SIAM J. APPLIED DYNAMICAL SYSTEMS
Vol. 21, No. 2, pp. 1597–1630

Sequential Attractors in Combinatorial Threshold-Linear Networks*

Caitlyn Parmelee[†], Juliana Londono Alvarez[‡], Carina Curto[‡], and Katherine Morrison[§]

Abstract. Sequences of neural activity arise in many brain areas, including cortex, hippocampus, and central pattern generator circuits that underlie rhythmic behaviors like locomotion. While network architectures supporting sequence generation vary considerably, a common feature is an abundance of inhibition. In this work, we focus on architectures that support sequential activity in recurrently connected networks with inhibition-dominated dynamics. Specifically, we study emergent sequences in a special family of threshold-linear networks, called combinatorial threshold-linear networks (CTLNs), whose connectivity matrices are defined from directed graphs. Such networks naturally give rise to an abundance of sequences whose dynamics are tightly connected to the underlying graph. We find that architectures based on generalizations of cycle graphs produce limit cycle attractors that can be activated to generate transient or persistent (repeating) sequences. Each architecture type gives rise to an infinite family of graphs that can be built from arbitrary component subgraphs. Moreover, we prove a number of *graph rules* for the corresponding CTLNs in each family. The graph rules allow us to strongly constrain, and in some cases fully determine, the fixed points of the network in terms of the fixed points of the component subnetworks. Finally, we also show how the structure of certain architectures gives insight into the sequential dynamics of the corresponding attractor.

Key words. threshold-linear networks, attractor dynamics, network architectures, neuronal sequences

AMS subject classifications. 92C20, 34A34

DOI. 10.1137/21M1445120

1. Introduction.

Sequences of neural activity arise in many brain areas, including cortex [26, 7, 37], hippocampus [33, 31, 24], and central pattern generator circuits that underlie rhythmic behaviors like locomotion [27, 19]. Moreover, fast sequences during sharp wave ripple events in hippocampus are believed to be critical for memory processing and cortico-hippocampal communication [8, 18, 16]. Such sequences are examples of emergent or *internally generated activity*, that is, neural activity that is shaped primarily by the structure of a recurrent network rather than inherited from a changing external input. A fundamental question is to understand how a network's connectivity shapes neural activity, and what types of network architectures underlie emergent sequences.

Inhibition has long been viewed as a key component of sequence generation in central pattern generators. It also plays an important role in generating rhythmic and sequential

[*] Received by the editors September 8, 2021; accepted for publication (in revised form) by K. Josic February 1, 2022; published electronically June 24, 2022. The third and fourth authors contributed equally to this work.
https://doi.org/10.1137/21M1445120

Funding: The work of the third author was partially supported by NIH R01 NS120581 and NSF DMS-1951165. The work of the third and fourth authors was partially supported by NIH R01 EB022862. The work of the fourth author was partially supported by NSF DMS-1951599.

[†] Keene State College, Keene, NH 03431 USA (caitlyn.parmelee@keene.edu).
[‡] Pennsylvania State University, University Park, PA 16802 USA (jbl5958@psu.edu, ccurto@psu.edu).
[§] University of Northern Colorado, Greeley, CO 80639 USA (katherine.morrison@unco.edu).

activity in cortex and hippocampus [37, 27, 6, 20, 17, 25, 2]. Roughly speaking, inhibition creates competition among neurons, resulting in a tendency for neurons to take turns reaching peak activity levels and thus to fire in sequence. In particular, inhibition-dominated networks exhibit emergent sequences even in the absence of an obvious chain-like architecture, such as a synfire chain [35, 28]. In this work, we analyze a variety of network architectures that give rise to sequential neural activity in a simple nonlinear model of recurrent networks with inhibition-dominated dynamics.

Mathematical setup. We study sequential dynamics in a family of threshold-linear networks (TLNs). The firing rates $x_1(t), \dots, x_n(t)$ of n recurrently connected neurons evolve in time according to the standard TLN equations:

$$\frac{dx_i}{dt} = -x_i + \left[\sum_{j=1}^{n} W_{ij}x_j + b_i\right]_+, \quad i = 1, \dots, n, \tag{1}$$

where $[\cdot]_+ = \max\{0, \cdot\}$ is the threshold nonlinearity. A given TLN is specified by the choice of a connection strength matrix W and a vector of external inputs $b \in \mathbb{R}^n$. TLNs have been widely used in computational neuroscience as a framework for modeling recurrent neural networks, including associative memory networks [34, 23, 22, 36, 14, 13, 12, 5].

In order to investigate how network architectures support sequential dynamics, we consider the special family of *combinatorial threshold-linear networks* (CTLNs). These are inhibition-dominated TLNs where the matrix $W = W(G, \varepsilon, \delta)$ is determined by a simple[1] directed graph G, as follows:

$$W_{ij} = \begin{cases} 0 & \text{if } i = j, \\ -1 + \varepsilon & \text{if } j \to i \text{ in } G, \\ -1 - \delta & \text{if } j \not\to i \text{ in } G. \end{cases} \tag{2}$$

Note that $j \to i$ indicates the presence of an edge from j to i in the graph G, while $j \not\to i$ indicates the absence of such an edge. Additionally, CTLNs typically have a constant external input $b_i = \theta$ in order to ensure the dynamics are internally generated and not inherited from a changing or spatially heterogeneous input. We require the three parameters to satisfy $\theta > 0$, $\delta > 0$, and $0 < \varepsilon < \frac{\delta}{\delta+1}$; when these conditions are met, we say that the parameters are within the *legal range*.[2] Note that the upper bound on ε implies $\varepsilon < 1$, and so the W matrix is always effectively inhibitory.

One of the most striking features of CTLNs is the strong connection between dynamic attractors and unstable fixed points [29, 32]. A fixed point x^* of a TLN is a solution that satisfies $dx_i/dt|_{x=x^*} = 0$ for each $i \in [n]$. The *support* of a fixed point is the subset of active neurons, $\operatorname{supp} x = \{i \mid x_i > 0\}$. For a given network, there can be at most one fixed point per support. Thus, we can label all the fixed points of a network by their support, $\sigma = \operatorname{supp} x^* \subseteq [n]$, where $[n] \overset{\text{def}}{=} \{1, \dots, n\}$. We denote this collection of supports by

[1] A graph is *simple* if it does not have self-loops or multiple edges (in the same direction) between a pair of nodes.

[2] The upper bound on ε is motivated by a theorem in [30]. It ensures that subgraphs consisting of a single directed edge $i \to j$ are not allowed to support stable fixed points.

$$\text{FP}(G) = \text{FP}(G, \varepsilon, \delta) \overset{\text{def}}{=} \{\sigma \subseteq [n] \mid \sigma \text{ is a fixed point support of } W(G, \varepsilon, \delta)\}.$$

In prior work, a series of *graph rules* were proven that can be used to determine fixed points of a CTLN by analyzing the structure of the graph G [11, 10]. These rules are all independent of the choice of parameters ε, δ, and θ.

Sequences from limit cycles. Limit cycles are dynamic attractors corresponding to periodic solutions. A *sequential* limit cycle produces a repeating sequence of neural activations. Limit cycles thus provide a basic mechanism for generating sequences in the context of attractor neural networks.

It is easy to see computationally that a CTLN corresponding to a cyclic graph produces a sequential attractor. Figures 1A–C show limit cycles corresponding to the graph G being a 3-cycle (panel A), a 4-cycle (panel B), or a 5-cycle (panel C). In each case, the solution exhibits a sequence of peak activations that matches the order of neurons in the cycle of the graph. Note that although all connections are effectively inhibitory, the activity appears to follow the edges in the graph. A rigorous proof for the existence of these limit cycles was given in [4].

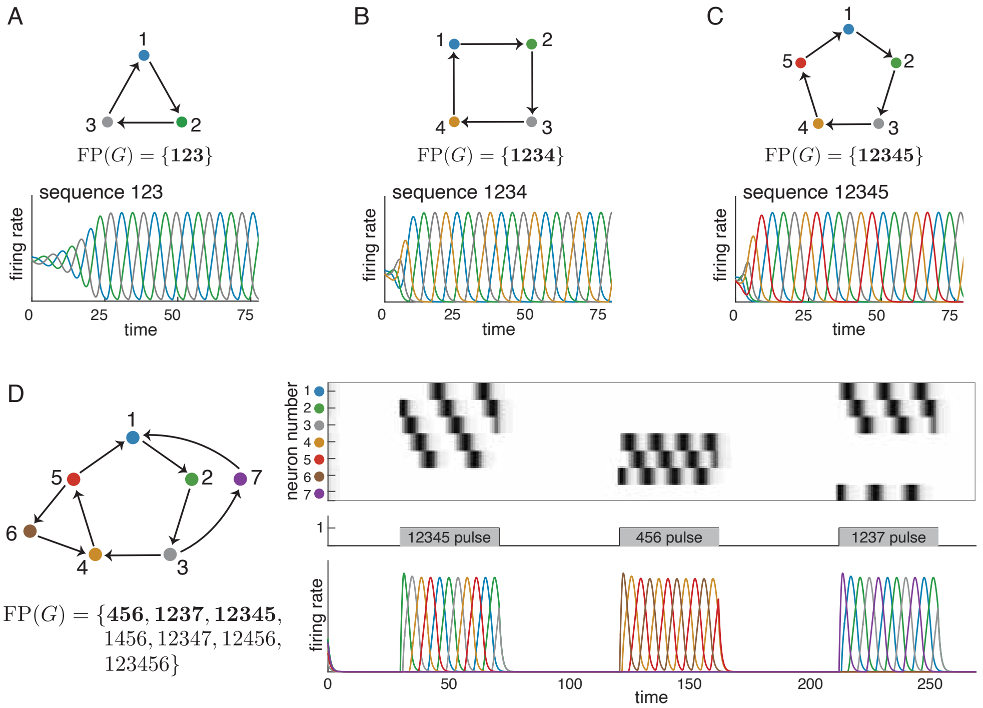


Figure 1. *Sequential attractors from cycle graphs.* (A)–(C) *CTLNs corresponding to a* 3*-cycle, a* 4*-cycle, and a* 5*-cycle each produce a limit cycle where the neurons reach their peak activations in the expected sequence. Colored curves correspond to solutions* $x_i(t)$ *for matching node* i *in the graph.* (D) *Attractors corresponding to the embedded* 3*-cycle,* 4*-cycle, and* 5*-cycle of the network are transiently activated to produce sequences matching those of the isolated cycle networks in* (A)–(C)*. For each network in* (A)–(D), FP(G) *is shown, with the minimal fixed points bolded. To simplify notation for* FP(G)*, we denote a subset* $\{i_1, \dots, i_k\}$ *by* $i_1 \cdots i_k$*. For example,* 12345 *denotes the set* $\{1, 2, 3, 4, 5\}$*. Unless otherwise noted (as in section* 4*), all simulations have CTLN parameters* $\varepsilon = 0.25, \delta = 0.5$*, and* $\theta = 1$*.*

To obtain shorter sequences, these attractors may be transiently activated by an external drive that is time dependent. Figure 1D shows the solution for a CTLN with a graph on seven neurons (left). Here we have chosen $\theta = 0$ as a baseline, with step function pulses of $\theta_i = 1$ for different subsets of neurons. A single simulation is shown, with localized pulses activating the 5-cycle, the 3-cycle, and finally the 4-cycle. Although these cycles overlap, each pulse activates a sequence involving only the neurons in the stimulated subnetwork. Depending on the duration of the pulse, the sequence may play only once or repeat two or more times.

Notice that the minimal fixed points of the network in Figure 1D reflect the subsets of neurons active in the attractors. In related work, we have seen a close correspondence between certain minimal fixed points, called *core motifs*, and the attractors of a network [32]. Thus, $\text{FP}(G)$ is often predictive of limit cycles and other dynamic attractors of a network.

The above mechanism for sequence generation differs from that observed in synfire chains [1, 3, 21] where neural activity flows through a feedforward network, transiently activating neurons in sequence. In section 2.3, we provide a generalization of synfire chain structure, known as *directional chains*, that allow for some local recurrence while still yielding sequences from their transient activity. But the primary focus of this work is on architectures that support sequential attractors, such as limit cycles, with transient sequences emerging from transient activation of these networks.

Graphs that are cycles were the most obvious candidate to produce sequential attractors. But not all CTLN attractors are limit cycles, and not all limit cycles generate sequences. What other architectures can support sequential attractors? This is the main question we address in this paper. We investigate four architectures that generalize the cyclic structure of graphs that are cycles. These are cyclic unions, directional cycles, simply-embedded partitions, and simply-embedded directional cycles. A common feature of all these architectures is that the neurons of the network are partitioned into components $\tau_1, \ldots, \tau_N$, organized in a cyclic manner, whose disjoint union equals the full set of neurons $[n] \overset{\text{def}}{=} \{1, \ldots, n\}$. The induced subgraphs $G|_{\tau_i}$ are called component subgraphs. We will prove a series of theorems about these architectures connecting the fixed points of a graph G to the fixed points of the component subgraphs $G|_{\tau_i}$. As shown in [32], there is a striking correspondence between certain unstable fixed points of a network and its dynamic attractors. Our theorems about the fixed points thus provide valuable insight into the dynamics associated to these network architectures.

Cyclic unions. The most straightforward generalization of a cycle is the *cyclic union*, an architecture first introduced in [11]. Given a set of component subgraphs $G|_{\tau_1}, \ldots, G|_{\tau_N}$, on subsets of nodes $\tau_1, \ldots, \tau_N$, the *cyclic union* is constructed by connecting these subgraphs in a cyclic fashion so that there are edges forward from every node in τ_i to every node in τ_{i+1} (cyclically identifying τ_N with τ_0), and there are no other edges between components (see Figure 2A).

The top graphs in Figures 2B–D are examples of cyclic unions with three components. All the nodes at a given height comprise a τ_i component, and we see that there are edges forward from every node in one component to each node in the next one. Next to each graph is a solution to a corresponding CTLN, which is a global attractor of the network. Note that the activity traverses the components in cyclic order. Cyclic unions are particularly well-behaved architectures where the fixed point supports can be fully characterized in terms of

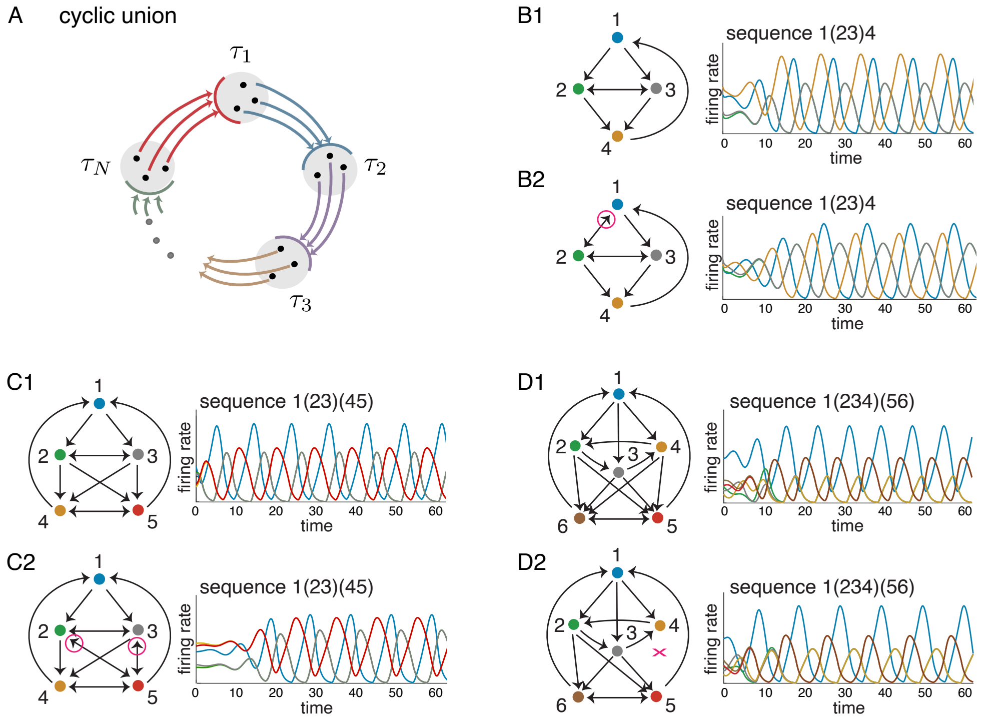


Figure 2. *Cyclic unions and related variations.* (A) *A cyclic union has component subgraphs with subsets of nodes $\tau_1, \ldots, \tau_N$, organized in a cyclic manner. While edges within each $G|_{\tau_i}$ can be arbitrary, edges between components are determined as follows: every node in τ_i sends an edge to every node τ_{i+1}, with τ_N sending edges to τ_1.* (B1), (C1), (D1) *Three cyclic unions with firing rate plots showing solutions to a corresponding CTLN. Above each solution the associated sequence of firing rate peaks is given, with synchronously firing neurons denoted by parentheses.* (B2), (C2), (D2) *These graphs are all variations on the cyclic unions above them, with some edges added or dropped (highlighted in magenta). Solutions of the corresponding CLTNs qualitatively match the solutions of the corresponding cyclic unions. In each case, the sequence is identical.*

those of the components. Specifically, in [11] it was shown that the fixed points of a cyclic union G are precisely the unions of supports of the component subgraphs, exactly one per component.

