1 told us that if G is an oriented graph with no sinks, the associated CTLN has no stable fixed points. Such CTLNs are thus guaranteed to only exhibit dynamic attractors. Here we present a set of eight simple graph rules, all proven in [CGM19], that are easy to understand and give a flavor of the kinds of theorems we have found.

We will use the following graph theoretic terminology. A *source* is a node with no incoming edges, while a *sink* is a node with no outgoing edges. Note that a node can be a source or sink in an induced subgraph $G|_\sigma$, while not being one in G. An *independent set* is a collection of nodes with no edges between them, while a *clique* is a set of nodes that is all-to-all bidirectionally connected. A *cycle* is a graph (or an induced subgraph) where each node has exactly one incoming and one outgoing edge, and they are all connected in a single directed cycle. A *directed acyclic graph* (DAG) is a graph with a topological ordering of vertices such that $i \nrightarrow j$ whenever $i > j$; such a graph does not contain any directed cycles. Finally, a *target* of a graph $G|_\sigma$ is a node k such that $i \to k$ for all $i \in \sigma \setminus \{k\}$. Note that a target may be inside or outside $G|_\sigma$.

Examples of graph rules:

Rule 1 (independent sets): If $G|_\sigma$ is an independent set, then $\sigma \in \mathrm{FP}(G)$ if and only if each $i \in \sigma$ is a sink in G.

Rule 2 (cliques): If $G|_\sigma$ is a clique, then $\sigma \in \mathrm{FP}(G)$ if and only if there is no node k of G, $k \notin \sigma$, such that $i \to k$ for all $i \in \sigma$. In other words, $\sigma \in \mathrm{FP}(G)$ if and only if $G|_\sigma$ is a target-free clique. If $\sigma \in \mathrm{FP}(G)$, the corresponding fixed point is stable.

Rule 3 (cycles): If $G|_\sigma$ is a cycle, then $\sigma \in \mathrm{FP}(G)$ if and only if there is no node k of G, $k \notin \sigma$, such that k receives two or more edges from σ. If $\sigma \in \mathrm{FP}(G)$, the corresponding fixed point is unstable.

Rule 4 (sources): (i) If $G|_\sigma$ contains a source $j \in \sigma$, with $j \to k$ for some $k \in [n]$, then $\sigma \notin \mathrm{FP}(G)$. (ii) Suppose $j \notin \sigma$, but j is a source in G. Then $\sigma \in \mathrm{FP}(G|_{\sigma \cup j}) \Leftrightarrow \sigma \in \mathrm{FP}(G|_\sigma)$.

Rule 5 (targets): (i) If σ has target k, with $k \in \sigma$ and $k \nrightarrow j$ for some $j \in \sigma$ ($j \neq k$), then $\sigma \notin \mathrm{FP}(G|_\sigma)$ and thus $\sigma \notin \mathrm{FP}(G)$. (ii) If σ has target $k \notin \sigma$, then $\sigma \notin \mathrm{FP}(G|_{\sigma \cup k})$ and thus $\sigma \notin \mathrm{FP}(G)$.

Rule 6 (sinks): If G has a sink $s \notin \sigma$, then $\sigma \cup \{s\} \in \mathrm{FP}(G) \Leftrightarrow \sigma \in \mathrm{FP}(G)$.

Rule 7 (DAGs): If G is a directed acyclic graph with sinks $s_1, \ldots, s_\ell$, then $\mathrm{FP}(G) = \{\cup s_i \mid s_i \text{ is a sink in } G\}$, the set of all $2^\ell - 1$ unions of sinks.

Rule 8 (parity): For any G, $|\mathrm{FP}(G)|$ is odd.

In many cases, particularly for small graphs, our graph rules are complete enough that they can be used to fully work out $\mathrm{FP}(G)$. In such cases, $\mathrm{FP}(G)$ is guaranteed to be parameter-independent (since the graph rules do not depend on ε and δ). As an example, consider the graph on $n = 5$ nodes in Figure 13A; we will show that $\mathrm{FP}(G)$ is completely determined by graph rules. Going through the possible subsets σ of different sizes, we find that for $|\sigma| = 1$ only $3, 4 \in \mathrm{FP}(G)$ (as those are the sinks). Using Rules 1, 2, and 4, we see that the only $|\sigma| = 2$ elements in $\mathrm{FP}(G)$ are the clique 15 and the independent set 34. A crucial ingredient for determining the fixed point supports of sizes 3 and 4 is the sinks rule, which guarantees that 135, 145, and 1345 are the only supports of these sizes. Finally, notice that the total number of fixed points up through size $|\sigma| = 4$ is odd. Using Rule 8 (parity), we can thus conclude that there is no fixed point of full support—that is, with $|\sigma| = 5$. It follows that $\mathrm{FP}(G) = \{3, 4, 15, 34, 135, 145, 1345\}$; moreover, this result is parameter-independent because it was determined purely from graph rules.

