

**Network hierarchy and pattern recovery in directed sparse Hopfield networks**Niall Rodgers <sup>\*</sup>*School of Mathematics, University of Birmingham, Birmingham B15 2TT, United Kingdom  
and Topological Design Centre for Doctoral Training, University of Birmingham, Birmingham B15 2TT, United Kingdom*Peter Tiño *School of Computer Science, University of Birmingham, Birmingham B15 2TT, United Kingdom*

Samuel Johnson

*School of Mathematics, University of Birmingham, Birmingham B15 2TT, United Kingdom  
and The Alan Turing Institute, British Library, 96 Euston Road, London NW1 2DB, United Kingdom*

(Received 1 February 2022; accepted 22 May 2022; published 13 June 2022)

Many real-world networks are directed, sparse, and hierarchical, with a mixture of feedforward and feedback connections with respect to the hierarchy. Moreover, a small number of master nodes are often able to drive the whole system. We study the dynamics of pattern presentation and recovery on sparse, directed, Hopfield-like neural networks using trophic analysis to characterize their hierarchical structure. This is a recent method which quantifies the local position of each node in a hierarchy (trophic level) as well as the global directionality of the network (trophic coherence). We show that even in a recurrent network, the state of the system can be controlled by a small subset of neurons which can be identified by their low trophic levels. We also find that performance at the pattern recovery task can be significantly improved by tuning the trophic coherence and other topological properties of the network. This may explain the relatively sparse and coherent structures observed in the animal brain and provide insights for improving the architectures of artificial neural networks. Moreover, we expect that the principles we demonstrate here, through numerical analysis, will be relevant for a broad class of system whose underlying network structure is directed and sparse, such as biological, social, or financial networks.

DOI: [10.1103/PhysRevE.105.064304](https://doi.org/10.1103/PhysRevE.105.064304)**I. INTRODUCTION**

Models of the brain provided the original inspiration for the invention of artificial neural networks. However, biological neural networks have a much richer structure than their artificial counterparts. In particular, they are not exclusively feedforward like conventional deep network architectures, yet there is a direction to information processing, unlike in recurrent network models. For example, the neural network of the nematode *C. elegans* [1,2], the only animal nervous system to have been fully mapped at the level of neurons and synapses, is quite sparse and displays a nontrivial mix of feedforward and feedback connections, as revealed by a recent technique from the field of complex networks called trophic analysis [3]. What might explain this particular neural-network architecture? We address this question by studying the relationship between trophic structure and the dynamics of a simple model which we refer to as a Hopfield-like neural network.

Trophic analysis, inspired by ecological networks, assigns to each node a trophic level, which can be regarded as a position in a hierarchy, and measures the trophic coherence of the whole network, a property which indicates to what extent this hierarchy is well defined, conferring on the network an

overall directionality. In this work we take the convention that the bottom of the hierarchy is where information enters the system, just as energy flows up from plants in a food web. This may be different in other fields, for example, in the study of hierarchical trees, but all definitions are equivalent, including relabeling top and bottom or by reversing the edge directions. When the *C. elegans* neural network is visualized so as to show the trophic level of each neuron, as in Fig. 1, it is observed that while most of the synapses are consistent with an overall direction, there are some which feed back as in a recurrent architecture. In fact, when the trophic coherence is calculated, it lies exactly halfway between a maximally coherent (i.e., entirely feedforward) network and one which is entirely incoherent (fully recurrent). Moreover, it has been shown previously that this level of coherence amounts to a significant deviation from the kind of networks which arise from random graph models such as the Erdős-Rényi model [6,7].

How the dynamics and hierarchy interact is demonstrated in this paper by performing a pattern recognition task (described in detail in the next section) on network architectures which span a range of hierarchical structures. We find that trophic coherence is very strongly linked to the ability to correctly recognize and display the pattern shown. Maximally coherent networks lack the feedback to store patterns, while maximally incoherent networks are unable to change state