Theorem 1.1 (cyclic unions, Theorem 13 in [11]). *Let G be a cyclic union of component subgraphs $G|_{\tau_1}, \ldots, G|_{\tau_N}$. For any $\sigma \subseteq [n]$, let $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$. Then*

$$\sigma \in \text{FP}(G) \quad \Leftrightarrow \quad \sigma_i \in \text{FP}(G|_{\tau_i}) \quad \textit{for all } i \in [N].$$

The bottom graphs in Figures 2B–D have very similar dynamics to the ones above them, but do not have a perfect cyclic union structure (each graph has some added back edges or dropped forward edges highlighted in magenta). Despite deviations from the cyclic union architecture, these graphs produce sequential dynamics that similarly traverse the components in cyclic order. In fact, they are examples of a more general class of architectures: directional cycles.

Directional cycles. In a cyclic union, if we restrict to the subnetwork consisting of a pair of consecutive components, $G|_{\tau_i \cup \tau_{i+1}}$, we find that activity initialized on τ_i flows forward and ends up concentrated on τ_{i+1}. Thus, there appears to be a directionality to the flow $\tau_i \to \tau_{i+1}$. Moreover, the fixed points of $G|_{\tau_i \cup \tau_{i+1}}$ are all confined to live in τ_{i+1}, and so the concentration of neural activity coincides with the subnetwork supporting the fixed points. This is a phenomenon we have observed more generally that motivates us to define *directional graphs.*

We say that a graph is *directional* whenever we have $\text{FP}(G) \subseteq \text{FP}(G|_\tau)$ for some $\tau \subsetneq [n]$. In this case, we denote the complementary set as $\omega = [n] \setminus \tau$ and say that G has direction $\omega \to \tau$. We additionally require a more technical condition that allows us to prove that certain natural compositions, like chaining directional graphs together, produce a new directional graph (see Definition 2.3 for the full definition). In simulations, we have seen that directional graphs have the desired directionality of neural activity, so that activity initialized on ω will flow through the network and become concentrated on the nodes of τ.

Note that while we predict that directional graphs have feedforward dynamics, they need not have a feedforward architecture. In Figure 2, each subgraph consisting of a pair of consecutive components is directional. For example, the subgraph $G|_{\{1,2,3\}}$ in B2 is directional with direction $\{1\} \to \{2,3\}$, so that activity initialized on node 1 tends to flow forward to nodes 2 and 3, despite the presence of the back edge $2 \to 1$. Similarly, in C2, the subgraph $G|_{\{2,3,4,5\}}$ is directional with direction $\{2,3\} \to \{4,5\}$ despite the back edges $5 \to 2$ and $5 \to 3$. The subgraph $G|_{\{2,3,4,5,6\}}$ in D2 is also directional with direction $\{2,3,4\} \to \{5,6\}$. Note that it is not necessary to have edges forward from every node in ω to every node in τ.

With this broader notion of directional graph, we obtain our first generalization of cyclic unions, known as *directional cycles.* We define a *directional cycle* as a graph with a partition of its nodes such that each $G|_{\tau_i \cup \tau_{i+1}}$ is directional with direction $\tau_i \to \tau_{i+1}$ (cyclically identifying τ_N with τ_0). We predict that these graphs will have a cyclic flow to their dynamics, hitting each τ_i component in cyclic order. Figures 2B–D (bottom) give examples of directional cycles and their dynamics, as do Figures 3B,D. While we have not been able to explicitly prove this property of the dynamics, we can prove that all the fixed point supports have such a cyclic structure.

Theorem 1.2 (cyclic fixed points of directional cycles). *Let G be a directional cycle with components $\tau_1, \ldots, \tau_N$. Then for any $\sigma \in \text{FP}(G)$, the graph $G|_\sigma$ contains an undirected cycle*[3] *that intersects every τ_i in cyclic order.*

Observe that unlike the case of cyclic unions, in directional cycles we do not have the property that fixed points σ of the full network restrict to fixed points $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$ of the component subnetworks $G|_{\tau_i}$. For example, in Figure 3B, $\sigma = \{1,3,5\} \in \text{FP}(G_2)$, but $\sigma_2 = \{3\} \notin \text{FP}(G_2|_{\tau_2})$, since the only fixed point of $G_2|_{\tau_2}$ is the full support $\{2,3,4\}$. But Theorem 1.2 does guarantee that $\sigma_i \neq \emptyset$ for all $i \in [N]$. It turns out that there is a key structural property of cyclic unions that guarantees the fixed points are unions of component graph fixed points: such networks have what we call a *simply-embedded partition.*

[3] An *undirected cycle* is a sequence of nodes connected by edges that form a cycle within the underlying undirected graph, in which the direction of edges is simply ignored.

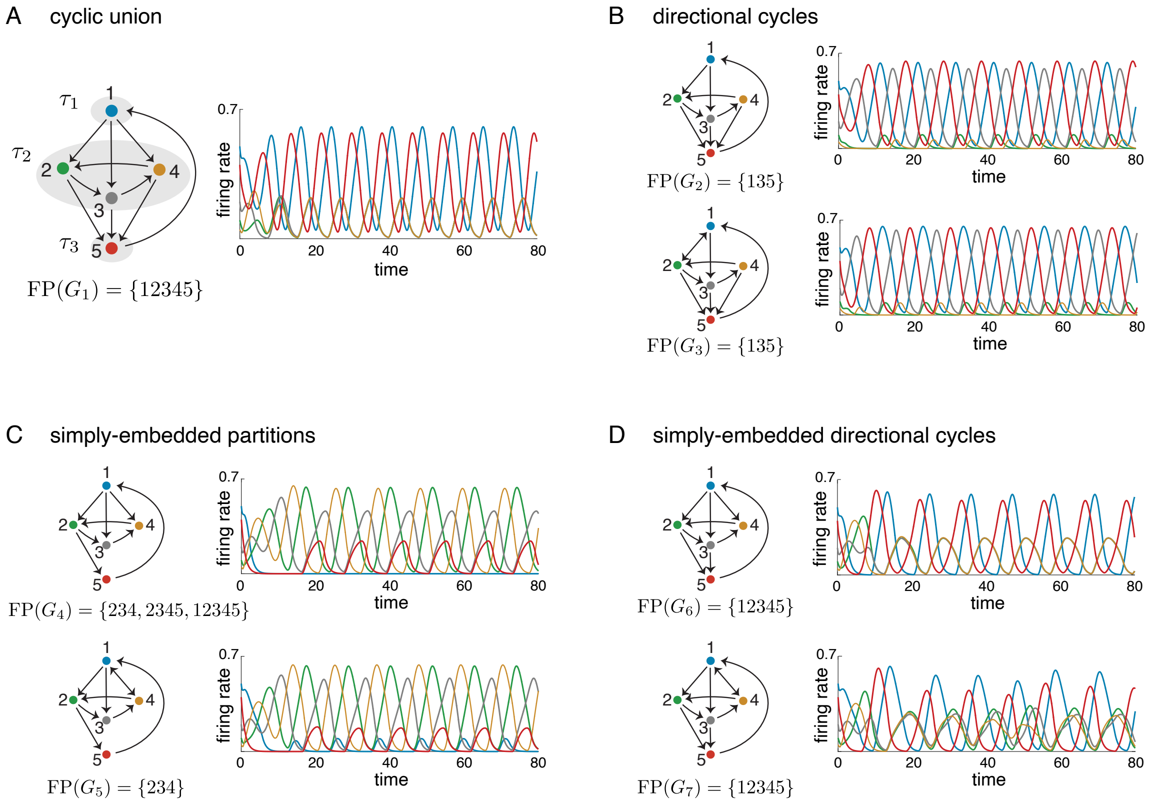


Figure 3. *Example graphs generalizing cyclic union structure.* (A) *A cyclic union of component subgraphs* $G|_{\tau_1}, \ldots, G|_{\tau_3}$ *together with its* $\mathrm{FP}(G)$ *and the global attractor of its corresponding CTLN.* (B)–(D) *Different generalizations of the cyclic union structure of the graph in* A. *Each graph has the same component subgraphs* $G|_{\tau_1}, \ldots, G|_{\tau_3}$, *but different conditions on the edges between these components. For each graph,* $\mathrm{FP}(G)$ *and the global attractor are shown.*

Simply-embedded partitions. The notion of simply added splits was introduced in [11], where it was shown that fixed points behave particularly nicely in networks that have this structure. Given a graph G and a partition of its nodes into two components, $\{\omega|\tau\}$, we say that ω is *simply added* onto τ if for each $k \notin \omega$, either $k \to j$ for all $j \in \tau$ or $k \not\to j$ for all $j \in \tau$. In this case, we say that τ is *simply-embedded* in G. Here we introduce the more general notion of a simply-embedded partition. The key idea is that all nodes in a component τ_i are simply-embedded and thus treated identically by each node outside that component.

Definition 1.3 (simply-embedded partition). *Given a graph* G, *a partition of its nodes* $\{\tau_1|\cdots|\tau_N\}$ *is called a* simply-embedded partition *if every* τ_i *is simply-embedded in* G. *In other words, for each* τ_i *and each* $k \notin \tau_i$, *either* $k \to j$ *for all* $j \in \tau_i$ *or* $k \not\to j$ *for all* $j \in \tau_i$.

For the pair of graphs in Figure 3C, $\{1\,|\,2, 3, 4\,|\,5\}$ is a simply-embedded partition: in each graph, the nodes $2, 3,$ and 4 receive identical inputs from node 1 as well as from node 5. For singleton components, the simply-embedded partition does not impose any constraints. It turns out that this simply-embedded partition structure is sufficient to guarantee that all the fixed points of G restrict to fixed points of the component subgraphs. This means that the fixed points of the components provide a kind of "menu" from which the fixed points of G are made: each fixed point of the full network has support that is the union of component fixed point supports.

Theorem 1.4 ($\mathrm{FP}(G)$ menu for simply-embedded partitions). *Let G have a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$. For any $\sigma \subseteq [n]$, let $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$. Then*

$$\sigma \in \mathrm{FP}(G) \quad \Rightarrow \quad \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\} \quad \text{for all } i \in [N].$$

In other words, every fixed point support of G is a union of component fixed point supports σ_i, at most one per component.

Simply-embedded directional cycles. While the simply-embedded partition generalizes one key property of $\mathrm{FP}(G)$ from cyclic unions, it does not guarantee that every fixed point intersects every component, nor does it guarantee the cyclic flow of the dynamics through the components, as we see in Figure 3C. But combining Theorems 1.2 and 1.4, we immediately see that a *simply-embedded directional cycle* has the desired fixed point properties while maintaining cyclic dynamics.

Theorem 1.5 (simply-embedded directional cycles). *Let G be a directional cycle whose components form a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$. For any $\sigma \subseteq [n]$, let $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$. Then*

$$\sigma \in \mathrm{FP}(G) \quad \Rightarrow \quad \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \quad \text{for all } i \in [N].$$

In other words, every fixed point support of G is a union of (nonempty) component fixed point supports, exactly one per component.

Figure 4 provides a simple visual summary of the different architectures generalizing the cyclic union, together with the main results on their fixed point supports.

We conjecture that the backward direction of the statement in Theorem 1.5 also holds, yielding an if and only if characterization of the fixed point supports.

Conjecture 1.6. *Let G be a directional cycle whose components form a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$. Then*

$$\sigma \in \mathrm{FP}(G) \quad \Leftrightarrow \quad \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \quad \text{for all } i \in [N].$$

In other words, $\mathrm{FP}(G)$ consists of all possible unions of (nonempty) component fixed point supports taking exactly one per component.

If the conjecture is true, then $\mathrm{FP}(G)$ for a simply-embedded directional cycle is identical to that of the cyclic union with the same component subgraphs. While we cannot prove this conjecture in general, we have observed that it holds in computational analyses of over 10,000 simply-embedded directional cycles. For example, Figure 5 shows a larger example of a cyclic union and a simply-embedded directional cycle on the same component subgraphs, and $\mathrm{FP}(G)$ is identical for both networks. Figure 5 also shows an attractor for each network, and we see that the dynamics are qualitatively the same between the cyclic union and the simply-embedded directional cycle. Interestingly, the activity of the simply-embedded directional

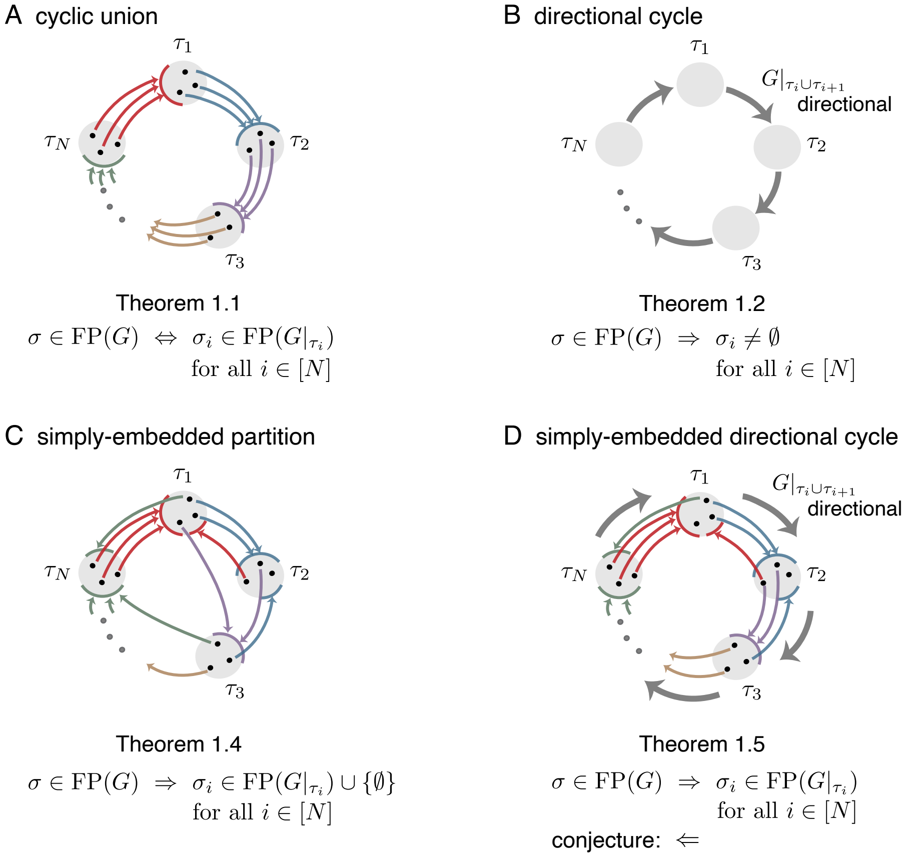

A cyclic union
τ_1
τ_N
τ_2
τ_3
Theorem 1.1
$\sigma \in \mathrm{FP}(G) \Leftrightarrow \sigma_i \in \mathrm{FP}(G|_{\tau_i})$ for all $i \in [N]$
B directional cycle
τ_1
$G|_{\tau_i \cup \tau_{i+1}}$ directional
τ_N
τ_2
τ_3
Theorem 1.2
$\sigma \in \mathrm{FP}(G) \Rightarrow \sigma_i \neq \emptyset$ for all $i \in [N]$
C simply-embedded partition
τ_1
τ_N
τ_2
τ_3
Theorem 1.4
$\sigma \in \mathrm{FP}(G) \Rightarrow \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\}$ for all $i \in [N]$
D simply-embedded directional cycle
τ_1
$G|_{\tau_i \cup \tau_{i+1}}$ directional
τ_N
τ_2
τ_3
Theorem 1.5
$\sigma \in \mathrm{FP}(G) \Rightarrow \sigma_i \in \mathrm{FP}(G|_{\tau_i})$ for all $i \in [N]$
conjecture: $\Leftarrow$

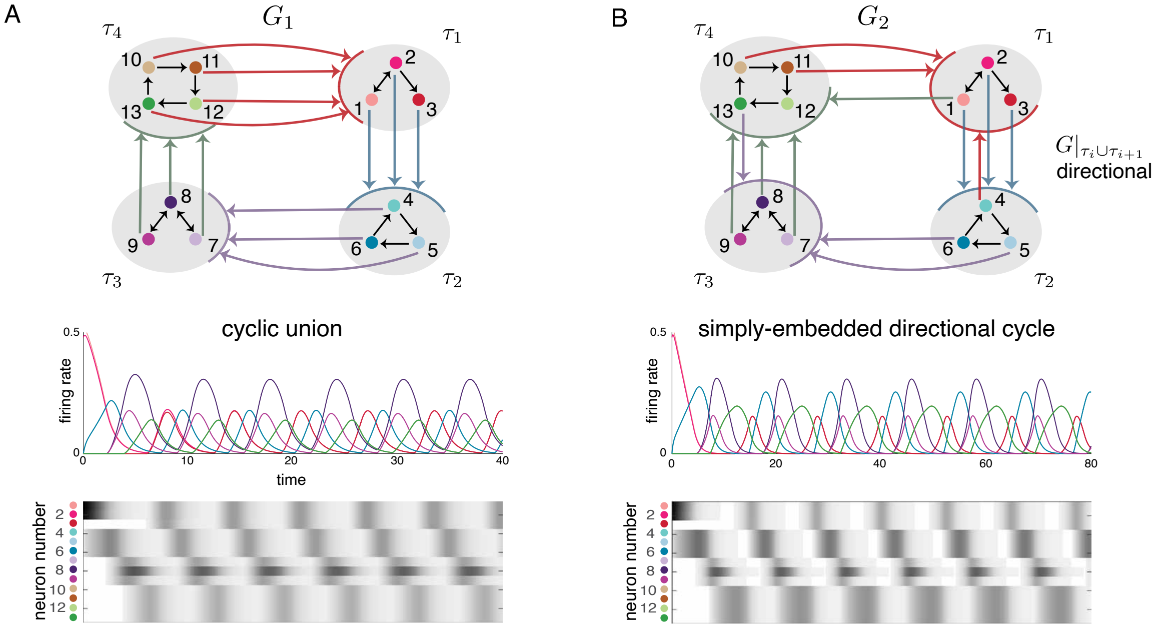
Figure 4. *Summary of main results. In each graph, colored edges from a node to a component indicate that the node projects edges out to all the nodes in the receiving component, as needed for a simply-embedded partition. Thick gray edges indicate directionality of the subgraph $G|_{\tau_i \cup \tau_{i+1}}$.*

cycle is significantly slower (see the differing time axes), with a period approximately twice as long as that of the perfect cyclic union.