We leave it as an exercise to use graph rules to show that $\mathrm{FP}(G) = \{134\}$ for the graph in Figure 13B, and $\mathrm{FP}(G) = \{4, 12, 124\}$ for the graph in Figure 13C. For the graph in C, it is necessary to appeal to a more general rule for *uniform in-degree* subgraphs, which we review next.

Rules 1–7, and many more, all emerge as corollaries of more general rules. In the next few subsections, we will introduce the uniform in-degree rule, graphical domination, and simply-embedded subgraphs. These results form part of a collection of *elementary graph rules*, from which all other known graph rules can be derived. A complete list of elementary graph rules can be found in [CM23].

5.2. **Uniform in-degree rule.** It turns out that Rules 1, 2, and 3 (for independent sets, cliques, and cycles) are all corollaries of a single rule for graphs of *uniform in-degree*.

Definition 5.1. We say that $G|_\sigma$ has *uniform in-degree* d if every node $i \in \sigma$ has d incoming edges from within $G|_\sigma$.

Note that an independent set has uniform in-degree $d = 0$, a cycle has uniform in-degree $d = 1$, and an n-clique is uniform in-degree with $d = n-1$. But, in general, uniform in-degree graphs need not be symmetric. For example, the induced subgraph $G|_{145}$ in Figure 13A is uniform in-degree, with $d = 1$.

Theorem 5.2 ([CGM19]). *Let $G|_\sigma$ be an induced subgraph of G with uniform in-degree d. For $k \notin \sigma$, let d_k denote the*

number of edges $i \to k$ for $i \in \sigma$. Then $\sigma \in \mathrm{FP}(G|_\sigma)$, and

$$\sigma \in \mathrm{FP}(G|_{\sigma\cup k}) \;\Leftrightarrow\; d_k \le d.$$

In particular, $\sigma \in \mathrm{FP}(G)$ if and only if there does not exist $k \notin \sigma$ such that $d_k > d$.

Uniform in-degree fixed points are also uniform in value. If $G|_\sigma$ is uniform in-degree d, then the fixed point x^* supported on σ has the same value for all entries in σ [CGM19, Lemma 18]:

$$x^*_i = \frac{\theta}{|\sigma| + \delta(|\sigma| - d - 1) - \varepsilon d} \quad \text{for each } i \in \sigma.$$

Interestingly, $x^*_i = x^*_j$ for all $i, j \in [n]$, even for uniform in-degree graphs that are not symmetric.

5.3. **Graphical domination.** More generally, fixed points can have very different values across neurons. However, there is some level of "graphical balance" that is required of $G|_\sigma$ for any fixed point support σ. For example, if σ contains a pair of neurons j, k that have the property that all neurons sending edges to j also send edges to k, and $j \to k$ but $k \not\to j$, then σ cannot be a fixed point support. This is because k is receiving a strict superset of the inputs to j, and this imbalance rules out their ability to coexist in the same fixed point support. This motivates the following definition.

Definition 5.3. We say that k *graphically dominates* j *with respect to* σ in G if the following three conditions all hold:

1. For each $i \in \sigma \setminus \{j, k\}$, if $i \to j$ then $i \to k$.
2. If $j \in \sigma$, then $j \to k$.
3. If $k \in \sigma$, then $k \not\to j$.

We refer to this as "inside-in" domination if $j, k \in \sigma$ (see Figure 14A). In this case, we must have $j \to k$ and $k \not\to j$. Remaining cases are shown in Figure 14B-D.

What graph rules does domination give us? Intuitively, when inside-in domination is present, the "graphical balance" necessary to support a fixed point is violated, and so $\sigma \notin \mathrm{FP}(G)$. When k outside-in dominates j for $j \in \sigma$, again there is an imbalance, and this time it guarantees that neuron k turns on, since it received all the inputs that were sufficient to turn on neuron j. Thus, there cannot be a fixed point with support σ since node k will violate the off-neuron conditions. We can draw similar conclusions in the other cases of graphical domination as well, as Theorem 5.4 shows. This theorem was originally proven in [CGM19], but a more elementary proof of this result is given in [CM23].