<sup>\*</sup>NXR081@student.bham.ac.uk

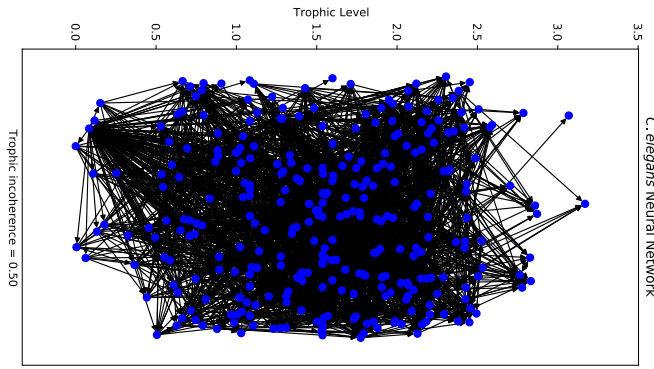


FIG. 1. Illustration of the real-world connectome of *C. elegans* which has intermediate incoherence with node height drawn using trophic levels. Data were taken from [4]. The figure was drawn using the NETWORKX graph package [5].

when presented with fractions of new patterns. The optimal configuration is intermediate coherence, a mixture of feed-forward and feedback structure which is shared with many biological systems. A similar result was reported in Ref. [8] for a system in which elements followed majority rule dynamics where the stability of the *C. elegans* neural network was analyzed using trophic analysis. We concentrate our study on synthetic networks inspired by this network which can be made dense enough to store multiple pattern states and lack basal nodes (nodes with no in-degree) which would act as input to the system without being influenced by it.

There are clear differences between the structures of biological neural networks and artificial recurrent neural networks, such as standard implementations of the Hopfield model. Biological networks are sparse, whereas the artificial versions are often based on complete or very dense graphs. They are also directed, since chemical synapses have a pre- and a postsynaptic neuron [9], while some models such as that of Hopfield tend to assume symmetric synapses in order to avoid the possible periodic or chaotic behavior associated with asymmetric interactions [10] or to align with experimental data limited to the undirected case [9].

In addition, in nature there are a limited number of sensory neurons which receive information directly from the outside world, a fact not usually replicated in Hopfield models. However, it is possible to implement a Hopfield-like model on sparse directed networks and to present stimuli only to a subset of neurons, as we go on to do here in order to investigate how the dynamics is affected by modifying the trophic structure.

Feedforward artificial neural networks, such as those used in deep learning, in these respects resemble more closely the architectures of biological neural networks, at least in the case of nature's only fully mapped connectome, that of *C. Elegans*. The main difference is that deep neural networks tend to be maximally coherent, with each layer corresponding to a distinct (integer) trophic level.

We show, through numerical analysis, that network hierarchy can be exploited in order to use a small subset of neurons to drive the system, with how well a pattern is recovered being strongly influenced by where in the hierarchy it is received.

Hierarchical structure creates heterogeneous dynamics with different parts of the network recovering patterns differently. Additionally, we show that by preferentially adding edges to lower-level nodes, pattern recovery can be made more consistent. This has potential applications for how artificial neural networks are designed [11,12] as well as for controllability of dynamics on general directed complex networks [13–15], which could range from biological neural networks [16,17] to ecosystems, economies [18], or the Internet [19]. In particular, Hopfield networks have recently been used to model gene regulatory networks [20,21]. We will therefore use this model to highlight principles which may be of general application to any system wired according to a directed network.

## II. USING TROPHIC ANALYSIS TO QUANTIFY NETWORK HIERARCHY

Trophic analysis is a method of quantifying the hierarchy of nodes and the global directionality of a directed complex network, first introduced by Johnson *et al.* [22], which is based on the ecological concept of trophic level [23]. A directed network, or graph, can be represented via an adjacency matrix, defined as

$$A_{ij} = \begin{cases} 1 & \text{if there exists an edge } i \rightarrow j \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Unlike in undirected networks, this matrix is not necessarily symmetric,  $A_{ij} \neq A_{ji}$ . Directed networks have the additional complexity of the notion of in- and out-degrees, where the in-degree is the number of incoming edges a vertex receives and the out-degree is the number of edges leaving a vertex. In undirected networks the in- and out-degrees coincide. Directed networks can also be weakly or strongly connected. Weakly connected means that there is a path between all pairs of vertices if you ignore the edge directions, while strongly connected means there is such a path respecting the edge directions. The networks studied in this work are all weakly connected but may not be strongly connected.