In the special case where G is a simply-embedded directional cycle with a unique fixed point per component, Theorem 1.5 shows that the only candidate fixed point support in $\mathrm{FP}(G)$ is the union of these component fixed point supports. Since any network must have at least one fixed point [11, Theorem 1], we immediately obtain the following result.

Proposition 1.7. *Let G be a directional cycle whose components form a simply-embedded partition $\{\tau_1 | \cdots | \tau_N\}$, and suppose $\mathrm{FP}(G|_{\tau_i})$ has a unique fixed point for every τ_i. Then $\mathrm{FP}(G)$ has a unique fixed point with support $\sigma = \cup_{i=1}^{N} \sigma_i$, where σ_i is the unique fixed point support in $\mathrm{FP}(G|_{\tau_i})$.*

In particular, observe that Proposition 1.7 implies that the conjecture holds in the special case where the components each have a unique fixed point. As an illustration of this result, notice that the graphs in Figures 2B–D and Figure 3D are all simply-embedded directional cycles that have a unique fixed point with full support, which is the union of the unique



C For $G = G_1$ or $G = G_2$:

$\text{FP}(G|_{\tau_1}) = \{12,\ 3,\ 123\}$ $\text{FP}(G|_{\tau_2}) = \{456\}$

$\text{FP}(G|_{\tau_3}) = \{78,\ 89,\ 789\}$ $\text{FP}(G|_{\tau_4}) = \{10\,11\,12\,13\}$

$\text{FP}(G) = \{12\,456\,78\,10\,11\,12\,13,\ 12\,456\,89\,10\,11\,12\,13,\ 12\,456\,789\,10\,11\,12\,13,$
$3\,456\,78\,10\,11\,12\,13,\ 3\,456\,89\,10\,11\,12\,13,\ 3\,456\,789\,10\,11\,12\,13,$
$123\,456\,78\,10\,11\,12\,13,\ 123\,456\,89\,10\,11\,12\,13,\ 123\,456\,789\,10\,11\,12\,13\}$

Figure 5. *Cyclic union, simply-embedded directional cycle, their dynamics and* $\mathbf{FP(G)}$. (A) *(Top) A cyclic union G_1 with component subgraphs $G|_{\tau_1}, \ldots, G|_{\tau_4}$. Thick colored edges from a node to a component indicate that the node projects edges out to all the nodes in the receiving component. (Bottom) A solution for the corresponding CTLN.* (B) *(Top) A simply-embedded directional cycle G_2 with the same component subgraphs as G_1. (Bottom) A solution for the corresponding CTLN with the same initial condition as the network in panel* A. *The solution of this network is qualitatively the same as that for the network in* A *except that the period is twice as long (note the different time axes).* (C) *(Left) The fixed point supports* $\text{FP}(G|_{\tau_i})$ *of each component subgraph. (Right)* $\text{FP}(G)$ *is identical for both G_1 and G_2: it consists of unions of component fixed point supports, exactly one per component. To highlight this structure of all the fixed point supports, the portion of each support that each τ_i is color-coded.*

full-support fixed points of each component. For each of these graphs, their $\text{FP}(G)$ coincides with that of the corresponding perfect cyclic union, and their dynamics are qualitatively identical.

Roadmap. The remainder of the paper is organized as follows. Section 2 focuses on constructions involving directional graphs. This includes *directional chains*, which generalize synfire chains, as well as directional cycles. The proof of Theorem 1.2 characterizing $\text{FP}(G)$ for directional cycles is given in section 2.4.

Section 3 is focused on simply-embedded partitions and the constraints they impose on $\text{FP}(G)$. The first section recaps Theorem 1.4 and illustrates it with some examples, then highlights some other interesting consequences about when a node is *removable* from a network. The remaining sections focus on graphs that have a simply-embedded partition together with some additional structure. Section 3.2 examines simply-embedded directional cycles and provides the proof of Theorem 1.5. Section 3.3 explores simple linear chains, which have a purely feedforward chain-like architecture, but without a guarantee of intrinsically feedforward activity (in contrast to directional chains). Section 3.4 characterizes $\mathrm{FP}(G)$ for graphs with a *strongly simply-embedded partition*. The proofs of all the results in section 3 require significant technical machinery and are thus given in the appendix in the supplementary material, sections SM1.2–SM1.6 (except for the straightforward proof of Theorem 1.5).

Finally, in section 4, we analyze a number of networks of size $n = 5$ to show how directional cycle graph architecture is predictive of the structure of corresponding sequential attractors, particularly when a graph has a simply-embedded directional cycle representation. From this analysis, we see that these architectures give insight into the sequences of neural activity that emerge, and not only the structure of fixed point supports.

2. Directional graphs, chains, and cycles.

In this section, we focus on generalizing cyclic unions in a way that preserves the cyclic dynamics of the associated attractor. We refer to these networks as *directional cycles*, which are built from component subgraphs where each consecutive pair forms a *directional graph*. In order to make these notions more precise, we first provide a brief overview of key concepts about fixed points of CTLNs and some graph rules constraining the fixed point supports.

2.1. Preliminaries and prior graph rules.

In this subsection we recall the results from [11] that are relevant for this work. A *fixed point* of a CTLN is simply a fixed point of the network equations (1). In other words, it is a vector $x^* \in \mathbb{R}^n_{\geq 0}$ such that $\frac{dx_i}{dt}|_{x=x^*} = 0$ for all $i \in [n]$. The fixed points of CTLNs can be labeled by their *supports*, and for a given G the set of all fixed point supports is denoted $\mathrm{FP}(G) = \mathrm{FP}(G, \varepsilon, \delta)$.[4]

In [11], multiple characterizations of $\mathrm{FP}(G)$ were developed for *nondegenerate*[5] inhibitory threshold-linear networks in general as well as CTLNs specifically, including a variety of graph rules for CTLNs. As an immediate consequence of one of these characterizations, it was shown that σ is the support of a fixed point, i.e., $\sigma \in \mathrm{FP}(G)$, precisely when

1. $\sigma \in \mathrm{FP}(G|_\sigma)$, and
2. $\sigma \in \mathrm{FP}(G|_{\sigma\cup\{k\}})$ for every $k \notin \sigma$.

(See the appendix section SM1.1 and Corollary SM1.3 for more details). We say that σ is a *permitted motif* of G when it is a fixed point of its restricted subnetwork, so that condition 1 holds. And we say that a permitted motif σ *survives* to support a fixed point in the full network when condition 2 is satisfied. Note that whether a subset σ is permitted depends only on the subgraph $G|_\sigma$ (and potentially the choice of parameters ε and δ), while its survival will depend on the embedding of this subgraph in the full graph. Importantly, condition 2 shows that survival can be checked one external node k at a time. Moreover, it turns out that the only aspect of the embedding that is relevant to survival is the edges from σ *out* to node k; the edges from k back to σ or to any other nodes in G do not affect the survival of σ.

As our first example of permitted motifs, we consider *uniform in-degree* graphs. This family is particularly nice because the survival rules are parameter independent and can be easily checked directly from the graph.

[4] As a slight abuse of notation, we typically omit the dependence of $\mathrm{FP}(G)$ on ε and δ for simplicity. Whenever a fixed point support can be determined using graph rules, its existence is independent of parameters, and thus this simplified notation is appropriate.

[5] See section SM1.1 for the precise definition of nondegeneracy.

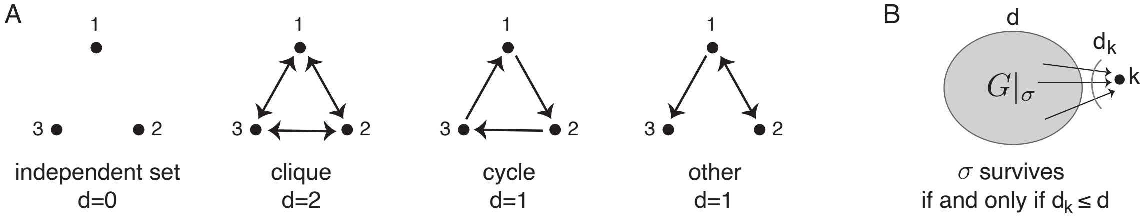


Figure 6. *Uniform in-degree graphs.* (A) *All $n = 3$ graphs with uniform in-degree.* (B) *Cartoon showing survival rule for an arbitrary subgraph with uniform in-degree d.*

Definition 2.1. *Let G be a graph on n nodes and $\sigma \subseteq [n]$. We say that $G|_\sigma$ has uniform in-degree d if every $i \in \sigma$ has in-degree $d_i^{\text{in}} = d$ within $G|_\sigma$, i.e., every node $i \in \sigma$ has d incoming edges from other nodes in σ.*

Rule 1 (uniform in-degree [11]). *Let G be a graph and suppose $G|_\sigma$ has uniform in-degree d. For $k \notin \sigma$, let $d_k \overset{\text{def}}{=} |\{i \in \sigma \mid i \to k\}|$ be the number of edges k receives from σ. Then*

$$\sigma \in \text{FP}(G) \quad \Leftrightarrow \quad d_k \leq d \ \textit{ for all } \ k \notin \sigma.$$

Figure 6A shows all the uniform in-degree graphs of size $n = 3$ together with some general graph theory terminology. Specifically, an *independent set* is a graph with uniform in-degree $d = 0$. A k-*clique* is an all-to-all bidirectionally connected graph with uniform in-degree $d = k - 1$. An n-*cycle* is a graph with n edges, $1 \to 2 \to \cdots \to n \to 1$, which has uniform in-degree $d = 1$. Note that these families of uniform in-degree graphs are all cyclically symmetric; however, this is not necessary for uniform in-degree graphs in general, as the last graph in Figure 6A shows. Rule 1 guarantees that independent sets, cliques, and cycles all have a full-support fixed point. In fact, this fixed point is symmetric, with $x_i^* = x_j^*$ for all $i, j \in [n]$. This is true even for uniform in-degree graphs that are not symmetric. Moreover, Rule 1 guarantees that these fixed points survive within a larger network whenever each external node receives only a limited number of inputs from the subnetwork.

More generally, fixed points can have very different values across neurons and their survival cannot be determined simply by the number of outgoing edges. 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 nodes j, k that have the property that all nodes 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 strictly more inputs than j, and this imbalance rules out their ability to coexist in the same fixed point support. A similar analysis of relative inputs to different nodes can be used to determine fixed point survival in certain cases. These ideas are made more precise below with the notion of *graphical domination*, first introduced in [11].

Definition 2.2. *We say that k* graphically dominates *j with respect to σ, and write $k >_\sigma j$, if $\sigma \cap \{j, k\} \neq \emptyset$ and 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 $j \to k$, and*
3. *if $k \in \sigma$, then $k \not\to j$.*

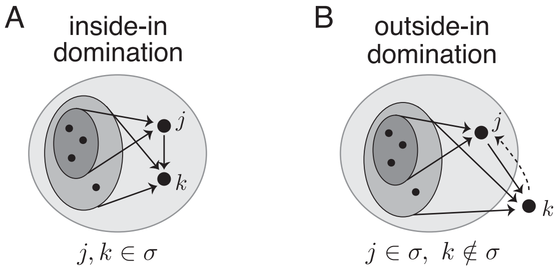


Figure 7. *The three cases of graphical domination in Rule 2. In each panel, k graphically dominates j with respect to σ (the outermost shaded region). The inner shaded regions illustrate the subsets of nodes that send edges to j and k. Note that the vertices sending edges to j are a subset of those sending edges to k, but this containment need not be strict. Dashed arrows indicate optional edges between j and k.*

Figure 7 shows the three cases of domination in which we can conclude whether σ supports a fixed point of the network. Specifically, if there is inside-in domination (panel A), then σ will not be a permitted motif, and thus $\sigma \notin \mathrm{FP}(G)$. If there is outside-in domination by node k (panel B), then σ does not *survive* the addition of node k, and so again $\sigma \notin \mathrm{FP}(G)$. In contrast, if there is inside-out domination (panel C), then σ is guaranteed to survive the addition of node j whenever σ is a permitted motif. These cases were proven in [11, Theorem 4] and are summarized below in Rule 2.

Rule 2 (graphical domination [11]). *Suppose k graphically dominates j with respect to σ. Then the following statements all hold:*

a. *(inside-in) If $j, k \in \sigma$, then $\sigma \notin \mathrm{FP}(G|_\sigma)$, and thus $\sigma \notin \mathrm{FP}(G)$.*
b. *(outside-in) If $j \in \sigma$ and $k \notin \sigma$, then $\sigma \notin \mathrm{FP}(G|_{\sigma \cup \{k\}})$, and thus $\sigma \notin \mathrm{FP}(G)$.*
c. *(inside-out) If $k \in \sigma$ and $j \notin \sigma$, then $\sigma \in \mathrm{FP}(G|_{\sigma \cup \{j\}})$ if and only if $\sigma \in \mathrm{FP}(G|_\sigma)$.*

One case where graphical domination is guaranteed to exist is when σ has a *target*. We say that k is a *target* of σ if $i \to k$ for all $i \in \sigma \setminus \{k\}$. Whenever σ has a target node k, if $k \notin \sigma$, then we are guaranteed that $\sigma \notin \mathrm{FP}(G)$ by outside-in domination. On the other hand, for $k \in \sigma$, if there is any node $j \in \sigma$ such that $k \not\to j$, then we have inside-in domination $k >_\sigma j$ and so again $\sigma \notin \mathrm{FP}(G)$.

At the other extreme, if there is a $j \notin \sigma$ such that j does not receive any edges from σ, then we are guaranteed that every $k \in \sigma$ inside-out dominates node j. Thus, by Rule 2c, σ survives the addition of node j whenever σ is a permitted motif.

Rules 1 and 2 provide some graphical constraints on possible fixed point supports and on when a fixed point of a subnetwork survives to the full network. Rule 3 provides one more constraint on $\mathrm{FP}(G)$. Rule 3 is particularly useful for figuring out if there is a full-support fixed point whenever we know which proper subgraphs are permitted and which yield surviving fixed points. Recall that these graph rules are for nondegenerate CTLNs.