Theorem 5.4 ([CGM19]). *Suppose k graphically dominates j with respect to σ in G. Then the following all hold:*

1. *(inside-in) If $j, k \in \sigma$, then $\sigma \notin \mathrm{FP}(G|_\sigma)$ and thus $\sigma \notin \mathrm{FP}(G)$.*
2. *(outside-in) If $j \in \sigma$, $k \notin \sigma$, then $\sigma \notin \mathrm{FP}(G|_{\sigma\cup k})$ and thus $\sigma \notin \mathrm{FP}(G)$.*

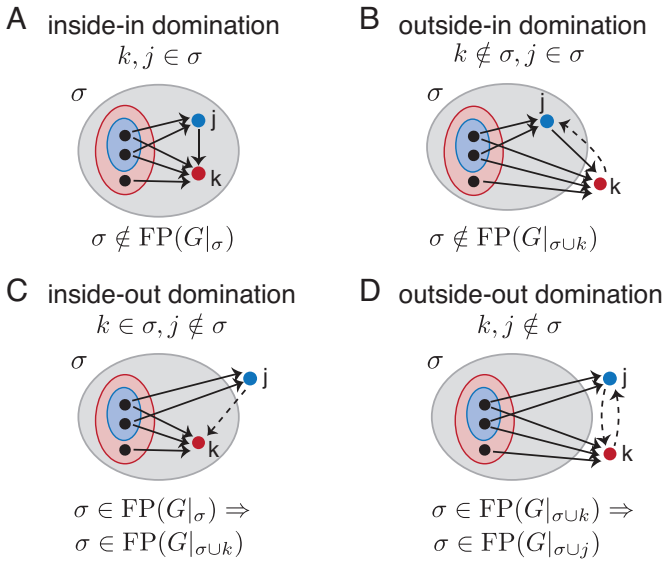


Figure 14. Graphical domination: four cases. In all cases, k graphically dominates j with respect to σ. In particular, the set of vertices of $\sigma \setminus \{j, k\}$ sending edges to k (red ovals) always contains the set of vertices sending edges to j (blue ovals).

3. *(inside-out) If $k \in \sigma$, $j \notin \sigma$, then $\sigma \in \mathrm{FP}(G|_\sigma) \Rightarrow \sigma \in \mathrm{FP}(G|_{\sigma\cup j})$.*
4. *(outside-out) If $j, k \notin \sigma$, then $\sigma \in \mathrm{FP}(G|_{\sigma\cup k}) \Rightarrow \sigma \in \mathrm{FP}(G|_{\sigma\cup j})$.*

To see how this theorem can be used to prove simpler graph rules, consider a graph with a source $j \in \sigma$ that has an edge $j \to k$ for some $k \in [n]$. Since j is a source, it has no incoming edges from within σ. If $k \in \sigma$, then k inside-in dominates j and so $\sigma \notin \mathrm{FP}(G)$. If $k \notin \sigma$, then k outside-in dominates j and again $\sigma \notin \mathrm{FP}(G)$. Rule 4(i) immediately follows. We leave it as an exercise to prove Rules 4(ii), 5(i), 5(ii), and 7.

5.4. **Simply-embedded subgraphs and covers.** Finally, we introduce the concept of simply-embedded subgraphs.

Definition 5.5 (simply-embedded). We say that a subgraph $G|_\tau$ is *simply-embedded in G* if for each $k \notin \tau$, either

(i) $k \to i$ for all $i \in \tau$, or
(ii) $k \not\to i$ for all $i \in \tau$.

In other words, while $G|_\tau$ can have any internal structure, the rest of the network treats all nodes in τ equally (see Figure 15A). By abuse of notation, we sometimes say that the corresponding subset of vertices $\tau \subseteq [n]$ is simply-embedded in G.

We have the following key lemma (see Figure 15B):

Lemma 5.6. *Let $G|_\tau$ be simply-embedded in G. Then for any $\sigma \subseteq [n]$,*

$$\sigma \in \mathrm{FP}(G) \;\Rightarrow\; \sigma \cap \tau \in \mathrm{FP}(G|_\tau) \cup \{\emptyset\}.$$

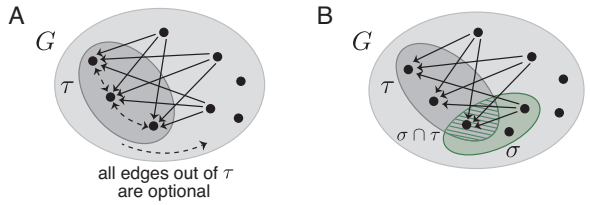


Figure 15. Simply-embedded subgraphs.

What happens if we consider more than one simply-embedded subgraph? It is not difficult to see that intersections of simply-embedded subgraphs are also simply-embedded. However, the union of two simply-embedded subgraphs is only guaranteed to be simply-embedded if the intersection is nonempty. If we have two or more simply-embedded subgraphs, $G|_{\tau_i}$ and $G|_{\tau_j}$, we know that for any $\sigma \in \mathrm{FP}(G)$, σ must restrict to a fixed point $\sigma_i = \sigma \cap \tau_i$ and $\sigma_j = \sigma \cap \tau_j$ in each of those subgraphs. But when can we *glue* together such a $\sigma_i \in \mathrm{FP}(G|_{\tau_i})$ and $\sigma_j \in \mathrm{FP}(G|_{\tau_j})$ to produce a larger fixed point support $\sigma_i \cup \sigma_j$ in $\mathrm{FP}(G|_{\tau_i \cup \tau_j})$?