Trophic analysis was recently extended and redefined to cover more general networks [3], removing the requirement that networks must have basal nodes (nodes with in-degree 0). This is the definition that will be used in this work. Trophic structure has been used to study spreading processes in neural and epidemiological settings [18], infrastructure [24,25], and the structure of organizations [26]. Trophic analysis is composed of two parts: the node level information, i.e., the trophic level, which describes where each node sits in the overall hierarchy of a network; and the global information of how directed, or coherent, the overall network is. The idea of trophic level arises from ecology where the lowest trophic level nodes represent plants which sit at the bottom of the network hierarchy and the highest trophic level nodes are carnivores at the top of the food chain. The trophic level can be calculated for a network of  $N$  nodes by solving the  $N \times N$  matrix equation

$$\Lambda h = v, \quad (2)$$

where  $h$  is the vector of trophic levels,  $v$  is the imbalance of in-degree and out-degree of a node,  $v_i = k_i^{\text{in}} - k_i^{\text{out}}$ , and  $\Lambda$  is

the Laplacian matrix

$$\Lambda = \text{diag}(u) - A - A^T. \quad (3)$$

This depends on the sum of the in- and out-degrees of each node  $u_i = k_i^{\text{in}} + k_i^{\text{out}}$  and the adjacency matrix  $A$  of the graph and its transpose  $A^T$ . This definition can also be extended to cover weighted adjacency matrices [3]. The solutions to Eq. (2) can be modified by adding a constant vector since  $\Lambda$  acting on a constant vector is zero. This allows the minimum level to be set at zero by convention and fully coherent networks to have integer levels.

Trophic coherence is based upon the distribution of trophic levels of the nodes in a network. How coherent or incoherent a network is can be described by the parameter

$$F = \frac{\sum_{ij} A_{ij} (h_j - h_i - 1)^2}{\sum_{ij} A_{ij}}. \quad (4)$$

We call  $F$  the trophic incoherence such that when  $F = 0$  the network is completely coherent and when  $F = 1$  it is completely incoherent. This depends on the levels of each node  $h_i$  and the entries of the adjacency matrix  $A_{ij}$ . Loosely speaking,  $F$  quantifies, per connection in the graph, to what degree the connections  $i \rightarrow j$  are not one-step connections in the order of trophic levels, i.e., by how much  $h_j - h_i$  differs (in the mean square sense) from 1. In principle, these could have positive weights, but throughout this work we will take the entries of the adjacency matrix to always be 0 or 1 to avoid confusion with the trained weights associated with the neural network. A network for which  $F = 0$  is acyclic and completely free from any feedback, with the amount of feedback and cycles growing as this parameter increases to 1 [3]. This is reflected in results showing an increase in spectral radius and a reduction in the deviation from normality of the adjacency matrix, how far the matrix is from commuting with its transpose, as incoherence increases [3,7].

Note that the levels  $h$ , defined by Eq. (2), can be regarded as the argument which minimizes  $F$ , as given by Eq. (4) [3]. One can therefore think of the trophic levels of a network as those which maximize its trophic coherence which relates to how it was derived in [3].

### III. HOPFIELD-LIKE NETWORKS

The Hopfield model is a recurrent neural-network model which is very similar to the Ising model studied in statistical physics [10]. The neurons can take binary states  $+1$  or  $-1$ . Due to similarity to the Ising model, these neuron states are sometimes referred to as spins and the order parameter measuring the state of the system can be referred to as a magnetization. A Hopfield network can store binary memories, or patterns, by setting the weights of connections between neurons such that when an update rule is applied the system moves across an energy landscape to its attractors, which correspond to the stored patterns. This system can, in some cases, be studied via mean-field theory or other theoretical methods [27]. In our case, however, due to the asymmetric connections and complex network topology, we will use numerical simulations.