Rule 3 (parity [11]). *For any graph G, the total number of fixed points $|\mathrm{FP}(G)|$ is odd.*

2.2. Directional graphs. With this background in place, we can now precisely define directional graphs.

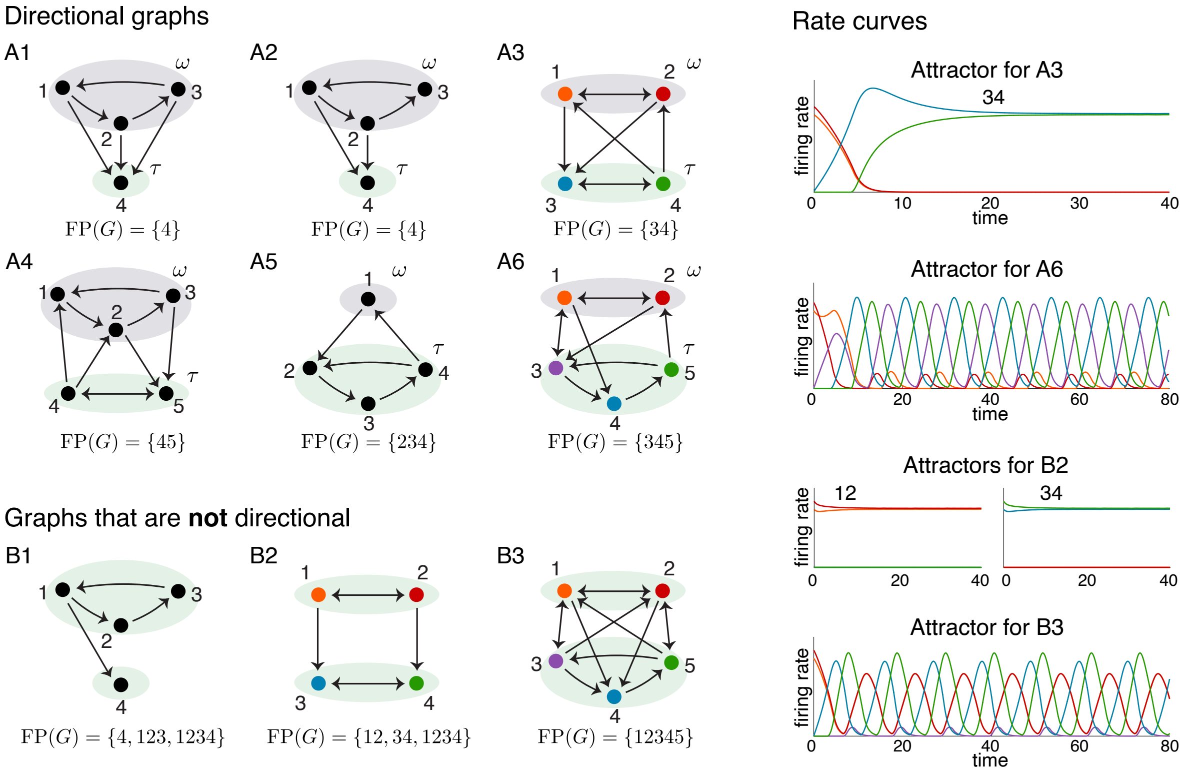


Figure 8. *Directional graphs: examples and non-examples.* (A) *Example directional graphs and their* FP(G)*. On the right, solutions for* A3 *and* A6 *are shown where the activity was initialized on the nodes in* ω*.* (B) *Example nondirectional graphs with their* FP(G)*, as well as solutions for the networks in* B2 *and* B3*.*

Definition 2.3 (directional graphs). *We say that a graph G on n nodes is* directional, *with* direction $\omega \to \tau$*, if $\omega \cup \tau = [n]$ is a nontrivial partition of the nodes ($\omega, \tau \neq \emptyset$) such that the following property holds: for every $\sigma \not\subseteq \tau$, there exists some $j \in \sigma \cap \omega$ and $k \in [n]$ such that k graphically dominates j with respect to σ. In particular, this property guarantees that* $\text{FP}(G) \subseteq \text{FP}(G|_\tau)$ *for all ε, δ in the legal range.*[6]

We call these graphs directional because in simulations we have observed that activity flows from ω to τ, converging to an attractor concentrated on τ. The top panel of Figure 8 shows some example directional graphs. Notice that each graph has a partition of the nodes $\omega \cup \tau$ such that all the fixed point supports are confined to τ; moreover, each subset of nodes that intersects ω does not yield a fixed point as a result of graphical domination.

Example 2.4. Consider the graph G in panel A3. To see that G is directional, observe that any $\sigma \subseteq \{1, 2, 3, 4\}$ containing node 1 cannot support a fixed point because node 3 will dominate 1 with respect to σ since (1) node 3 receives all the inputs that node 1 receives, (2) $1 \to 3$, and (3) $3 \not\to 1$. Similarly, any σ containing node 2 cannot support a fixed point

[6] Note that we could guarantee $\text{FP}(G) \subseteq \text{FP}(G|_\tau)$ in a parameter-independent way without requiring that the dominated node $j \in \sigma$ live in ω. However, for Lemma 2.6 (pairwise chaining) and Theorem 1.2 (directional cycles), we must further require $j \in \sigma \cap \omega$.

since node 3 also dominates node 2. Thus, every $\sigma \in \text{FP}(G)$ must be a subset of τ, and so $\text{FP}(G) \subseteq \text{FP}(G|_\tau)$ as a result of graphical domination.

To the right of A3, we see the dynamics of the network when the activity has been initialized on the nodes in ω. The activity quickly flows from ω and converges to the stable fixed point supported on τ, as predicted by the directionality of G. This flow of activity occurs despite the multiple edges back from nodes in τ to nodes in ω. Similarly, graphs A4–A6 have equal numbers of forward and backward edges between ω and τ, but in each case the dynamics flow toward τ. In particular, the attractor for A6, obtained by initializing activity on ω, is a sequential limit cycle supported on τ.

In contrast, panels B1–B3 exhibit graphs that are *not* directional: in particular, each one has a full-support fixed point. The graph in B2 is somewhat surprising, because it is similar to A3 but with a more obviously feedforward architecture. Dynamically, however, this graph is not directional and in fact supports two stable fixed point attractors that together involve all four nodes (see the attractors shown to the right). Thus, feedforward architecture alone is not sufficient to guarantee feedforward dynamics. Moreover, directional graphs can have feedforward dynamics even in the absence of feedforward architecture, as the graph in A3 demonstrates (see also Figures 2 and 3 of [15]).

We can also see directional graphs inside cyclic unions. For example, in Figure 3A, the induced subgraphs $G|_{\tau_1 \cup \tau_2}$, $G|_{\tau_2 \cup \tau_3}$, and $G|_{\tau_3 \cup \tau_1}$ are all directional. In fact, in any cyclic union, the induced subgraph on a pair of adjacent components is always directional. Such subgraphs are a special case of the family of graphs in Figure 9, where the τ component contains a target of ω and there are no back edges from τ to ω.

Lemma 2.5. *Suppose G has a nontrivial partition of its nodes $\omega \cup \tau = [n]$ such that there are only forward edges from ω to τ and at least one node in τ is a target of ω (as in Figure 9). Then G is directional with direction $\omega \to \tau$.*

Proof. Let $k \in \tau$ be a target of ω. Consider any $\sigma \not\subseteq \tau$ and let $j \in \sigma \cap \omega$. We will show that the target node k graphically dominates j with respect to σ. First we must show that for all $i \in \sigma \setminus \{j,k\}$, if $i \to j$, then $i \to k$. Since there are no back edges from τ to ω, the only $i \in \sigma$ with $i \to j$ are $i \in \sigma \cap \omega$. But k is a target of ω, and so $i \in \omega$ implies that $i \to k$. Thus condition 1 of graphical domination holds. Moreover, $j \to k$ since k is a target, and $k \not\to j$ since there are no back edges from τ. Thus, conditions 2 and 3 hold as well, and so k dominates j with respect to σ. Therefore G is directional with direction $\omega \to \tau$. ■

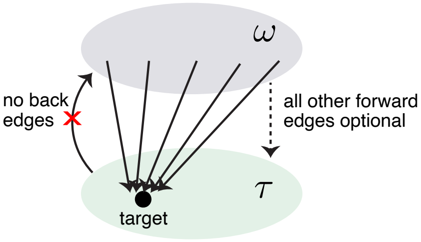


Figure 9. *Family of directional graphs.*

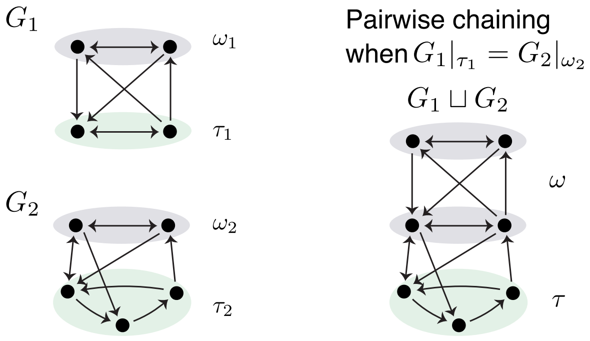


Figure 10. *Pairwise chaining of two directional graphs. (Left) Two directional graphs G_1 and G_2 with direction $\omega_i \to \tau_i$, where $G_1|_{\tau_1} = G_2|_{\omega_2}$. (Right) The graph $G_1 \sqcup G_2$ formed from chaining together G_1 and G_2 by identifying τ_1 with ω_2. By Lemma 2.6, $G_1 \sqcup G_2$ is directional for $\omega = \omega_1 \cup \omega_2$ (in gray) and $\tau = \tau_2$ (in green).*

2.3. Directional chains.

One of the valuable features of directional graphs is that we can chain them together to produce new directional graphs. Namely, if the graphs overlap so that the τ part of one graph coincides with the ω part of the next one, then the resulting "chain" graph is also directional (see Figure 10).

Lemma 2.6 (pairwise chain). *Suppose G_1 and G_2 are directional graphs with directions $\omega_1 \to \tau_1$ and $\omega_2 \to \tau_2$, respectively, that satisfy $G_1|_{\tau_1} = G_2|_{\omega_2}$. Consider the pairwise chain $G_1 \sqcup G_2$ formed by identifying τ_1 with ω_2 (as in Figure 10). Then $G_1 \sqcup G_2$ is directional with $\omega = \omega_1 \cup \omega_2$ and $\tau = \tau_2$.*

Proof. Let $G \overset{\text{def}}{=} G_1 \sqcup G_2$ be the pairwise chain with vertex set $[n] = \omega_1 \cup \omega_2 \cup \tau_2$ (where τ_1 was identified with ω_2), and let $\omega \overset{\text{def}}{=} \omega_1 \cup \omega_2$, and $\tau \overset{\text{def}}{=} \tau_2$. Consider $\sigma \subseteq [n]$ with $\sigma \cap \omega \neq \emptyset$. We will show that there exists a $j \in \sigma \cap \omega$ and $k \in [n]$ such that k graphically dominates j with respect to σ. Observe that if $\sigma \subseteq \omega_1 \cup \omega_2 = \omega_1 \cup \tau_1$ and $\sigma \cap \omega_1 \neq \emptyset$, then such a j and k pair exist within G_1 since G_1 is directional; the same holds if $\sigma \subseteq \omega_2 \cup \tau_2$. Thus we need only consider σ that overlaps both G_1 and G_2, without being fully contained in either. In other words, we have $\sigma \cap \omega_1 \neq \emptyset$ and $\sigma \cap \tau_2 \neq \emptyset$.

Let $\sigma_1 \overset{\text{def}}{=} \sigma \cap (\omega_1 \cup \tau_1)$ be σ restricted to G_1. By the directionality of G_1, there exists $j \in \sigma_1 \cap \omega_1$ and $k \in \omega_1 \cup \tau_1$ such that k graphically dominates j with respect to σ_1. We will show that k also dominates j with respect to the full σ in G. First observe that conditions 2 and 3 of graphical domination are automatically satisfied for σ by way of being satisfied for σ_1. For condition 1, we must show that for all $i \in \sigma \setminus \{j, k\}$, if $i \to j$, then $i \to k$. Since $j \in \omega_1$, the only possible nodes in G that can send edges to j are nodes in G_1, since ω_1 is not in the overlap with G_2 so cannot receive edges from any nodes in G_2 outside of that overlap. Thus, the only $i \in \sigma$ with $i \to j$ are nodes within σ_1, and for all $i \in \sigma_1 \setminus \{j, k\}$, whenever $i \to j$, we have $i \to k$ by the graphical domination relationship with respect to σ_1. Therefore, condition 1 holds for all of σ, and so k dominates j with respect to σ in G. Thus, G is directional with $\omega \to \tau$. ■

Lemma 2.6 motivates the following definition of a *directional chain*, obtained from iteratively chaining directional graphs together (see Figure 11).

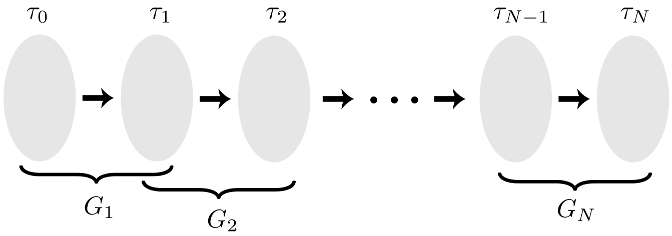


Figure 11. *Directional chain. A cartoon of a directional chain with components $\tau_0, \tau_1, \ldots, \tau_N$. For each $i = 1, \ldots, N$, the induced subgraph $G_i \overset{\text{def}}{=} G|_{\tau_{i-1} \cup \tau_i}$ is directional. The arrows between components indicate the directionality $\tau_{i-1} \to \tau_i$. Note there may be edges in both directions between adjacent components, as in the example directional graphs in Figure 8, but there are no edges between nonadjacent components.*

Definition 2.7 (directional chain). *Let G be a graph with a partition of its nodes $\{\tau_0|\tau_1|\cdots|\tau_N\}$. For each $i = 1, \ldots, N$, let $G_i \overset{\text{def}}{=} G|_{\tau_{i-1}\cup\tau_i}$ be the induced subgraph on adjacent components. We say that G is a* directional chain *if each G_i is directional with direction $\tau_{i-1} \to \tau_i$, and every edge of G is an edge in some G_i (i.e., there are no edges between nonadjacent τ_i components).*

Iteratively applying Lemma 2.6, we immediately obtain the following result showing that every directional chain is directional.

Proposition 2.8 (directional chain). *Let G be a directional chain with components $\tau_0, \tau_1, \ldots, \tau_N$ and directional graphs $G_i = G|_{\tau_{i-1}\cup\tau_i}$. Then G is directional with direction $\omega \to \tau$ for $\omega = \tau_0 \cup \cdots \cup \tau_{N-1}$ and $\tau = \tau_N$. In particular,* $\text{FP}(G) \subseteq \text{FP}(G|_{\tau_N})$.

Proof. For each $G_i \overset{\text{def}}{=} G|_{\tau_{i-1}\cup\tau_i}$, denote the directional components of G_i as ω_i and τ_i, as in Lemma 2.6, so that $\omega_i = \tau_{i-1}$. Observe $G_{12} \overset{\text{def}}{=} G_1 \sqcup G_2$ is a pairwise chain, so by Lemma 2.6, G_{12} is directional with $\omega_{12} = \omega_1 \cup \omega_2$ and $\tau_{12} = \tau_2$. Similarly, $G_{123} \overset{\text{def}}{=} (G_1 \sqcup G_2) \sqcup G_3 = G_{12} \sqcup G_3$ is also a pairwise chain, and so by Lemma 2.6, G_{123} is directional with $\omega_{123} = \omega_{12} \cup \omega_3 = \omega_1 \cup \omega_2 \cup \omega_3$ and $\tau_{123} = \tau_3$. We can continue iterating in this fashion to see G is a pairwise chain of directional graphs $G_{1\ldots N-1} \sqcup G_N$, and thus by Lemma 2.6, G is directional with direction $\omega \to \tau$ for $\omega = \omega_{1\ldots N-1} \cup \omega_N = \tau_0 \cup \cdots \cup \tau_{N-1}$ and $\tau = \tau_N$. ■

By Proposition 2.8, we see that for any CTLN whose graph is a directional chain G, we must have $\text{FP}(G) \subseteq \text{FP}(G|_{\tau_N})$. In other words, all fixed points are confined to the last τ of the chain. Figure 12A gives an example of such a chain built from directional graphs $G_1, \ldots, G_4$ where $G_i|_{\tau_i} = G_{i+1}|_{\tau_i}$ for each $i = 1, \ldots, 3$. In Figure 12B, we see the resulting dynamics when the activity of the network is initialized on nodes 1 and 2, at the start of the chain. We see a clear sequence of activation, from 1 and 2 to $3, 5, 6$, and 7, and then stabilizing on the fixed point attractor for the clique $\{9, 10\}$. In other words, the activity flows along the directional chain, generating a sequence that reflects the directionality of the construction. Note that the network behaves in the expected feedforward manner dynamically despite the existence of several feedback edges: $4 \to 1, 2$, $8 \to 5$, and $9 \to 7$.

We see that directional chains produce sequences of neural activity in their transient dynamics, similar to that of synfire chains [1, 3, 21]. In contrast to synfire chains, though,

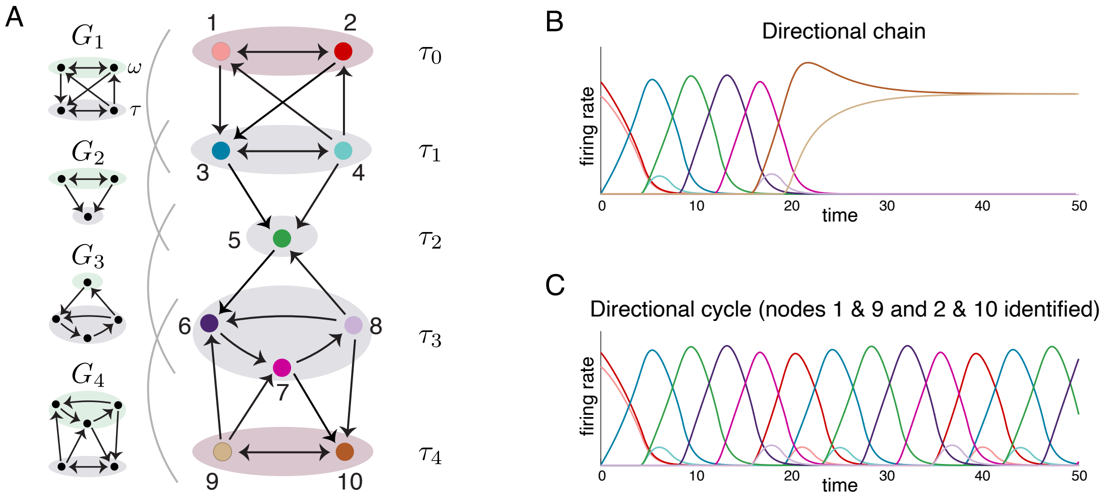


Figure 12. *Directional chain and directional cycle.* (A) *A graph built from chaining together directional graphs $G_1, \ldots, G_4$, where $G_i = G|_{\tau_{i-1} \cup \tau_i}$.* (B) *A solution of the CTLN for the directional chain in* A *when the activity is initialized on nodes* 1 *and* 2*. Activity flows through the chain, eventually stabilizing on the fixed point attractor of τ_4.* (C) *A solution of the CTLN for the directional cycle obtained from the graph in* A *by identifying τ_4 with τ_0 to make the chain wrap cyclically wrap around. The activity is initialized on nodes* 1 *and* 2 *and falls into a limit cycle hitting all the components τ_i in cyclic order.*

directional chains can have recurrent connectivity throughout and do not rely on a purely feedforward architecture.