Lemma 5.7 precisely answers this question. It uses the following notation: $\widehat{\mathrm{FP}}(G) \overset{\text{def}}{=} \mathrm{FP}(G) \cup \{\emptyset\}$.

Lemma 5.7 (pairwise gluing). *Suppose $G|_{\tau_i}, G|_{\tau_j}$ are simply-embedded in G, and consider $\sigma_i \in \widehat{\mathrm{FP}}(G|_{\tau_i})$ and $\sigma_j \in \widehat{\mathrm{FP}}(G|_{\tau_j})$ that satisfy $\sigma_i \cap \tau_j = \sigma_j \cap \tau_i$ (so that σ_i, σ_j agree on the overlap $\tau_i \cap \tau_j$). Then*

$$\sigma_i \cup \sigma_j \in \widehat{\mathrm{FP}}(G|_{\tau_i \cup \tau_j})$$

if and only if one of the following holds:

(i) $\tau_i \cap \tau_j = \emptyset$ *and* $\sigma_i, \sigma_j \in \widehat{\mathrm{FP}}(G|_{\tau_i \cup \tau_j})$, *or*

(ii) $\tau_i \cap \tau_j = \emptyset$ *and* $\sigma_i, \sigma_j \notin \widehat{\mathrm{FP}}(G|_{\tau_i \cup \tau_j})$, *or*

(iii) $\tau_i \cap \tau_j \neq \emptyset$.

6. Gluing Rules

So far we have seen a variety of graph rules and the elementary graph rules from which they are derived. These rules allow us to rule in and rule out potential fixed points in $\mathrm{FP}(G)$ from purely graph-theoretic considerations. In this section, we consider networks whose graph G is composed of smaller induced subgraphs, $G|_{\tau_i}$, for $i \in [N] = \{1, \dots, N\}$. What is the relationship between $\mathrm{FP}(G)$ and the fixed points of the components, $\mathrm{FP}(G|_{\tau_i})$?

It turns out we can obtain nice results if the induced subgraphs $G|_{\tau_i}$ are all simply-embedded in G. In this case, we say that G has a simply-embedded cover.

Definition 6.1 (simply-embedded covers). We say that $\mathcal{U} = \{\tau_1, \dots, \tau_N\}$ is a *simply-embedded cover* of G if each τ_i is simply-embedded in G, and for every vertex $j \in [n]$, there exists an $i \in [N]$ such that $j \in \tau_i$. In other words, the τ_i's are a vertex cover of G. If the τ_i's are all disjoint, we say that $\mathcal{U}$ is a *simply-embedded partition* of G.

In the case that G has a simply-embedded cover, Lemma 5.6 tells us that all "global" fixed point supports in $\mathrm{FP}(G)$ must be unions of "local" fixed point supports in the $\mathrm{FP}(G|_{\tau_i})$, since every $\sigma \in \mathrm{FP}(G)$ restricts to $\sigma \cap \tau_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\}$. But what about the other direction?

Question 4. When does a collection of local fixed point supports $\{\sigma_i\}$, with each nonempty $\sigma_i \in \mathrm{FP}(G|_{\tau_i})$, glue together to form a global fixed point support $\sigma = \cup \sigma_i \in \mathrm{FP}(G)$?

To answer this question, we develop some notions inspired by sheaf theory. For a graph G on n nodes, with a simply-embedded cover $\mathcal{U} = \{\tau_1, \dots, \tau_N\}$, we define the *gluing complex* as:

$$\mathcal{F}_G(\mathcal{U}) \overset{\text{def}}{=} \{\sigma = \cup_i \sigma_i \mid \sigma \neq \emptyset, \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\}, \\ \text{and } \sigma_i \cap \tau_j = \sigma_j \cap \tau_i \text{ for all } i, j \in [N]\}.$$

In other words, $\mathcal{F}_G(\mathcal{U})$ consists of all $\sigma \subseteq [n]$ that can be obtained by gluing together local fixed point supports $\sigma_i \in \mathrm{FP}(G|_{\tau_i})$. Note that in order to guarantee that $\sigma_i = \sigma \cap \tau_i$ for each i, it is necessary that the σ_i's agree on overlaps $\tau_i \cap \tau_j$ (hence the last requirement). This means that $\mathcal{F}_G(\mathcal{U})$ is equivalent to:

$$\mathcal{F}_G(\mathcal{U}) = \{\sigma \neq \emptyset \mid \sigma \cap \tau_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\} \ \forall\, i \in [N]\}.$$