We want the system to update in such a way that it moves towards the minima in the energy landscape defined by

$$E = - \sum_{ij} w_{ij} A_{ij} s_i s_j, \quad (5)$$

where  $w_{ij}$  is the coupling between neurons  $i$  and  $j$ , which may be positive or negative depending on patterns stored. The states of the neurons take values  $s_i = \pm 1$  and  $A_{ij}$  are the elements of the adjacency matrix, as defined by Eq. (1). There are many possible update rules which can achieve the desired behavior, such as the Metropolis-Hastings algorithm [28]. We use a sigmoid probability function such that

$$s_i(t + \Delta t) = -s_i(t) \quad (6)$$

with

$$\text{Prob} = \frac{1}{1 + \exp \frac{\Delta E}{T}}, \quad (7)$$

where  $\Delta E$  is the energy change associated with flipping the neuron state and  $T$  is a temperature parameter which makes the system stochastic. To reduce complexity and uncertainty, the results we present here are for a temperature very close to zero,  $T = 10^{-5}$ , so the dynamics is essentially deterministic and equivalent to using the sign of the incoming field, the sum of the states of the in neighbors, as the update rule. The system can therefore be referred to as Hopfield-like, or simply as a Hopfield network, which is generally taken to be deterministic, as opposed to Boltzmann machines, which are stochastic [29]. However, even in this regime the asymmetry in  $A$  leads to a range of surprising behaviors not observed in undirected networks [30].

Updates to the system can be made in parallel or asynchronously. We use a parallel update rule, which allows for complex behavior such as limit cycles [13].

#### A. Training the network

Setting the weights so that the attractors of the system correspond to the random binary patterns we wish to store in the network is a key part of the process. The traditional method of setting weights in a Hopfield network so that the network recalls the desired patterns is Hebb's rule [31]. This is often summarized as "neurons that fire together wire together." That is, if two neurons have the same state in a particular pattern the connection between them is strengthened and if they are in opposite states it is decreased. For learning  $P$  patterns, where for each pattern each neuron has a fixed state  $\xi_i^p = \pm 1$ , the rule sets the weights as

$$w_{ij} = \frac{1}{P} \sum_{p=1}^P \xi_i^p \xi_j^p. \quad (8)$$

This very simple rule works and can be used on any network topology. It has the benefit of being a one-shot rule in that it only requires one loop over the set of patterns to train the network. However, it suffers from the fact that on a graph which is not complete the information about the correlations between disconnected neurons is not used. We found during initial tests that on very sparse directed networks the memory capacity of the network was substantially reduced. This is very similar

**Algorithm 1.** Iterative Hebb rule [32].

---



---

```

Set the initial weights  $w_{ij} = 0$  for all nodes  $i, j$ .
Set the stop condition flag,  $\text{flag} = 0$ .
Set the step counter,  $\text{steps} = 0$ .
while  $\text{flag} = 0$  and  $\text{steps} < \text{steps}_{\max}$  do
   $\text{flag} = 1$ ;
  for  $p$  in range  $P$  do
    for  $i$  in range  $N$  do
       $\text{field} = 0$ ;
      for  $j$  in range  $N$  do
         $\text{field} \leftarrow \text{field} + A_{ji} w_{ji} \xi_j^p$ 
      end
      if  $\text{field} \times (\xi_i^p) < \delta$  then
        for  $q$  in range  $N$  do
           $w_{qi} \leftarrow w_{qi} + \frac{A_{qi} \xi_q^p \xi_i^p}{N}$ ;
           $\text{flag} = 0$ 
        end
      end
    end
  end
   $\text{steps} \leftarrow \text{steps} + 1$ 
end

```

---



---

to the finding of Tanaka *et al.* [32] for undirected networks. They remedy this issue by adopting an iterative version of Hebb's rule based on earlier work [33,34], which was found to increase capacity substantially, with other similar results noted in the literature [35]. For the remainder of this work we implement this rule [32]. Both the original Hebb rule and the adapted version are local, in that synaptic weights are updated using only information from the pre- and postsynaptic neurons, as also happens, we believe, in the brain [36].

The iterative Hebb rule works to set the weights so that every pattern corresponds to a local minimum of the energy landscape where updates of the system stop. This condition can be expressed as

$$\xi_i^p \left( \sum_j A_{ji} w_{ji} \xi_j^p \right) \geq \delta \quad (9)$$

for all  $P$  patterns and  $N$  nodes, and  $\delta$  a positive constant. This means that at each node, for every pattern the polarities of the state and the incoming field are the same. As a result it is always energetically unfavorable to flip the state at zero temperature so the system is stable.