2.4. Directional cycles.

We can also chain directional graphs together in a cyclic manner, so the directional chain wraps around and τ_N is identified with τ_0. We call any graph G that can be created in this way a *directional cycle.*

Definition 2.9 (directional cycle). *Let G be a graph with node partition $\{\tau_1 | \cdots | \tau_N\}$. For each $i = 1, \ldots, N$, let $G_i \overset{\text{def}}{=} G|_{\tau_{i-1} \cup \tau_i}$ be the induced subgraph on adjacent components (cyclically identifying $\tau_N = \tau_0$). We say that G is a* directional cycle *if each G_i is directional with direction $\tau_{i-1} \to \tau_i$, and every edge of G is an edge in some G_i (i.e., there are no edges between nonadjacent τ_i components).*

For directional cycles, the chain has no beginning or end and so the fixed points cannot all lie in some final τ_N. Instead, they become highly distributed across the network, intersecting each and every τ_i. In particular, in Theorem 1.2, we show that every fixed point support of a directional cycle contains an *undirected cycle*[7] that hits each τ_i in cyclic order.

Figure 12 provides an illustration of this. In the directional chain of panel A, suppose we identify τ_4 with τ_0, so nodes 1 and 9 are identified as are 2 and 10. Then the resulting network becomes a directional cycle. Figure 12C shows the activity obtained by initializing on nodes

[7] To any directed graph G, we can associate a simple undirected graph $\widehat{G}$ by ignoring the direction on the edges. An *undirected cycle* is a sequence of nodes connected by edges that form a cycle within the underlying undirected graph. For example, 2458 is an undirected cycle in the graph in Figure 12A when node 10 is identified with node 2.

1 and 2. We see a clear and repeating sequence of activity emerge, corresponding to the cycle 23567 in the graph, whose existence is predicted by Theorem 1.2. Note that for this network $\mathrm{FP}(G) = \{23567\}$ with the unique fixed point corresponding to the cycle motif giving rise to the sequence. In other words, directional cycles produce periodic sequences of activity that cycle around the chain in the expected direction.

The remainder of this section is dedicated to the proof of Theorem 1.2 (reprinted below).

Theorem 1.2 (cyclic fixed points of directional cycles). *Let G be a directional cycle with components $\tau_1, \ldots, \tau_N$ and directional graphs $G_i = G|_{\tau_{i-1} \cup \tau_i}$ (cyclically identifying $\tau_N = \tau_0$). Then for any $\sigma \in \mathrm{FP}(G)$, the graph $G|_\sigma$ contains an undirected cycle that intersects every τ_i in cyclic order (see the illustration in Figure 13A).*

To prove Theorem 1.2, we first need the following lemma that shows that for any fixed point support σ of a directional cycle, there is always an edge feeding into σ_i (σ restricted to the graph G_i) from the previous graph G_{i-1}.

Lemma 2.10. *Let G be a directional cycle with components $\tau_1, \ldots, \tau_N$ and directional graphs $G_i = G|_{\tau_{i-1} \cup \tau_i}$ (cyclically identifying $\tau_N = \tau_0$). For each G_i, let $\omega_i = \tau_{i-1}$, so that G_i has direction $\omega_i \to \tau_i$. For $\sigma \in \mathrm{FP}(G)$, let $\sigma_i \overset{\mathrm{def}}{=} \sigma \cap (\omega_i \cup \tau_i)$ denote σ restricted to graph G_i. For any $v \in \sigma_i \cap \omega_i$, there exists $j \in \sigma_i \cap \omega_i$ (j could equal v) and an $\ell \in \sigma_{i-1} \cap \omega_{i-1}$ such that $\ell \to j$ in G and there is an undirected path from v to j in σ_i (see the illustration in Figure 13B).*

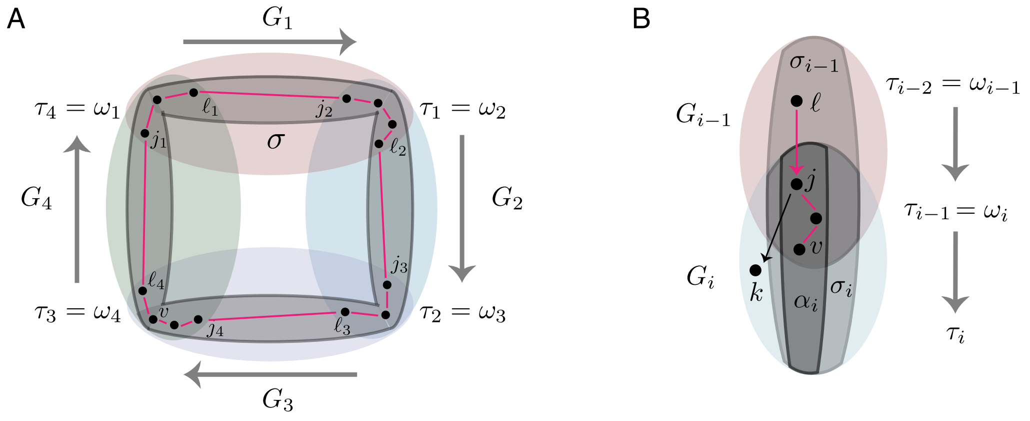


Figure 13. *Illustrations for Theorem 1.2 and Lemma 2.10.* (A) *A cartoon of a directional cycle; each pastel colored blob is a directional graph G_i with direction $\omega_i = \tau_{i-1} \to \tau_i$ indicated by arrows along the outside. Note that all vertices of G lie within an overlap of adjacent G_i, but each G_i has edges between the two overlaps τ_{i-1} and τ_i. Within the directional cycle, a fixed point support $\sigma \in \mathrm{FP}(G)$ is shown in dark gray. Theorem 1.2 guarantees that $G|_\sigma$ contains an undirected cycle that hits all the τ_i in cyclic order (shown in magenta). The vertices in the cycle are labeled following the notation for the proof of Theorem 1.2.* (B) *Cartoon for setup of Lemma 2.10. The pale pink and blue blobs depict overlapping directional graphs G_{i-1} and G_i. The restriction of fixed point support $\sigma \in \mathrm{FP}(G)$ to this subgraph is shown with its component subgraphs $\sigma_i \overset{\mathrm{def}}{=} \sigma \cap (\omega_i \cup \tau_i)$ denoted with light gray blobs. The subgraph $G|_{\sigma_i}$ can be broken into its connected components, and α_i (dark gray) denotes one such component. There exists a $j \in \alpha_i \cap \omega_i$ such that j is dominated by some k in G_i with respect to α_i. Then Lemma 2.10 guarantees that there is some $\ell \in \sigma_{i-1} \cap \omega_{i-1}$ such that $\ell \to j$.*

Proof. Let $\sigma \in \text{FP}(G)$, $\sigma_i = \sigma \cap (\omega_i \cup \tau_i)$, and let α_i be the connected component[8] of σ_i that contains v, so that $\alpha_i \cap \omega_i \neq \emptyset$. Since G_i is directional, there exists $j \in \alpha_i \cap \omega_i$ and $k \in \omega_i \cup \tau_i$ such that k graphically dominates j with respect to α_i. Since $\sigma \in \text{FP}(G)$, we cannot have any j graphically dominated by k with respect to all of σ in G, by Rule 2. Thus, there must exist some $\ell \in \sigma$ such that $\ell \to j$ but $\ell \not\to k$ (in order to violate condition 1 of the definition of graphical domination). Moreover, we must have $\ell \in \tau_{i-2}$, τ_{i-1}, or τ_i, since $j \in \omega_i = \tau_{i-1}$ and $\ell \to j$, and there are no edges between nonadjacent components in a directional cycle. We cannot have ℓ in G_i, i.e., $\ell \notin \tau_{i-1} \cup \tau_i$, since there are no nodes in $\sigma_i \setminus \alpha_i$ that send edges into α_i, by definition of connected component. Thus, we must have $\ell \in \tau_{i-2} = \omega_{i-1}$. Therefore, we have $\ell \in \sigma_{i-1} \cap \omega_{i-1}$ and $j \in \alpha_i \cap \omega_i$ such that $\ell \to j$. ■

We can now prove Theorem 1.2, using Lemma 2.10 to trace a path in σ backward through the directional cycle, demonstrating the existence of an undirected cycle in σ that hits every τ_i in cyclic order.

Proof of Theorem 1.2. To set notation, for each $G_i = G|_{\tau_{i-1} \cup \tau_i}$, denote the directional components of G_i as ω_i and τ_i, so that $\omega_i = \tau_{i-1}$. For $\sigma \subseteq [n]$, let $\sigma_i \overset{\text{def}}{=} \sigma \cap (\omega_i \cup \tau_i)$ denote the restriction of σ to the graph G_i.

Let $\sigma \in \text{FP}(G)$ and let $v \in \sigma$. Observe that $v \in \omega_i$ for some graph G_i (since every node in G is contained in some $\omega_i = \tau_{i-1}$). Without loss of generality, let $v \in \omega_N$. By Lemma 2.10, there exists a $j_N \in \sigma_N \cap \omega_N$ and $\ell_{N-1} \in \sigma_{N-1} \cap \omega_{N-1}$ such that $\ell_{N-1} \to j_N$ and there is an undirected path from v to j_N. Next, consider ℓ_{N-1} playing the role of v in $\sigma_{N-1} \cap \omega_{N-1}$. We can again apply Lemma 2.10 to obtain a $j_{N-1} \in \sigma_{N-1} \cap \omega_{N-1}$ and $\ell_{N-2} \in \sigma_{N-2} \cap \omega_{N-2}$ such that $\ell_{N-2} \to j_{N-1}$ and there is an undirected path from ℓ_{N-1} to j_{N-1}. Thus, we have an undirected path from v to j_N to ℓ_{N-1} to j_{N-1} and finally ℓ_{N-2} (see Figure 13A starting in the bottom left ω_4).

Continuing in this manner, we see that $G|_\sigma$ has an undirected path containing all these $j_i \in \omega_i = \tau_{i+1}$, and hitting each of the intersections τ_i in cyclic order. To see that this path can eventually be closed to yield a cycle, notice that we can keep following this path backward from G_i to G_{i-1} as it wraps around G, since every σ_i on this path must have some edge into it from σ_{i-1} that can be followed backward. Since each σ_i has a finite number of connected components, by the pigeonhole principle, the path through σ must at some point revisit a connected component α_i for some i. Since α_i is connected, we can close our cycle by walking from the current node on the path through the component to the node previously visited in an earlier portion of the path. Thus we have found an undirected path through σ that starts and ends at the same point in some σ_i, yielding an undirected cycle that hits every τ_j in cyclic order. ■

3. Simply-embedded structure and graph rules.

In this section, we focus on graphs with a *simply-embedded partition*. These graphs generalize cyclic unions in a way that preserves strong constraints on $\text{FP}(G)$. We begin by considering simply-embedded partitions in their full generality, and then move to some families of graphs that have additional structure. Note

[8] In a slight abuse of language, we use connected component here to refer to the connected component of the undirected graph associated to G. Thus, a connected component consists of all nodes that are reachable by undirected paths, where the direction of edges in G is ignored.

that most of the results in this section require significant technical machinery to prove (initially developed in [11]), and thus we save the proofs for the appendix in sections SM1.2–SM1.6.

3.1. Simply-embedded partitions.

Recall that a simply-embedded partition of a graph is a partition of the nodes such that all nodes within a single component are treated identically by any node outside of that component. More precisely, we give Definition 1.3 again.

Definition 1.3 (simply-embedded partition). *Given a graph G, a partition of its nodes $\{\tau_1|\cdots|\tau_N\}$ is called a* simply-embedded partition *if every τ_i is simply-embedded in G. In other words, for each τ_i and each $k \notin \tau_i$, either $k \to j$ for all $j \in \tau_i$ or $k \not\to j$ for all $j \in \tau_i$.*

Notice that the definition is trivially satisfied in the cases where (a) there are no $k \notin \tau_i$ or (b) there is only a single $j \in \tau_i$ for every i. Thus, every graph has two trivial simply-embedded partitions: one where all the nodes are in one component and one where every node is in its own component. Neither of these partitions is useful for giving information about the structure of G. But when a graph has a nontrivial simply-embedded partition, this structure is sufficient to dramatically constrain the possible fixed point supports of G to unions of fixed points chosen from a *menu* of component fixed point supports, $\text{FP}(G|_{\tau_i})$.

Theorem 1.4 ($\text{FP}(G)$ menu for simply-embedded partitions). *Let G have a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$. For any $\sigma \subseteq [n]$, let $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$. Then*

$$\sigma \in \text{FP}(G) \quad \Rightarrow \quad \sigma_i \in \text{FP}(G|_{\tau_i}) \cup \{\emptyset\} \quad \textit{for all } i \in [N].$$

In other words, every fixed point support of G is a union of component fixed point supports σ_i, at most one per component.

Theorem 1.4 gives significant restrictions on the possible supports in $\text{FP}(G)$ in terms of the component fixed point supports. However, the converse is not true—not every union of supports from the menu is guaranteed to yield a fixed point support in $\text{FP}(G)$. The following examples illustrate the range of $\text{FP}(G)$ that can emerge from the same menu.

Example 3.1. Consider the component subgraphs shown in Figure 14A together with their $\text{FP}(G|_{\tau_i})$. By Theorem 1.4, any graph G with a simply-embedded partition of these component subgraphs has a restricted menu for $\text{FP}(G)$ consisting of the component fixed point supports (the set of all possible supports derived from this menu is shown on the bottom of panel A). Note that an arbitrary graph on 7 nodes could have up to $2^7 - 1 = 127$ possible fixed point supports, but the simply-embedded partition structure narrows the options to only 15 candidate fixed points. Figures 14B–E show four possible graphs with simply-embedded partitions of these component subgraphs, together with $\text{FP}(G)$ for each of the graphs.

Observe that the graph in Figure 14B is a *disjoint union* of its component subgraphs. For this graph, $\text{FP}(G)$ consists of all possible unions of at most one fixed point support per component subgraph (see [11, Theorem 11]). Thus, every choice from the menu provided by Theorem 1.4 does in fact yield a fixed point for G.

In contrast, the graph in Figure 14C is a *cyclic union* of the component subgraphs. For this graph, $\text{FP}(G)$ only has sets that contain a fixed point support from every component, i.e., $\sigma_i \neq \emptyset$ for all $i \in [N]$ (by Theorem 1.1). Thus, any subset from the menu of Theorem 1.4 that does not intersect every τ_i does not produce a fixed point for G.

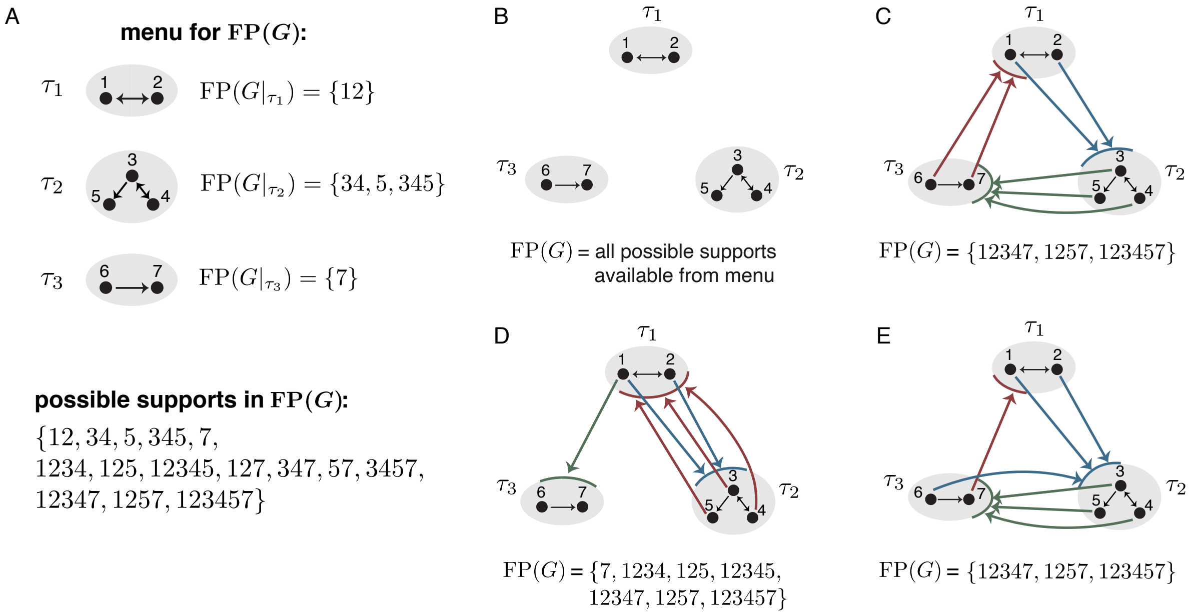


Figure 14. *Graphs with a simply-embedded partition from Example* 3.1. (A) *(Top) A collection of component subgraphs with their* $\mathrm{FP}(G|_{\tau_i})$. *(Bottom) The set of possible fixed point supports for any graph that has these subgraphs in the simply-embedded partition.* (B)–(E) *Example graphs with a simply-embedded partition with the component subgraphs from* A*, together with their* $\mathrm{FP}(G)$. *In* (C)–(E), *thick colored edges from a node to a component indicate that the node projects edges out to all the nodes in the receiving component.*

Meanwhile, the graph in Figure 14D is a simply-embedded partition with heterogeneity in the outgoing edges from a component (notice different nodes in τ_1 treat τ_3 differently). $\mathrm{FP}(G)$ has a mixture of types of supports: there are some $\sigma \in \mathrm{FP}(G)$ that do not intersect every component, and others that do.