It will also be useful to consider the case where $\sigma \cap \tau_i$ is not allowed to be empty for any i. This is defined as

$$\mathcal{F}^*_G(\mathcal{U}) \overset{\text{def}}{=} \{\sigma \subseteq [n] \mid \sigma \cap \tau_i \in \mathrm{FP}(G|_{\tau_i}) \ \forall\, \tau_i \in \mathcal{U}\}.$$

Translating Lemma 5.6 into the new notation yields the following:

Lemma 6.2. *A CTLN with graph G and simply-embedded cover $\mathcal{U}$ satisfies*

$$\mathrm{FP}(G) \subseteq \mathcal{F}_G(\mathcal{U}).$$

The central question addressed by gluing rules (Question 4) thus translates to: What elements of $\mathcal{F}_G(\mathcal{U})$ are actually in $\mathrm{FP}(G)$?

Our strategy to address this question will be to identify architectures where we can iterate the pairwise gluing rule, Lemma 5.7. Iteration is possible in a simply-embedded cover $\mathcal{U} = \{\tau_i\}$ provided the unions at each step, $\tau_1 \cup \tau_2 \cup \cdots \cup \tau_\ell$, are themselves simply-embedded (this may depend on the order). Fortunately, this is the case for several types of natural constructions, including *disjoint unions* and *clique unions*, which we consider next. It also holds for *connected unions*, which are introduced in [CM23]. Finally, we will examine the case of *cyclic unions*, where pairwise gluing rules cannot be iterated, but for which we find an equally clean characterization of $\mathrm{FP}(G)$.

6.1. **Disjoint, clique, and cyclic unions.** The following graph constructions all arise from simply-embedded *partitions*.

Definition 6.3. Consider a graph G with induced subgraphs $\{G|_{\tau_i}\}$ corresponding to a vertex partition $\mathcal{U} = \{\tau_1, \dots, \tau_N\}$. Then

- G is a *disjoint union* if there are no edges between τ_i and τ_j for $i \neq j$. (See Figure 16A.)
- G is a *clique union* if it contains all possible edges between τ_i and τ_j for $i \neq j$. (See Figure 16B.)
- G is a *cyclic union* if it contains all possible edges from τ_i to τ_{i+1}, for $i = 1, \dots, N-1$, as well as all possible edges from τ_N to τ_1, but no other edges between distinct components τ_i, τ_j. (See Figure 16C.)

Note that in each of these cases, $\mathcal{U}$ is a simply-embedded partition of G.

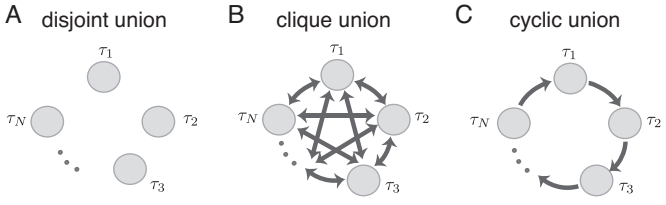


Figure 16. Disjoint unions, clique unions, and cyclic unions. In each architecture, the $\{\tau_i\}$ form a simply-embedded partition of G. Thick edges between components indicate that there are edges between every pair of nodes in the components.

Since the simply-embedded subgraphs in a partition are all disjoint, Lemma 5.7(i-ii) applies. Consequently, fixed point supports $\sigma_i \in \mathrm{FP}(G|_{\tau_i})$ and $\sigma_j \in \mathrm{FP}(G|_{\tau_j})$ will glue together if and only if either σ_i and σ_j both survive to yield fixed points in $\mathrm{FP}(G)$, or neither survives. For both disjoint unions and clique unions, it is easy to see that all larger unions of the form $\tau_1 \cup \tau_2 \cup \cdots \cup \tau_\ell$ are themselves simply-embedded. We can thus iteratively use the pairwise gluing Lemma 5.7. For disjoint unions, Lemma 5.7(i) applies, yielding our first gluing theorem. Recall that $\widehat{\mathrm{FP}}(G) = \mathrm{FP}(G) \cup \{\emptyset\}$.