The iterative Hebb rule is laid out in detail in Algorithm I. At each iteration the weights are updated by

$$w_{ji} \leftarrow w_{ji} + \frac{A_{ji} \xi_j^p \xi_i^p}{N}, \quad (10)$$

until the required condition is met. For this study  $\delta$  was always set at 1, but other values can be used to change the stability of the patterns. If a stable solution of this set of inequalities exists it should always converge in a finite amount of time [33]. However, a solution does not always exist for sparse directed networks, so the algorithm needs to be terminated after a chosen maximum number of iterations. Here we use 400 iterations. The patterns can still be quite successfully stored and recovered if full convergence has not been achieved, as

the number of weights continually updated is small after only a few iterations.

Pattern recovery is measured with an order parameter, which we call magnetization, and is defined for each pattern  $p$  as the scalar product of the state of the system and the pattern:

$$m_p = \frac{1}{N} \sum_{i=1}^N s_i \xi_i^p. \quad (11)$$

This is equivalent to the cosine of the angle between the state and the pattern. In this work we study patterns which are random, independent, and identically distributed. Correlation between patterns and between patterns and the network topology may affect the performance of the network in a wide variety of ways depending on the topology, sparsity, and nature of the correlation [32,37], so this may be a potential avenue for future work.

#### IV. NETWORK GENERATION

To generate networks with a specified trophic coherence and fixed numbers of nodes and edges, we use a variant of the generalized preferential preying model from Refs. [18,38], although the original work used a different definition of trophic level. We generate networks such that each node has in-degree at least 1. One reason for this is that if the network contains basal nodes (nodes with in-degree 0), one must choose whether their states  $s$  should remain constant, take random values at each time step, or act as external inputs to the system. Moreover, it is known that basal (or source) nodes can drive the dynamics on directed networks in certain contexts [8,39], but we investigate here the importance of trophic level for dynamics on networks without basal nodes.

The detailed steps of the generative process are laid out in Appendix B. In short, we randomly generate an initial configuration of  $N$  nodes where each node has in-degree 1 and then calculate the initial trophic level  $\tilde{h}$  of this configuration. Then edges are added until the specific number is reached where the probability of connecting node  $i$  to  $j$  is

$$P_{ij} = \exp \left( -\frac{(\tilde{h}_j - \tilde{h}_i - 1)^2}{2T_{\text{gen}}} \right). \quad (12)$$

Afterward, the updated trophic levels  $h$  are recalculated. With this method networks of any incoherence can be generated by varying the control parameter  $T_{\text{gen}}$ , as demonstrated in Fig. 2.

The networks generated via this method can act as an approximation to the hierarchical structures seen in real-world systems. In Ref. [3] it was shown that many real-world networks conform approximately to an analytical prediction for their scaled spectral radius  $\rho_s$  as a function of the incoherence parameter  $F$ . This relationship is

$$\rho_s = \exp \left( \frac{1 - \frac{1}{F}}{2} \right) \quad (13)$$

and can be derived from the coherence ensemble of random graphs [7]. Here  $\rho_s$  is defined such that it is scaled between 0 and 1 to compare networks of different sizes

$$\rho_s = \frac{\rho}{\|A\|_2}, \quad (14)$$

where  $\rho$  is the standard spectral radius of the adjacency matrix and  $\|A\|_2$  is the 2-norm of  $A$ , that is,  $\|A\|_2^2$  is the largest eigen-

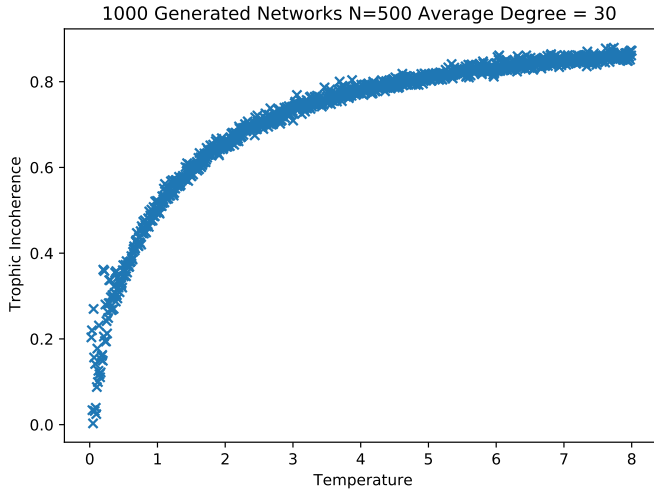


FIG. 2. Example of the distribution of trophic incoherence with the temperaturelike parameter  $T_{gen}$  in this generative model.

value of  $AA^T$ . As we show in Fig. 3, the generated networks we use in this work also have  $\rho_s$  close to the value given by Eq. (13). This justifies the assumption that the numerically generated networks reflect some of the characteristics exhibited by real-world networks.