Finally, the graph in Figure 14E is another simply-embedded partition with heterogeneity (notice different nodes in τ_3 treat τ_1 and τ_2 differently). However, for this graph, there is a uniform rule for the fixed point supports: every fixed point consists of exactly one fixed point support per component subgraph (identical to $\mathrm{FP}(G)$ for the graph in panel C).

Interestingly, some graphs have multiple nontrivial simply-embedded partitions, which can be analyzed in parallel to give further constraints on $\mathrm{FP}(G)$, and in some cases even fully nail down $\mathrm{FP}(G)$.

Example 3.2. Consider the graph in Figure 15 with the two simply-embedded partitions $\{1\,|2,3,4|\,5\}$ and $\{2\,|1,3,5|\,4\}$. Since 234 is the unique fixed point of τ_2 in the partition in panel A, Theorem 1.4 guarantees that any $\sigma \in \mathrm{FP}(G)$ that contains one of the nodes 2, 3, or 4 must contain all three nodes because $\sigma \cap \tau_2 \in \mathrm{FP}(G|_{\tau_2}) \cup \{\emptyset\}$. Similarly, since 135 is the unique fixed point of τ_2' in the partition in panel B, any $\sigma \in \mathrm{FP}(G)$ that contains one of the nodes 1, 3, or 5 must contain all three. Combining these two constraints, we see that we must have $\mathrm{FP}(G) = \{12345\}$.

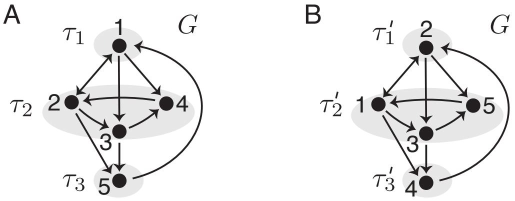


Figure 15. *Graph with two nontrivial simply-embedded partitions. The panels show two drawings of the same graph G to highlight two partitions:* (A) *the simply-embedded partition* $\{1\,|2,3,4|\,5\}$, (B) *the simply-embedded partition* $\{2\,|1,3,5|\,4\}$.

In general though, Theorem 1.4 is not sufficient to fully determine $\mathrm{FP}(G)$, but it does significantly limit the menu of fixed point supports. In particular, one direct consequence of Theorem 1.4 is that if there is some node $j \in \tau_i$ in G that does not participate in any fixed points of $G|_{\tau_i}$, then j cannot participate in any fixed point of the full graph G. Thus the supports of all the fixed points of G are confined to $[n] \setminus \{j\}$. For example, node 6 in Figure 14 does not appear in $\mathrm{FP}(G|_{\tau_3})$ and thus does not appear in any fixed point support for any of the graphs in panels B–E. It turns out that if the removal of node j does not change the fixed points of the component subgraph, i.e., if $\mathrm{FP}(G|_{\tau_i}) = \mathrm{FP}(G|_{\tau_i \setminus \{j\}})$, then we can actually remove j from the full graph G without changing $\mathrm{FP}(G)$. Thus we have the following theorem.

Theorem 3.3 (removable nodes). *Let G have a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$. Suppose there exists a node $j \in \tau_i$ such that $\mathrm{FP}(G|_{\tau_i}) = \mathrm{FP}(G|_{\tau_i\setminus\{j\}})$. Then $\mathrm{FP}(G) = \mathrm{FP}(G|_{[n]\setminus\{j\}})$.*

Theorem 3.3 shows that if a node j is locally removable without altering fixed points of its component, then node j is also globally removable without altering the fixed points of the full graph G. This result gives a new tool for determining that two graphs have the same collection of fixed points.

Corollary 3.4. *Let G have a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$ and suppose there exists $j \in \tau_i$ such that $\mathrm{FP}(G|_{\tau_i}) = \mathrm{FP}(G|_{\tau_i\setminus\{j\}})$. Let G' be any graph that can be obtained from G by deleting or adding outgoing edges from j to any other component without altering the simply-embedded structure of G. Then $\mathrm{FP}(G') = \mathrm{FP}(G)$.*

As an illustration of Corollary 3.4, let G be the graph from Figure 14C and G' be the graph from Figure 14E. It is easy to check that $\mathrm{FP}(G|_{\tau_3}) = \mathrm{FP}(G|_{\tau_3\setminus\{6\}})$, and so node 6 is removable. Since G and G' differ only in edges out from node 6 to other components, and the simply-embedded partition is maintained, Corollary 3.4 guarantees that $\mathrm{FP}(G) = \mathrm{FP}(G')$.

While Theorems 1.4 and 3.3 give significant constraints on $\mathrm{FP}(G)$, the simply-embedded partition structure alone is not sufficient to nail down $\mathrm{FP}(G)$. In the following subsections, we consider a variety of families of graphs that have additional structure that enables us to draw stronger conclusions about $\mathrm{FP}(G)$.

3.2. Directional cycles with a simply-embedded partition. We begin by considering simply-embedded directional cycles, i.e., directional cycles where the directional partition of the nodes into components $\{\tau_1|\cdots|\tau_N\}$ is also a simply-embedded partition. We expect these graphs to be similar to the corresponding cyclic union of the same component subgraphs, both in terms of their dynamics (as a result of the directionality property) and in terms of their $\mathrm{FP}(G)$ (as a result of the simply-embedded partition). Theorem 1.5 guarantees that every fixed point of a simply-embedded directional cycle has the same component structure as those of the corresponding cyclic union.

Theorem 1.5 (simply-embedded directional cycles). . *Let G be a directional cycle whose components form a simply-embedded partition $\{\tau_1|\cdots|\tau_N\}$. For any $\sigma \subseteq [n]$, let $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$. Then*

$$\sigma \in \mathrm{FP}(G) \quad \Rightarrow \quad \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \quad \textit{for all } i \in [N].$$

In other words, every fixed point support of G is a union of (nonempty) component fixed point supports, exactly one per component.

Proof. By Theorem 1.4, for any $\sigma \in \mathrm{FP}(G)$, we have $\sigma_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\}$. By Theorem 1.2, every $\sigma \in \mathrm{FP}(G)$ contains a cycle that intersects every τ_i, and so $\sigma_i \neq \emptyset$ for all $i \in [N]$. Thus, $\sigma_i \in \mathrm{FP}(G|_{\tau_i})$ for all $i \in [N]$. ■

We conjecture that the backward direction of the statement in Theorem 1.5 also holds, yielding an if and only if characterization of the fixed point supports. If this were true, then the fixed point supports of a simply-embedded directional cycle would be identical to those of the corresponding cyclic union. To prove this characterization, it is natural to try to mimic the proof of Theorem 1.1 (Theorem 13 in [11]), which is the analogous result for cyclic unions. The key to that proof is to induct on the size of the cyclic union by analyzing relevant subgraphs $G|_\sigma$. Since each $G|_\sigma$ is itself a cyclic union, the inductive hypothesis can then be applied. Unfortunately, when we consider the analogous subgraphs of a simply-embedded directional cycle, they need not be directional cycles themselves. Thus, induction cannot be used to prove the conjecture. However, computational analyses of over 10,000 simply-embedded directional cycles give us reasonable confidence that the conjecture holds. Specifically, we randomly sampled 1000s of simply-embedded directional cycles with 3, 4, or 5 components where the component subgraphs had up to 4 nodes each. For every one of these simply-embedded directional cycles, we found that $\mathrm{FP}(G) = \{\bigcup_{i=1}^N \sigma_i \mid \sigma_i \in \mathrm{FP}(G|_{\tau_i}) \text{ for all } i \in [N]\}$, as predicted. Moreover, as we saw in Figures 2 and 5, the dynamics of simply-embedded directional cycles tend to mimic those of the corresponding cyclic union.

3.3. Simple linear chains.

In the previous subsection, we saw that when we have a simply-embedded partition on top of a directional cycle structure, this adds significant constraints on $\mathrm{FP}(G)$. It is natural to ask what happens when we cut such a cyclic structure between components and are left with just a directional chain. Does the added structure of a simply-embedded partition similarly give a stronger handle on $\mathrm{FP}(G)$ for a directional chain?

Recall from Proposition 2.8 that a directional chain is provably directional onto the last component, and so $\mathrm{FP}(G) \subseteq \mathrm{FP}(G|_{\tau_N})$. Simply-embedded partitions only add the constraint that for each $\sigma \in \mathrm{FP}(G)$, we have $\sigma_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\}$. But this gives no new information since for directional chains, we are already guaranteed that $\sigma_i = \emptyset$ for all $i \neq N$ and $\sigma_N \in \mathrm{FP}(G|_{\tau_N})$. Figure 16A–C shows examples of directional chains both with and without simply-embedded structure. Notice that in all three of these graphs, $\mathrm{FP}(G)$ is identical, and it is fully predicted

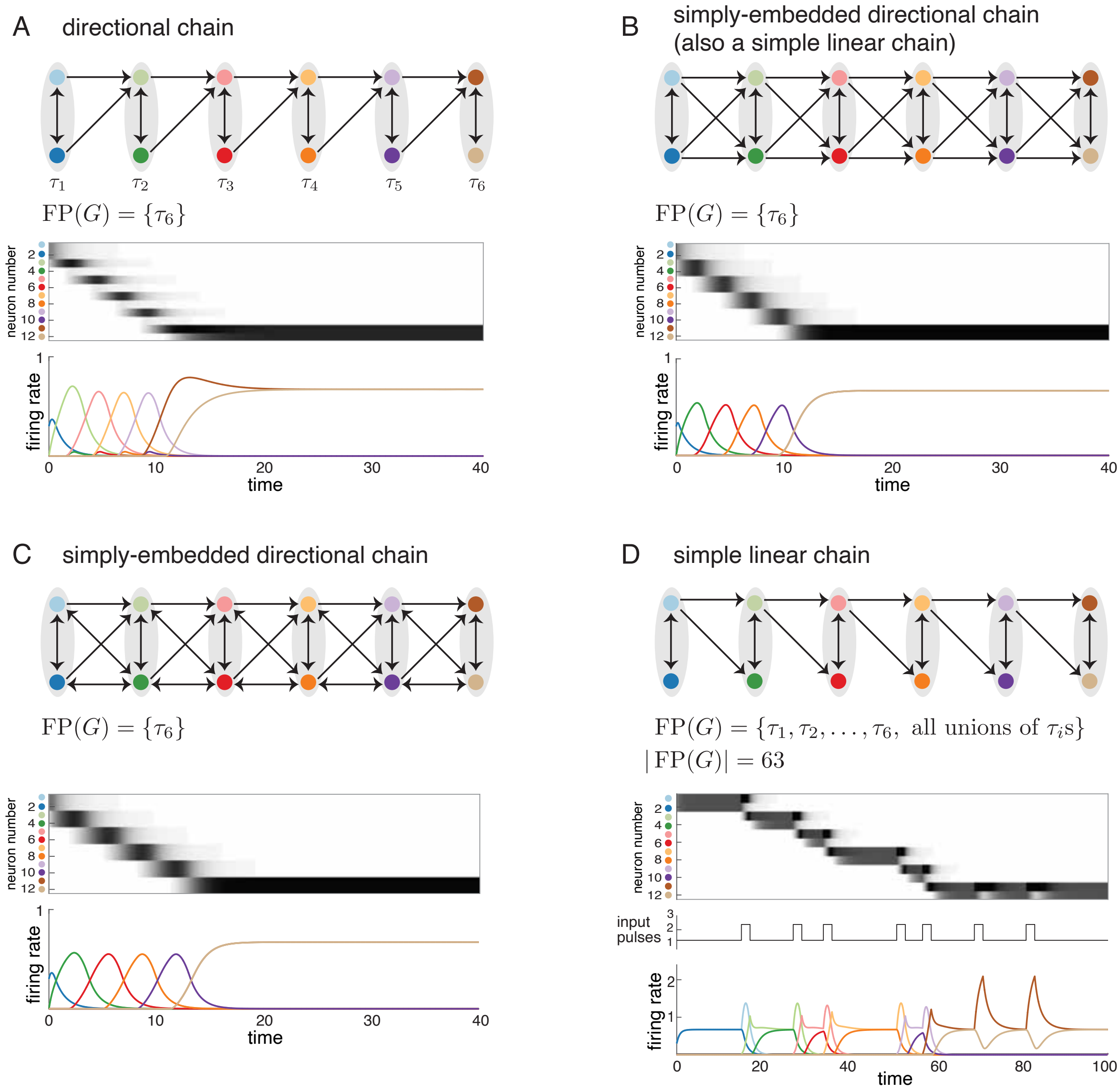


Figure 16. *Directional chains versus simple linear chains.* (A)–(C) *Graphs that are directional chains. Activity initialized on τ_1 flows through the chain, hitting each component in sequence, and converging on the nodes of τ_6.* (D) *A simple linear chain that is not directional. Each component clique supports a stable fixed point of the network. Unions of these component fixed point supports also yield fixed points. Activity initialized on τ_1 would stay indefinitely at the corresponding stable fixed point. Small kicks to the θ input (labeled as input pulses on the plot) can cause the activity to fall out of the current stable fixed point and move forward to converge onto τ_{i+1}. At time* 60*, the activity has converged to τ_6. After this point, all additional input pulses lead to increases in the activity of the nodes in τ_6, but the activity can never escape the final stable fixed point of the chain.*

by Proposition 2.8 since $\text{FP}(G) \subseteq \text{FP}(G|_{\tau_6}) = \{\tau_6\}$. Moreover, the dynamics progress forward down the directional chain and converge to the stable fixed point on τ_6, irrespective of any simply-embedded structure.

Combining simply-embedded partitions with directional chain structure does not yield any new information about $\text{FP}(G)$. But what about simply-embedded partitions in graphs that have a weaker chain-like structure, where all the edges feed forward between components, but there are not enough forward edges to guarantee directionality? For example, consider the graph in Figure 16D. All the edges between components feed forward following a chain-like architecture; moreover, each $j \in \tau_i$ treats the nodes in τ_{i+1} identically, and so $\{\tau_1|\cdots|\tau_6\}$ is a simply-embedded partition, but it is not directional. We refer to graphs with this chain-like architecture on a simply-embedded partition as *simple linear chains*.

Definition 3.5 (simple linear chain). *Let G be a graph with node partition $\{\tau_1|\cdots|\tau_N\}$. We say that G is a* simple linear chain *if the following two conditions hold:*

1. *the only edges between components go from nodes in τ_i to τ_{i+1}, and*
2. *for every $j \in \tau_i$, either $j \to k$ for every $k \in \tau_{i+1}$ or $j \not\to k$ for every $k \in \tau_{i+1}$.*

With simple linear chains, we are no longer guaranteed that the fixed points all collapse onto the last component. For example, in Figure 16D, each $\tau_i \in \text{FP}(G)$ since each clique survives in $G|_{\tau_i \cup \tau_{i+1}}$. Additionally, every union of τ_is is also a fixed point support. Since each surviving clique yields a stable fixed point, we see that the network dynamics in panel D do not naturally progress through the chain, but rather stabilize on an individual component. Interestingly, though, if we transiently kick all the neurons in the network by temporarily increasing θ, then the dynamics can escape from the current τ_i, and the activity flows forward and stabilizes on τ_{i+1}. The following theorem shows that the structure of $\text{FP}(G)$ illustrated in Figure 16D holds for simple linear chains more generally.