Theorem 6.4 ([CGM19, Theorem 11]). *If G is a disjoint union of subgraphs $\{G|_{\tau_i}\}_{i=1}^N$, with $\mathcal{U} = \{\tau_i\}_{i=1}^N$, then*

$$\begin{aligned}\mathrm{FP}(G) &= \mathcal{F}_G(\mathcal{U}) \\ &= \{\cup_{i=1}^N \sigma_i \mid \sigma_i \in \widehat{\mathrm{FP}}(G|_{\tau_i}) \,\forall\, i \in [N]\} \setminus \{\emptyset\}.\end{aligned}$$

On the other hand, for clique unions, we must apply Lemma 5.7(ii), which shows that only gluings involving a *nonempty* σ_i from each component are allowed. Hence $\mathrm{FP}(G) = \mathcal{F}_G^*(\mathcal{U})$. Interestingly, the same result holds for cyclic unions, but the proof is different because the simply-embedded structure does *not* get preserved under unions, and hence Lemma 5.7 cannot be iterated. These results are combined in the next theorem.

Theorem 6.5 ([CGM19, Theorems 12 and 13]). *If G is a clique union or a cyclic union of subgraphs $\{G|_{\tau_i}\}_{i=1}^N$, with $\mathcal{U} = \{\tau_i\}_{i=1}^N$, then*

$$\begin{aligned}\mathrm{FP}(G) &= \mathcal{F}_G^*(\mathcal{U}) \\ &= \{\cup_{i=1}^N \sigma_i \mid \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \,\forall\, i \in [N]\}.\end{aligned}$$

We end this section by revisiting core motifs. Recall that core motifs of CTLNs are subgraphs $G|_\sigma$ that support a unique fixed point, which has full-support: $\mathrm{FP}(G|_\sigma) = \{\sigma\}$. We denote the set of core motifs by

$$\mathrm{FP}_{\mathrm{core}}(G) \stackrel{\mathrm{def}}{=} \{\sigma \in \mathrm{FP}(G) \mid G|_\sigma \text{ is a core motif of } G\}.$$

For small CTLNs, we have seen that core motifs are predictive of a network's attractors [PMMC22].

What can gluing rules tell us about core motifs? It turns out that we can precisely characterize these motifs for clique and cyclic unions.

Corollary 6.6. *Let G be a clique union or a cyclic union of components $\tau_1, \dots, \tau_N$. Then*

$$\mathrm{FP}_{\mathrm{core}}(G) = \{\cup_{i=1}^N \sigma_i \mid \sigma_i \in \mathrm{FP}_{\mathrm{core}}(G|_{\tau_i})\}.$$

In particular, G is a core motif if and only if every G_{τ_i} is a core motif.

6.2. **Modeling with cyclic unions.** The power of graph rules is that they enable us to reason mathematically about the graph of a CTLN and make surprisingly accurate predictions about the dynamics. This is particularly true for cyclic unions, where the dynamics consistently appear to traverse the components in cyclic order. Consequently, these architectures are useful for modeling a variety of phenomena that involve sequential attractors. This includes the storage and retrieval of sequential memories, as well as CPGs responsible for rhythmic activity, such as locomotion [YMSL05].

Recall that the attractors of a network tend to correspond to core motifs in $\mathrm{FP}_{\mathrm{core}}(G)$. Using Corollary 6.6, we can easily engineer cyclic unions that have multiple sequential attractors. For example, consider the cyclic union in Figure 17A, with $\mathrm{FP}_{\mathrm{core}}(G)$ comprised of all cycles of length 5 that contain exactly one node per component. For parameters $\varepsilon = 0.75$, $\delta = 4$, the CTLN yields a limit cycle (Figure 17B), corresponding to one such core motif, with sequential firing of a node from each component. By symmetry, there must be an equivalent limit cycle for every choice of five nodes, one from each layer, and thus the network is guaranteed to have m^5 limit cycles. Note that this network architecture, increased to seven layers, could serve as a mechanism for storing phone numbers in working memory ($m = 10$ for digits 0–9).

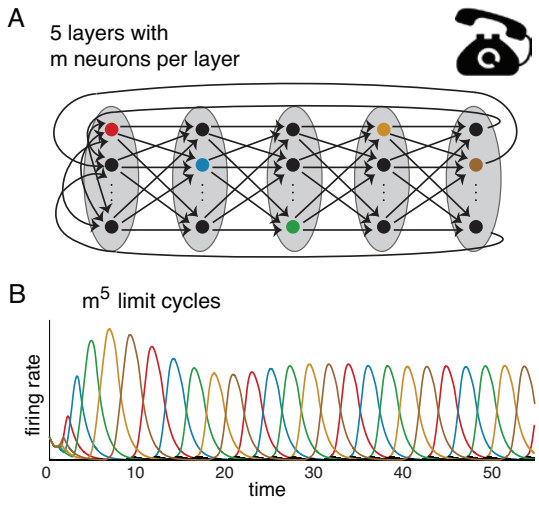


Figure 17. The phone number network. (A) A cyclic union with m neurons per layer (component), and all m^2 feedforward connections from one layer to the next. (B) A limit cycle for the corresponding CTLN (with parameters $\varepsilon = 0.75$, $\delta = 4$).