V. RESULTS

First, as we impose the constraint that only a subset of neurons is presented the pattern to make the setup more like real-world systems, we must decide which set of neurons is to be shown the pattern and assess the effects of this choice. It was chosen that for this model 20% of the neurons would be set into a pattern state and then it would be measured how well the system recovered the remainder of the pattern from this setup. The location of pattern presentation is analogous to the initial conditions of a dynamical system where the question would be which initial condition sends the system into the desired state given the constraint of only controlling

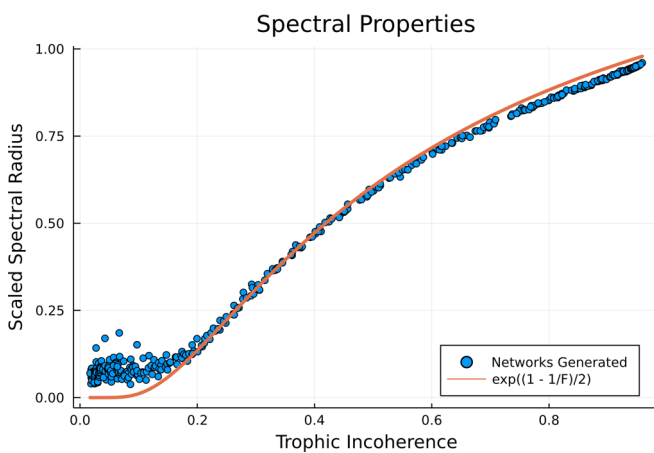


FIG. 3. Plot of scaled spectral radius of generated networks vs trophic incoherence following the same analytic prediction as real networks as shown in Ref. [3]. The number of nodes is always  $N = 500$  and the mean degree  $\langle k \rangle = 20$ .

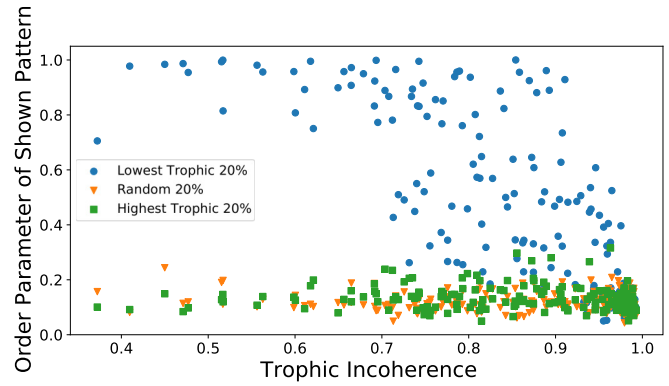


FIG. 4. Plot of performance of 200 networks with  $N = 500$  and  $\langle k \rangle = 100$  recovering ten patterns vs trophic incoherence. Patterns are shown for the different 20% sets of nodes.

a small number of elements. To assess the effect of hierarchy on pattern recovery, patterns were shown to the 20% of nodes with the lowest trophic level (at the bottom of the hierarchy), highest trophic level (at the top of the hierarchy), and a random 20% of nodes. The results are shown in Fig. 4 for networks spanning a range of trophic coherence. We plot the results for each individual network, rather than just the averages with error bars, in order to highlight the breadth and distribution of network behavior, which becomes more apparent as we study sparser networks in Sec. V A. These results demonstrate the difference in dynamics depending on the part of the network shown the pattern. When the pattern is shown to 20% of the nodes randomly this is not enough to move the system into a new state, so the shown pattern is not recovered well across the whole range of trophic coherence. It is only possible to extend down to networks of intermediate coherence at this edge density with the generative method used. When the perturbation is made to the state of the top 20% of nodes by the trophic level, it has little effect on the state of the system. This is because the perturbation cannot filter back down the system, so the top nodes do not drive the dynamics. For sparse enough networks and high coherence, it is unlikely there will be any paths from the highest trophic levels to other nodes further down. If the network is denser, such paths may exist, but they will still be few compared with the number of paths from lower levels to higher. Hence, information flow will always be predominantly from lower to higher trophic levels in coherent networks.