Theorem 3.6 (simple linear chains). *Let G be a simple linear chain with components $\tau_1, \ldots, \tau_N$.*

(i) *If $\sigma \in \text{FP}(G)$, then $\sigma_i \in \text{FP}(G|_{\tau_i}) \cup \{\emptyset\}$ for all $i \in [N]$, where $\sigma_i = \sigma \cap \tau_i$.*

(ii) *Consider a collection $\{\sigma_i\}_{i\in[N]}$ of $\sigma_i \in \text{FP}(G|_{\tau_i}) \cup \{\emptyset\}$. If additionally $\sigma_i \in \text{FP}(G|_{\tau_i \cup \tau_{i+1}}) \cup \{\emptyset\}$ for all $i \in [N]$, then*

$$\bigcup_{i\in[N]} \sigma_i \in \text{FP}(G).$$

In other words, $\text{FP}(G)$ is closed under unions of component fixed point supports that survive in $G|_{\tau_i \cup \tau_{i+1}}$.

Figure 17 illustrates Theorem 3.6 with an example simple linear chain. By Theorem 3.6(i), every fixed point support in $\text{FP}(G)$ restricts to a fixed point in $\text{FP}(G|_{\tau_i})$. Next consider a collection of σ_i such that $\sigma_i \in \text{FP}(G|_{\tau_i \cup \tau_{i+1}}) \cup \emptyset$ for all $i \in [N]$. First observe that each $\sigma_i \in \text{FP}(G|_{\tau_i \cup \tau_{i+1}})$ actually survives to the full network, and so $\sigma_i \in \text{FP}(G)$. This is guaranteed because σ_i has no outgoing edges to nodes outside of $\tau_i \cup \tau_{i+1}$ (Rule 2C). Moreover, by Theorem 3.6(ii), we see that every union of surviving component fixed points yields a fixed point of the full network, but additional fixed point supports are also possible.

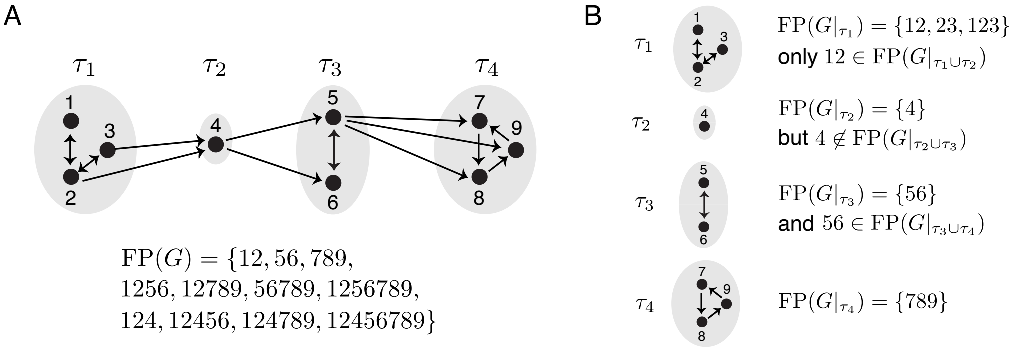

A
τ_1
τ_2
τ_3
τ_4
1
2
3
4
5
6
7
8
9
$\mathrm{FP}(G) = \{12, 56, 789,$
$1256, 12789, 56789, 1256789,$
$124, 12456, 124789, 12456789\}$
B
τ_1
$\mathrm{FP}(G|_{\tau_1}) = \{12, 23, 123\}$
only $12 \in \mathrm{FP}(G|_{\tau_1 \cup \tau_2})$
τ_2
$\mathrm{FP}(G|_{\tau_2}) = \{4\}$
but $4 \notin \mathrm{FP}(G|_{\tau_2 \cup \tau_3})$
τ_3
$\mathrm{FP}(G|_{\tau_3}) = \{56\}$
and $56 \in \mathrm{FP}(G|_{\tau_3 \cup \tau_4})$
τ_4
$\mathrm{FP}(G|_{\tau_4}) = \{789\}$

Figure 17. *Simple linear chain.* (A) *An example simple linear chain together with its* $\mathrm{FP}(G)$*. The first row of* $\mathrm{FP}(G)$ *gives the surviving fixed points from each component subgraph; the second row shows that all unions of these component fixed points are also in* $\mathrm{FP}(G)$ *(Theorem* 3.6(ii)*); the third row shows the additional fixed point supports in* $\mathrm{FP}(G)$ *that arise from the broader menu (Theorem* 3.6(i)*).* (B) $\mathrm{FP}(G|_{\tau_i})$ *for each component subgraph from* A*, and the list of which of these supports survive the addition of the next component in the chain.*

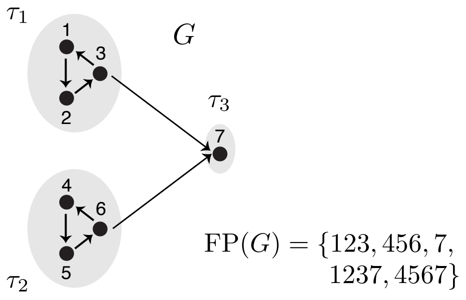


Figure 18. *Simple feedforward network. A feedforward network generalizing the conditions of the simple linear chain. Notice that* $\mathrm{FP}(G)$ *is* not *closed under unions of surviving component fixed points, since* 123, $456 \in \mathrm{FP}(G)$ *but* $123456 \notin \mathrm{FP}(G)$.

A natural generalization of simple linear chains is *simple feedforward networks* where G consists of ordered component subgraphs such that the only edges allowed between components are from a smaller numbered component to a larger one, and again we require that for any pair τ_i and τ_k with $k > i$, each $j \in \tau_i$ either sends edges to every node in τ_k or to no nodes in τ_k. Given that these simple feedforward networks have such similar structure to that of the simple linear chains, we might hope that an analogous result to Theorem 3.6 holds for these networks. These simple feedforward networks do have a simply-embedded partition structure, and so Theorem 3.6(i) holds for these networks as well (as an immediate corollary of Theorem 1.4). But an analogue of Theorem 3.6(ii) does *not* hold. Specifically, survival of component fixed points does not guarantee that the union of these component supports will yield a fixed point. Figure 18 provides an explicit counterexample: we see that the 3-cycles 123 and 456 both survive to $\mathrm{FP}(G)$ (by Rule 1), but their union $123456 \notin \mathrm{FP}(G)$ since it is uniform in-degree 1 with two outgoing edges to node 7.

3.4. Strongly simply-embedded partitions. Recall that $\{\tau_1|\cdots|\tau_N\}$ is a simply-embedded partition of a graph G if each component τ_i is simply-embedded in G so that every node in τ_i is treated identically by the rest of the graph; specifically, if any node outside of τ_i sends an edge to one node in τ_i, then it sends edges to *every* node in τ_i. In this context, there is still freedom allowing nodes to treat different components differently, e.g., node k may send edges to all nodes in τ_i, but send no edges to nodes in τ_j. In this subsection, we consider graphs with a more rigid partition structure, which we call a *strongly simply-embedded partition.* In these graphs, each node must treat all the components identically. More precisely, we have the following.

Definition 3.7 (strongly simply-embedded partition). *Let G be a graph with a partition of its nodes $\{\tau_1|\cdots|\tau_N\}$. The partition is called* strongly simply-embedded *if for every node j in G, either $j \to k$ for all $k \notin \tau_i$ or $j \not\to k$ for all $k \notin \tau_i$, where τ_i is the component containing j.*

Notice that in a strongly simply-embedded partition, each node j either projects edges onto every other node outside its component τ_i (in which case, we say that j is a *projector* onto $[n] \setminus \tau_i$) or it does not project any edges to nodes outside its component (in which case, we say that j is a *nonprojector* onto $[n] \setminus \tau_i$). The simplest examples of graphs with a strongly simply-embedded partition are *disjoint unions* and *clique unions*, which are building block constructions first studied in [11]. In a *disjoint union* of component subgraphs $G|_{\tau_1}, \ldots, G|_{\tau_N}$, there are no edges between components (see Figure 19A). In this case, every node in G is a

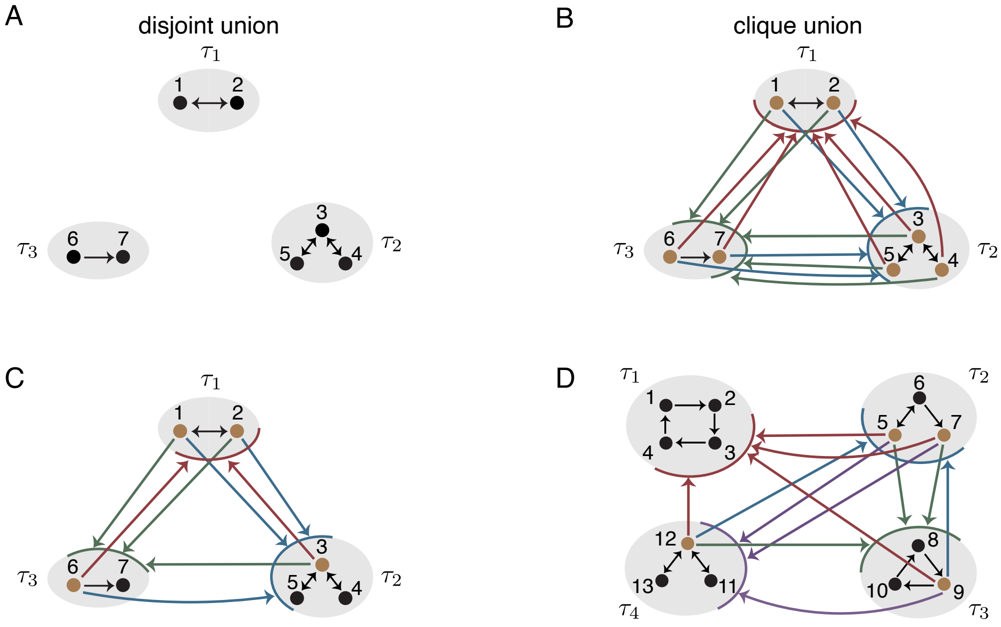


Figure 19. *Strongly simply-embedded partitions. Four example graphs with a strongly simply-embedded partition, characterized by the fact that each node treats all the other components identically. Thus, any node that sends an edge out to one component must in fact send edges out to every component (i.e., it must be a projector onto the rest of the graph). Projector nodes are colored brown.* (A) *A disjoint union.* (B) *A clique union.* (C)–(D) *Example graphs with a mix of projector and nonprojector nodes within each component.*

nonprojector onto the rest of the graph. At the other extreme, a *clique union* has bidirectional edges between every pair of nodes in different components. In a clique union, every node is a projector onto the rest of the graph (see Figure 19B). More generally, strongly simply-embedded partitions can have a mix of projector and nonprojector nodes even within the same component, as shown in Figures 19C and D (projector nodes are colored brown and have outgoing edges to every component).

Similar to simple linear chains, it turns out that strongly simply-embedded partitions also have the property that $\text{FP}(G)$ is closed under unions of surviving fixed point supports of the component subgraphs. With the added structure of the strongly simply-embedded partition, though, we can actually say something stronger—$\text{FP}(G)$ can be fully determined from knowledge of the component fixed point supports together with knowledge of which of those component fixed points survive in the full network. This complete characterization of $\text{FP}(G)$ is given in Theorem 3.8 below.

Theorem 3.8. *Suppose G has a strongly simply-embedded partition $\{\tau_1|\ldots|\tau_N\}$, and let $\sigma_i \overset{\text{def}}{=} \sigma \cap \tau_i$ for any $\sigma \subseteq [n]$. Then $\sigma \in \text{FP}(G)$ if and only if $\sigma_i \in \text{FP}(G|_{\tau_i}) \cup \{\emptyset\}$ for each $i \in [N]$, and either*

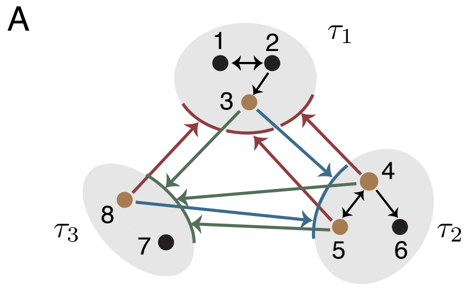
(a) *every σ_i is in $\text{FP}(G) \cup \{\emptyset\}$, or*

(b) *none of the σ_i are in $\text{FP}(G) \cup \{\emptyset\}$.*

In other words, $\sigma \in \text{FP}(G)$ if and only if σ either is a union of surviving fixed points σ_i, at most one per component, or is a union of dying fixed points, exactly one from every component.

A key to the proof of Theorem 3.8 is the significant additional constraints on the simply-embedded structure imposed by the strongly simply-embedded partition. Specifically, with a strongly simply-embedded partition, not only is the original partition $\{\tau_1|\cdots|\tau_N\}$ simply-embedded, but also *every coarsening* of the partition (where the components are unions of the τ_i) is simply-embedded. Notice this property does not hold in general for simply-embedded partitions. For example, given a cyclic union on $\{\tau_1|\cdots|\tau_N\}$, the coarser partition $\{\tau_1 \cup \tau_2 \mid \tau_3 \cup \cdots \cup \tau_N\}$ is not a simply-embedded partition since not all nodes in $\tau_1 \cup \tau_2$ are treated identically by the rest of the graph: the nodes in τ_1 receive edges from τ_N, while the nodes in τ_2 do not. The guarantee of the simply-embedded property for every coarser partition enables an inductive proof to fully nail down $\text{FP}(G)$ for strongly simply-embedded partitions.

As an application of Theorem 3.8, we can immediately recover characterizations of the fixed points of disjoint unions and clique unions previously given in [11, Theorems 11 and 12]. In a disjoint union, every component fixed point support survives to the full network since it has no outgoing edges (by Rule 2: inside-out domination). Thus, for a disjoint union, $\text{FP}(G)$ consists of all the fixed points of type (a) from Theorem 3.8: unions of (surviving) component fixed points σ_i, at most one per component. In contrast, in a clique union, every component fixed point support dies in the full network since it has a target that outside-in dominates it (in fact, every node outside of τ_i is a target of any subset of τ_i). Thus, for a clique union, $\text{FP}(G)$ consists of all the fixed points of type (b): unions of (dying) component fixed points σ_i, exactly one from every component. Both the disjoint union and the clique union characterizations of $\text{FP}(G)$ [11, Theorems 11 and 12] are now immediate corollaries of Theorem 3.8, and the earlier proofs of these results in [11] have a similar flavor to the proof of Theorem 3.8, which we provide in the appendix, section SM1.6.



B

$$\mathrm{FP}(G|_{\tau_1}) = \{12, 3, 123\} \qquad 12, 123 \in \mathrm{FP}(G)$$

$$\mathrm{FP}(G|_{\tau_2}) = \{45, 6, 456\} \qquad 6 \in \mathrm{FP}(G)$$

$$\mathrm{FP}(G|_{\tau_3}) = \{7, 8, 78\} \qquad 7 \in \mathrm{FP}(G)$$

$$\mathrm{FP}(G) = \{12, 123, 6, 7,$$
$$126, 127, 1267, 1236, 1237, 12367, 67,$$
$$3458, 34578, 34568, 345678\}$$

Figure 20. *Strongly simply-embedded partition with* $\mathrm{FP}(G)$. (A) *A graph with a strongly simply-embedded partition* $\{\tau_1|\tau_2|\tau_3\}$*. Projector nodes are colored brown.* (B) *(Top)* $\mathrm{FP}(G|_{\tau_i})$ *for each component subgraph together with the supports from each component that survive within the full graph. (Bottom)* $\mathrm{FP}(G)$ *for the strongly simply-embedded partition graph. The first two lines of* $\mathrm{FP}(G)$ *consist of unions of surviving fixed points, at most one per component. The third line gives the fixed points that are unions of dying fixed point supports, exactly one from every component.*

Corollary 3.9. *Let G be a graph with partition $\{\tau_1|\cdots|\tau_N\}$.*

(a) *If G is a disjoint union of $G|_{\tau_1}, \ldots, G|_{\tau_N}$, then $\sigma \in \mathrm{FP}(G)$ if and only if $\sigma_i \in \mathrm{FP}(G|_{\tau_i}) \cup \{\emptyset\}$ for all $i \in [N]$.*

(b) *If G is a clique union of $G|_{\tau_1}, \ldots, G|_{\tau_N}$, then $\sigma \in \mathrm{FP}(G)$ if and only if $\sigma_i \in \mathrm{FP}(G|_{\tau_i})$ for all $i \in [N]$.*

More generally, though, a strongly simply-embedded partition can have a mix of surviving and dying component fixed points, so that $\mathrm{FP}(G)$ has a mix of both type (a) and type (b) fixed point supports. Figure 20A gives an example of a strongly simply-embedded partition, and panel B shows both the set of component fixed point supports, $\mathrm{FP}(G|_{\tau_i})$ and the subset of those that survive to yield fixed points of the full network. Since there are dying fixed points in every component, we see that $\mathrm{FP}(G)$ has a mix of both type (a) and type (b) fixed point supports.

4. Applications to sequential attractor prediction. In this section, we consider a number of networks of size $n = 5$ to show how directional cycle graph architecture is predictive of the structure of corresponding sequential attractors, particularly when a graph has a simply-embedded directional cycle representation. From this analysis, we see that these graph structures are useful not only for predicting the set of fixed points of a network, but also for explicitly connecting architecture to the pattern of sequential neural activity that emerges in attractors.