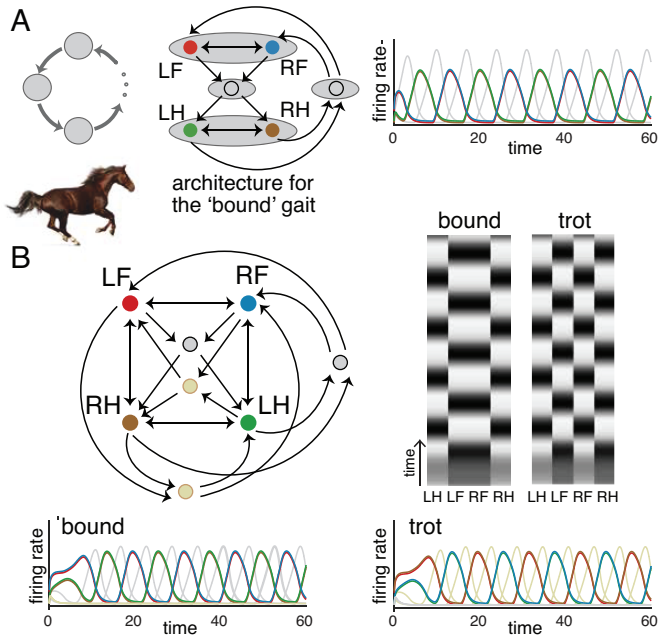


Figure 18. A Central Pattern Generator circuit for quadruped motion. (A) (Left) A cyclic union architecture on six nodes that produces the 'bound' gait. (Right) The limit cycle corresponding to the bound gait. (B) The graph on eight nodes is formed from merging together architectures for the individual gaits, 'bound' and 'trot'. Note that the positions of the two hind legs (LH, RH) are flipped for ease of drawing the graph.

As another application of cyclic unions, consider the graph in Figure 18A which produces the quadruped gait 'bound' (similar to gallop), where we have associated each of the four colored nodes with a leg of the animal. Notice that the clique between pairs of legs ensures that those nodes co-fire, and the cyclic union structure guarantees that the activity flows forward cyclically. A similar network was created for the 'trot' gait, with appropriate pairs of legs joined by cliques.

Figure 18B shows a network in which both the 'bound' and 'trot' gaits can coexist, with the network selecting one pattern (limit cycle) over the other based solely on initial conditions. This network was produced by essentially overlaying the two architectures that would produce the desired gaits, identifying the two graphs along the nodes corresponding to each leg. Notice that within this larger network, the induced subgraphs for each gait are no longer perfect cyclic unions (since they include additional edges between pairs of legs), and are no longer core motifs. And yet the combined network still produces limit cycles that are qualitatively similar to those of the isolated cyclic unions for each gait. It is an open question when this type of merging procedure for cyclic unions (or other types of subnetworks) will preserve the original limit cycles within the larger network.

7. Conclusions

Recurrent network models such as TLNs have historically played an important role in theoretical neuroscience; they give mathematical grounding to key ideas about neural dynamics and connectivity, and provide concrete examples of networks that encode multiple attractors. These attractors represent the possible responses, e.g., stored memory patterns, of the network.

In the case of CTLNs, we have been able to prove a variety of results, such as graph rules, about the fixed point supports FP(G)—yielding valuable insights into the attractor dynamics. Many of these results can be extended beyond CTLNs to more general families of TLNs, and potentially to other threshold nonlinearities. The reason lies in the combinatorial geometry of the hyperplane arrangements. In addition to the arrangements discussed in Section 2, there are closely related hyperplane arrangements given by the *nullclines* of TLNs, defined by $dx_i/dt = 0$ for each i. It is easy to see that fixed points correspond to intersections of nullclines, and thus the elements of FP(W, b) are completely determined by the combinatorial geometry of the nullcline arrangement. Intuitively, the combinatorial geometry of such an arrangement is preserved under small perturbations of W and b. This allows us to extend CTLN results and study how FP(W, b) changes as we vary the TLN parameters W_{ij} and b_i. These ideas, including connections to oriented matroids, are further developed in [CM23] and references therein. [CM23] also lays out a number of open questions on graph rules, gluing rules, core motifs, and the relationship between fixed points and attractors.

ACKNOWLEDGMENTS. We would like to thank Zelong Li, Nicole Sanderson, and Juliana Londono Alvarez for a careful reading of the manuscript. We also thank Caitlyn Parmelee, Caitlin Lienkaemper, Safaan Sadiq, Anda Degeratu, Vladimir Itskov, Christopher Langdon, Jesse Geneson, Daniela Egas Santander, Stefania Ebli, Alice Patania, Joshua Paik, Samantha Moore, Devon Olds, and Joaquin Castañeda for many useful discussions.