The dynamics is more complex when patterns are presented to the lowest-level nodes, since we observe different behaviors when trophic incoherence is varied. For the most incoherent networks, which are most similar to random graphs, the performance is on average poorer as the system is more stable due to the amount of feedback in the system. By stability we mean here the system’s resistance to changing state when a new pattern is presented. At intermediate coherence, the network has an overall direction, so the perturbation at low-level nodes is transmitted through the hierarchical network structure and pattern recovery is quite good even though only 20% of nodes are stimulated. This is behavior that would not be seen in a Hopfield model on a complete graph, nor on a random graph, since more than half the nodes would need to

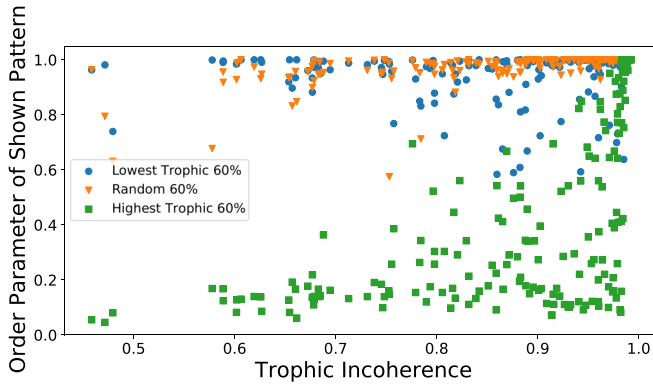


FIG. 5. Plot of performance of 200 networks with  $N = 500$  and  $\langle k \rangle = 100$  recovering ten patterns vs trophic incoherence. Patterns are shown for the different 60% sets of nodes.

be changed to a new pattern in order to change the state of the system. These results demonstrate the variety of dynamics that can be induced by the more complex hierarchical networks as compared to a complete or random graph [10,40,41].

When the constraint of a small number of input neurons is removed, the effect of hierarchy on the dynamics is less obvious. Figure 5 illustrates the case when the patterns are shown to 60% of neurons. When this many neurons receive an input the distinction between the outcome of showing a pattern to a random 60% and the lowest level 60% is blurred, with both being able to recover the pattern across a range of trophic coherence. This highlights the impact of removing the constraint of a small number of inputs. When the inputs are large the effect of the trophic level is hidden as randomly chosen inputs can control the system. However, for the highest-level nodes this is still not the case. Even at 60%, the higher-level nodes fail to influence the coherent networks, as the lowest-level nodes still have more control over the system and prevent the pattern from being modified. When the network is hierarchical, perturbations can be both amplified or damped by the structure, something not seen in either a complete or a random graph Hopfield network. This is again different behavior than what would be observed on a dense network with no internal structure, as 60% of neurons being flipped would be enough to change the state to that of the new pattern in all cases. This highlights the connection between the trophic level of a node and its ability to control the network: The high-level nodes have much less ability to influence the system than those at a low level. This difference remains at all levels of trophic coherence, but is most pronounced for more coherent structures. In all examples the trophic incoherence does not actually reach 1, where all the nodes would have the same level. This is because this only happens in balanced networks, such as a directed cycle, and the limit of our model is Erdős-Rényi random graphs, which have incoherence around 0.95. It is interesting to note the graphs which are random still have a slight hierarchical structure which can be revealed by the trophic levels.