We focus our analysis on a subset of the graphs of size $n = 5$ with the special property that $\mathrm{FP}(G) = \{12345\}$, i.e., the graphs have a unique fixed point, which has full support. In prior work [32], these types of networks, known as *core motifs*, were observed to produce attractors in which all the neurons are highly active, as opposed to the activity being concentrated on only a subnetwork of neurons. As a result, the global connectivity is relevant to shaping the activity of all attractors in these networks, and thus they are particularly well suited to analysis with the directional cycle framework. The networks analyzed here were taken from [9], which provided a comprehensive analysis of all the attractors of CTLNs of size $n = 5$ with

$\varepsilon = 0.51$, $\delta = 1.76$, $\theta = 1$; we follow the numbering of graphs used there and focus on the sequential attractors that emerge for that choice of parameters.

Figure 21 shows a sampling of graphs drawn in a way that highlights their directional cycle structure. Observe that for the cyclic unions and the more general simply-embedded directional cycles, the corresponding attractor has activity that flows through the components in precisely the cyclic order dictated by the directional cycle. For components that are cliques, the activity of the nodes in the clique is perfectly synchronized and thus only the activity of the highest-numbered node is visible in the plots. Also notice that the activity of nodes in a synchronous component is typically lower than that of the singleton components because there is competition among the nodes within a component, and the total population activity is bounded. When a component is not a clique, as in graph 23, we see that the activity may

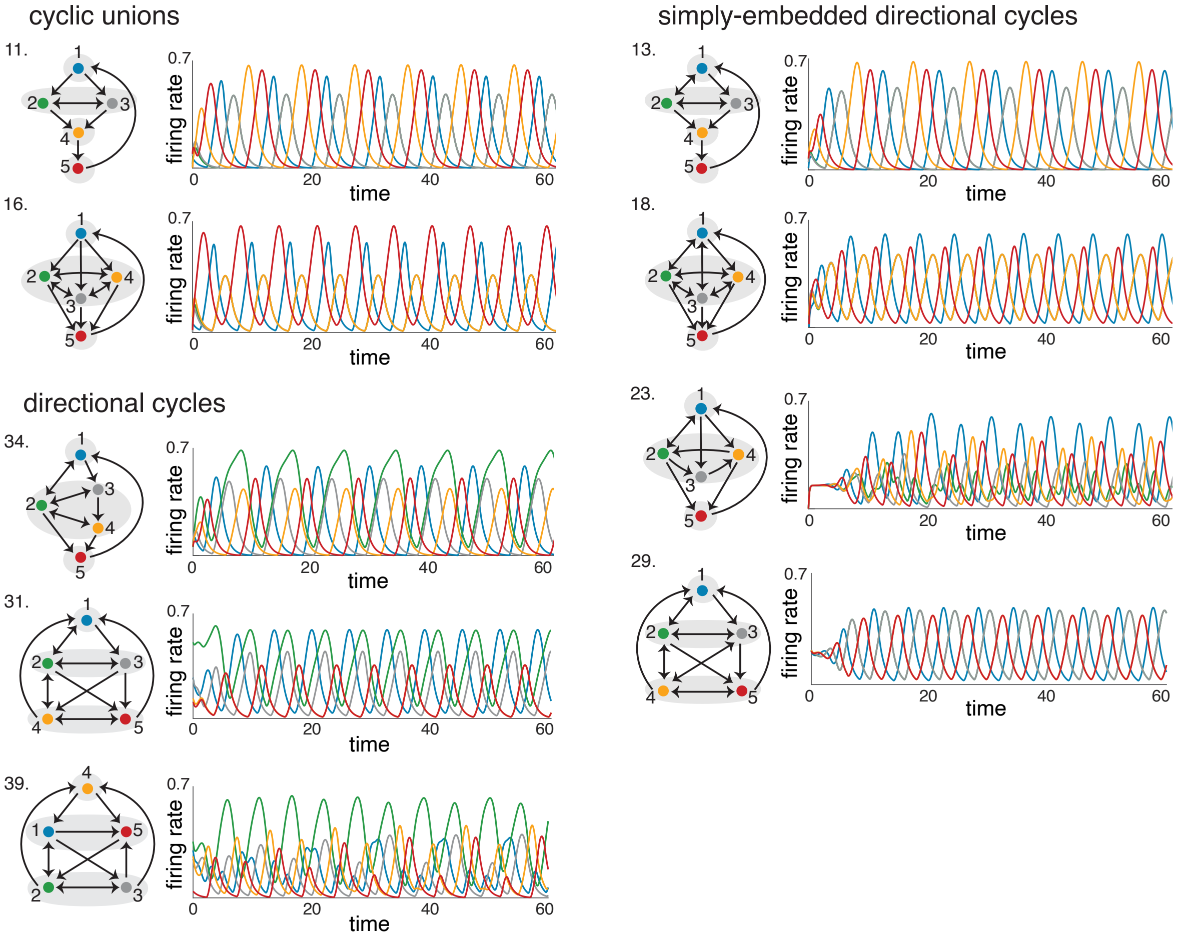


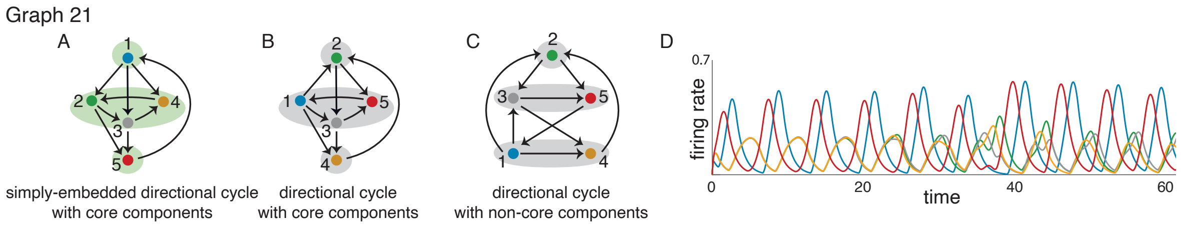
Figure 21. *Example graphs of size* 5. *A collection of example* $n = 5$ *graphs of different types and their corresponding dynamic attractors for* $\varepsilon = 0.51, \delta = 1.76, \theta = 1$. *The graphs are numbered following the ordering given in* [9], *which extensively catalogued* $\mathrm{FP}(G)$ *and the dynamic attractors for all graphs of size* 5 *for this parameter choice.*

cycle through the nodes in the component before the activity flows on to the next component in the directional cycle.

When a graph only has directional cycle representations that do not coincide with a simply-embedded partition, this architecture can still give insight into the sequential attractor, but this is not guaranteed. For example, for graph 34 in Figure 21, we see that the neural activity cycles through the components in the cyclic order of the directional cycle, despite the absence of a simply-embedded partition. In this attractor, there is interesting activity among nodes $2, 3$, and 4 (the second component): the firing rate curve of node 2 forms an "envelope" over those of 3 and 4 as a result of the bidirectional edges $2 \leftrightarrow 3$ and $2 \leftrightarrow 4$. For graph 31, we see that the activity also appears to cycle through the components in cyclic order, but the activity in the second component is not synchronized, despite this component being a clique; specifically, node 2 fires at a higher rate than 3, although they both peak at roughly the same time. This is likely because node 2 receives an extra input from node 4 that node 3 does not receive, which is precisely why this directional cycle representation does not correspond to a simply-embedded partition. Finally, graph 39 has a single directional cycle representation, but this structure does not appear to give any insight into the pattern of neural activity in the corresponding attractor.

It is worth noting that a given graph may have multiple directional cycle representations. Figure 22 shows the three directional cycle representations of graph 21 together with its sequential attractor. We see that the simply-embedded directional cycle is the best predictor of the attractor since it not only reflects the sequence in which the nodes will fire, but it also predicts that nodes 1 and 5 will be the highest firing, as they are the singleton components. In Figure 21, we have the same phenomenon for graph 13: it has seven directional cycle representations, but only the one shown is simply-embedded, and this is the only representation that reflects the activity in the attractor.

From these examples, it appears that directional cycle architecture and simply-embedded partitions are useful for understanding the structure of the sequential attractors of a network. The role of simply-embedded partitions is somewhat surprising given that this structure was primarily proposed for its utility in narrowing the set of possible fixed point supports, rather than for any explicit expected impact on neural activity. One possible explanation for the



simply-embedded directional cycle with core components

directional cycle with core components

directional cycle with non-core components

Figure 22. *The three directional cycle representations of graph* 21. (A) *The simply-embedded directional cycle representation of graph* 21. (B)–(C) *Directional cycle representations that are not from a simply-embedded partition.* (D) *The global attractor of the CTLN for graph* 21 *with* $\varepsilon = 0.51, \delta = 1.76, \theta = 1$. *The sequence of neural firing matches that of both of the directional cycle representations from* A *and* B. *But the structure of the attractor (with neurons* 1 *and* 5 *high firing) is best represented by the simply-embedded directional cycle from* A, *since singleton components yield the high-firing neurons in directional cycles.*

importance of simply-embedded partitions in predicting dynamics is that in such a partition, every node within a component receives identical inputs from the rest of the graph. This makes it more likely that nodes within a simply-embedded component will fire together or at least in close sequence in the corresponding attractor. Thus, whenever a graph has a nontrivial simply-embedded partition, only directional cycle structures that respect that partition are likely to predict the sequential structure of the attractor.

5. Conclusion.

In this work, we investigated different architectures that give rise to sequential attractors. We considered cyclic unions and two generalizations of cyclic union architecture: directional cycles and simply-embedded partitions. Computationally, we have seen that directional cycles yield sequential attractors that traverse the components in cyclic order. While we cannot yet prove that directional cycles produce these cyclic dynamics, Theorem 1.2 showed that every fixed point support contains a cycle that hits every component in cyclic order. Moreover, we have proven even stronger results about the fixed points of cyclic unions, simply-embedded partitions, and simply-embedded directional cycles (Theorems 1.1, 1.4, and 1.5).

Additionally, in section 4, we analyzed the sequential attractors of various networks of size $n = 5$ through the lens of directional cycle and simply-embedded architectures. We found that whenever a graph has a simply-embedded directional cycle structure, this architecture nicely predicts the sequential activity of the dynamic attractor. Moreover, directional cycles can provide insight into the expected sequential activity even when the partition is not simply-embedded. Thus, we conclude that directional cycle and simply-embedded architectures provide a useful framework for predicting (as well as engineering) persistent sequential activity in threshold-linear networks.

REFERENCES

[1] M. Abeles, *Local Cortical Circuits: An Electrophysiological Study*, Springer, Berlin, 1982.

[2] M. Arriaga and E. B. Han, *Dedicated hippocampal inhibitory networks for locomotion and immobility*, J. Neurosci., 37 (2017), pp. 9222–9238.

[3] Y. Aviel, E. Pavlov, M. Abeles, and D. Horn, *Synfire chain in a balanced network*, Neurocomput., 44 (2002), pp. 285–292.

[4] A. Bel, R. Cobiaga, W. Reartes, and H. G. Rotstein, *Periodic solutions in threshold-linear networks and their entrainment*, SIAM J. Appl. Dyn. Syst., 20 (2021), pp. 1177–1208.

[5] T. Biswas and J. E. Fitzgerald, *A Geometric Framework to Predict Structure from Function in Neural networks*, https://arxiv.org/abs/2010.09660, 2022.

[6] D. A. Burke, H. G. Rotstein, and V. A. Alvarez, *Striatal local circuitry: A new framework for lateral inhibition*, Neuron, 96 (2017), pp. 267–284.

[7] L. Carillo-Reid, J. K. Miller, J. P. Hamm, J. Jackson, and R. Yuste, *Endogenous sequential cortical activity evoked by visual stimuli*, J. Neurosci., 35 (2015), pp. 8813–8828.

[8] L. L. Colgin, *Rhythms of the hippocampal network*, Nat. Rev. Neurosci., 17 (2016), pp. 239–249.

[9] C. Curto, K. Morrison, C. Parmelee, S. Garai, and J. Paik, *n5-graphs-package*, https://github.com/ccurto/n5-graphs-package.

[10] C. Curto, J. Geneson, and K. Morrison, *Stable Fixed Points of Combinatorial Threshold-Linear Networks*, https://arxiv.org/abs/1909.02947, 2019.

[11] C. Curto, J. Geneson, and K. Morrison, *Fixed points of competitive threshold-linear networks*, Neural Comput., 31 (2019), pp. 94–155.

[12] C. Curto and K. Morrison, *Pattern completion in symmetric threshold-linear networks*, Neural Comput., 28 (2016), pp. 2825–2852.

[13] C. Curto, A. Degeratu, and V. Itskov, *Encoding binary neural codes in networks of threshold-linear neurons*, Neural Comput., 25 (2013), pp. 2858–2903.

[14] C. Curto, A. Degeratu, and V. Itskov, *Flexible memory networks*, Bull. Math. Biol., 74 (2012), pp. 590–614.

[15] D. Egas Santander, S. Ebli, A. Patania, N. Sanderson, F. Burtscher, K. Morrison, and C. Curto, *Nerve theorems for fixed points of neural networks*, in Research in Computational Topology 2, Assoc. Women Math. Ser. 30, E. Gasparovic, V. Robins, and K. Turner, eds., Springer, Cham, 2022.

[16] V. Ego-Stengel and M. A. Wilson, *Disruption of ripple-associated hippocampal activity during rest impairs spatial learning in the rat*, Hippocampus, 20 (2010), pp. 1–10.

[17] E. Fino and R. Yuste, *Dense inhibitory connectivity in neocortex*, Neuron, 69 (2011), pp. 1188–1203.

[18] G. Girardeau, K. Benchenane, S. I. Weiner, G. Buzsáki, and M. B. Zugaro, *Selective suppresion of hippocampal ripples impairs spatial memory*, Nature Neurosci., 12 (2009), pp. 1222–1223.

[19] S. Grillner and P. Wallén, *Cellular bases of a vertebrate locomotor system—steering, intersegmental and segmental co-ordination and sensory control*, Brain Res. Rev., 40 (2002), pp. 92–106.

[20] B. Haider, M. Hausser, and M. Carandini, *Inhibition dominates sensory responses in the awake cortex*, Nature, 493 (2013), pp. 97–100.

[21] G. Hayon, M. Abeles, and D. Lehmann, *A model for representing the dynamics of a system of synfire chains*, J. Comput. Neurosci., 18 (2005).

[22] R. H. Hahnloser, H. S. Seung, and J. J. Slotine, *Permitted and forbidden sets in symmetric threshold-linear networks*, Neural Comput., 15 (2003), pp. 621–638.

[23] 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), pp. 947–951.

[24] V. Itskov, C. Curto, E. Pastalkova, and G. Buzsáki, *Cell assembly sequences arising from spike threshold adaptation keep track of time in the hippocampus*, J. Neurosci., 31 (2011), pp. 2828–2834.

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

[26] A. Luczak, P. Barthó, S. L. Marguet, G. Buzsáki, and K. D. Harris, *Sequential structure of neocortical spontaneous activity in vivo*, Proc. Natl. Acad. Sci. USA, 104 (2007), pp. 347–352.

[27] E. Marder and D. Bucher, *Central pattern generators and the control of rhythmic movements*, Curr. Bio., 11 (2001), pp. R986–996.

[28] P. Malerba, G. P. Krishnan, J-M. Fellous, and M. Bazhenov, *Hippocampal CA*1 *ripples as inhibitory transients*, PLoS Comput. Biol., 12 (2016).

[29] K. Morrison and C. Curto, *Predicting neural network dynamics via graphical analysis*, in Algebraic and Combinatorial Computational Biology, R. Robeva and M. Macaulay eds., Elsevier, Amsterdam, 2018.

[30] K. Morrison, A. Degeratu, V. Itskov, and C. Curto, *Diversity of Emergent Dynamics in Competitive Threshold-Linear Networks: A Preliminary Report*, https://arxiv.org/abs/1605.04463, 2016.

[31] E. Pastalkova, V. Itskov, A. Amarasingham, and G. Buzsáki, *Internally generated cell assembly sequences in the rat hippocampus*, Science, 321 (2008), pp. 1322–1327.

[32] C. Parmelee, S. Moore, K. Morrison, and C. Curto, *Core motifs predict dynamic attractors in combinatorial threshold-linear networks*, PLOS ONE, 17 (2022), e0264456, https://doi.org/10.1371/journal.pone.0264456.

[33] E. Stark, L. Roux, R. Eichler, and G. Buzsáki, *Local generation of multineuronal spike sequences in the hippocampal CA*1 *region*, Proc. Natl. Acad. Sci. USA, 112 (2015), pp. 10521–10526.

[34] H. S. Seung and R. Yuste, *Appendix* E*: Neural networks*, in Principles of Neural Science, 5th ed., McGraw-Hill, New York, 2012, pp. 1581–1600.

[35] M. A. Whittington, R. D. Traub, N. Kopell, B. Ermentrout, and E. H. Buhl, *Inhibition-based rhythms: Experimental and mathematical observations on network dynamics*, Int. J. Psychophysiol., 38 (2000), pp. 315–336.

[36] X. Xie, R. H. Hahnloser, and H. S. Seung, *Selectively grouping neurons in recurrent networks of lateral inhibition*, Neural Comput., 14 (2002), pp. 2627–2646.

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