References

[BCRR21] Andrea Bel, Romina Cobiaga, Walter Reartes, and Horacio G. Rotstein, *Periodic solutions in threshold-linear networks and their entrainment*, SIAM J. Appl. Dyn. Syst. **20** (2021), no. 3, 1177–1208, DOI 10.1137/20M1337831. MR4279921

[BF22] T. Biswas and J. E. Fitzgerald, *Geometric framework to predict structure from function in neural networks*, Phys. Rev. Research, **4** (2022).

[CG83] Michael A. Cohen and Stephen Grossberg, *Absolute stability of global pattern formation and parallel memory storage by competitive neural networks*, IEEE Trans. Systems Man Cybernet. **13** (1983), no. 5, 815–826, DOI 10.1016/S0166-4115(08)60913-9. MR730500

[CDI13] Carina Curto, Anda Degeratu, and Vladimir Itskov, *Encoding binary neural codes in networks of threshold-linear neurons*, Neural Comput. **25** (2013), no. 11, 2858–2903, DOI 10.1162/NECO_a_00504. MR3136636

[CGM19] Carina Curto, Jesse Geneson, and Katherine Morrison, *Fixed points of competitive threshold-linear networks*, Neural Comput. **31** (2019), no. 1, 94–155, DOI 10.1162/neco_a_01151. MR3898981

[CM16] Carina Curto and Katherine Morrison, *Pattern completion in symmetric threshold-linear networks*, Neural Comput. **28** (2016), no. 12, 2825–2852, DOI 10.1162/neco_a_00869. MR3866422

[CM23] C. Curto and K. Morrison, *Graph rules for recurrent neural network dynamics: extended version*, Available at `https://arxiv.org/abs/2301.12638`, 2023.

[DA01] Peter Dayan and L. F. Abbott, *Theoretical neuroscience*, Computational Neuroscience, MIT Press, Cambridge, MA, 2001. Computational and mathematical modeling of neural systems. MR1985615

[HSM+00] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*, Nature, **405** (2000), 947–951.

[HSS03] R. H. Hahnloser, H. S. Seung, and J. J. Slotine, *Permitted and forbidden sets in symmetric threshold-linear networks*, Neural Comput., **15** (2003), no. 3, 621–638.

[HR58] H. K. Hartline and F. Ratliff, *Spatial summation of inhibitory influence in the eye of limulus and the mutual interaction of receptor units*, J. Gen. Physiol., **41** (1958), 1049–1066.

[Hop82] J. J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proc. Nat. Acad. Sci. U.S.A. **79** (1982), no. 8, 2554–2558, DOI 10.1073/pnas.79.8.2554. MR652033

[KAY14] M. M. Karnani, M. Agetsuma, and R. Yuste, *A blanket of inhibition: functional inferences from dense inhibitory connectivity*, Curr Opin Neurobiol, **26** (2014), 96–102.

[LBH09] A. Luczak, P. Barthó, and K.D. Harris, *Spontaneous events outline the realm of possible sensory responses in neocortical populations*, Neuron, **62** (2009), no. 3, 13–425

[MDIC16] K. Morrison, A. Degeratu, V. Itskov, and C. Curto, *Diversity of emergent dynamics in competitive threshold-linear networks*, Preprint, `arXiv:1605.04463`, 2022.

[PLACM22] Caitlyn Parmelee, Juliana Londono Alvarez, Carina Curto, and Katherine Morrison, *Sequential attractors in combinatorial threshold-linear networks*, SIAM J. Appl. Dyn. Syst. **21** (2022), no. 2, 1597–1630, DOI 10.1137/21M1445120. MR4444287

[PMMC22] C. Parmelee, S. Moore, K. Morrison, and C. Curto, *Core motifs predict dynamic attractors in combinatorial threshold-linear networks*, PLOS ONE, 2022.

[SY12] H. S. Seung and R. Yuste, *Principles of Neural Science*, McGraw-Hill Education/Medical, 5th edition, 2012.

[TSSM97] M. Tsodyks, W. Skaggs, T. Sejnowski, and B. McNaughton, *Paradoxical effects of external modulation of inhibitory interneurons*, Journal of Neuroscience, **17** (1997), no. 11, 4382–4388.

[YMSL05] R. Yuste, J. N. MacLean, J. Smith, and A. Lansner, *The cortex as a central pattern generator*, Nat. Rev. Neurosci., **6** (2005), 477–483.

Carina Curto

Katherine Morrison

Credits

All figures, including the opener, are courtesy of the authors.

Figures 6, 11, and 12 previously appeared in [PMMC22]. CC-By 2.0.

Photo of Carina Curto is courtesy of Carina Curto.

Photo of Katherine Morrison is courtesy of Woody Myers.