#### A. Sparser networks

When the networks are made sparser, that is, the average degree  $\langle k \rangle$  is reduced from 100 to 20, the results are broadly

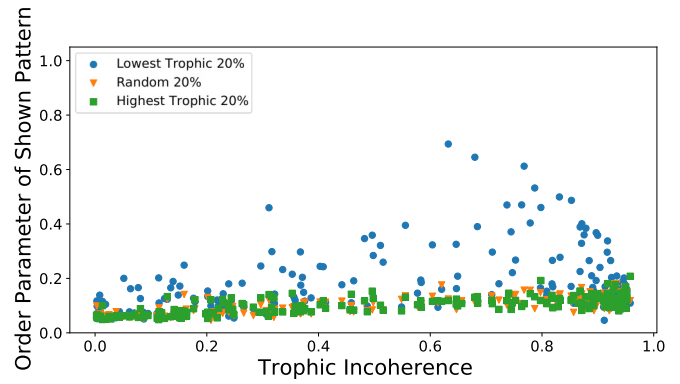


FIG. 6. Performance of 200 networks with trophic incoherence showing patterns to the 20% lowest (blue), highest (green), and random (orange) nodes by trophic level, for  $N = 500$  and  $\langle k \rangle = 20$ .

the same as on denser networks, but there is more variation in the performance of different networks, even for similar trophic coherence (see Fig. 6). For networks of this sparsity the whole range of coherence can be investigated, as there are no difficulties associated with generating the more coherent networks. For inputs to both randomly selected and highest-level nodes, the recovery is very poor, just as it was before. When it is the lowest-level nodes which receive the input, behavior depends on the trophic incoherence of the network. For the networks with lowest incoherence, the performance is generally very poor. This is due to the fact that these networks have very little feedback and small strongly connected components, so the patterns are not well recovered. For the intermediate coherence networks, performance is inconsistent. Some networks perform very well, with their structure being suited to controlling the system with only the low-level nodes, while other networks perform very badly. Finally, higher incoherence networks are again more likely to get stuck in a pattern rather than to respond to the stimulus at the lowest-level nodes, due to the high amount of feedback in the system, and the maximum performance begins to decrease again. Therefore, for sparser networks we find that the best performance is found at intermediate coherence, although not all networks in this range are necessarily high performing.

The relationship between average degree and recovery of patterns is shown in Fig. 7, where all networks have 500 nodes and are generated using  $T_{\text{gen}} = 1$ . The task cannot be performed by the most sparse networks, as they all fail to store any patterns. At an average degree of around 20, we reach the regime where some recovery is possible. For higher density, recovery reaches an inconsistent regime, where performance varies greatly for networks of similar degree and trophic properties. This kind of regime is most interesting to study since the dynamics has a great deal of variability and successful pattern recovery is possible but not sure. Above an average degree of about 200, the structural features of the network are lost as the network is too dense and it simply gets stuck in one state for the whole dynamics and there is no ability to update when presented with a small number of inputs. Hence, Fig. 7 demonstrates that increasing the network density can make performance at a pattern recovery task worse, which is counter to the general expectation for Hopfield networks where higher connectivity improves performance [41,42].

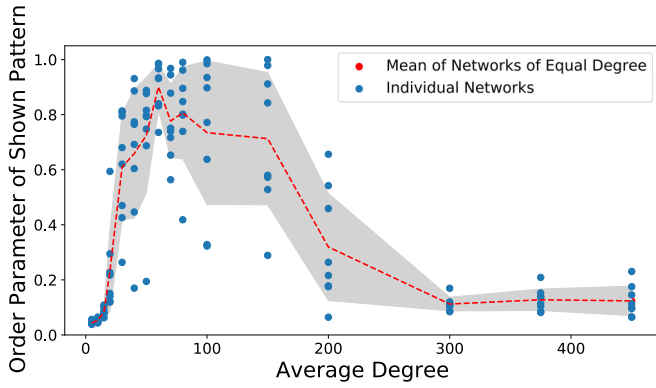


FIG. 7. Performance of networks of varying degree for fixed generation temperature  $T_{\text{gen}} = 1$ , showing patterns to the 20% lowest nodes by trophic level, for  $N = 500$ . The average trend is shown by the red dashed line and the standard deviation is shown in the shaded area.

**B. Comparison of targeting highest-degree nodes**

To validate our choice of nodes we compare our results to a targeted presentation of the pattern to the 20% of nodes of highest out-degree, which one might assume form the subset of nodes with greatest local influence on the system. This comparison is shown in Fig. 8, which compares the influence of the nodes of high degree to the selection of nodes by their local trophic properties. The set of nodes with highest degree do not influence the network to the extent that the lowest-level

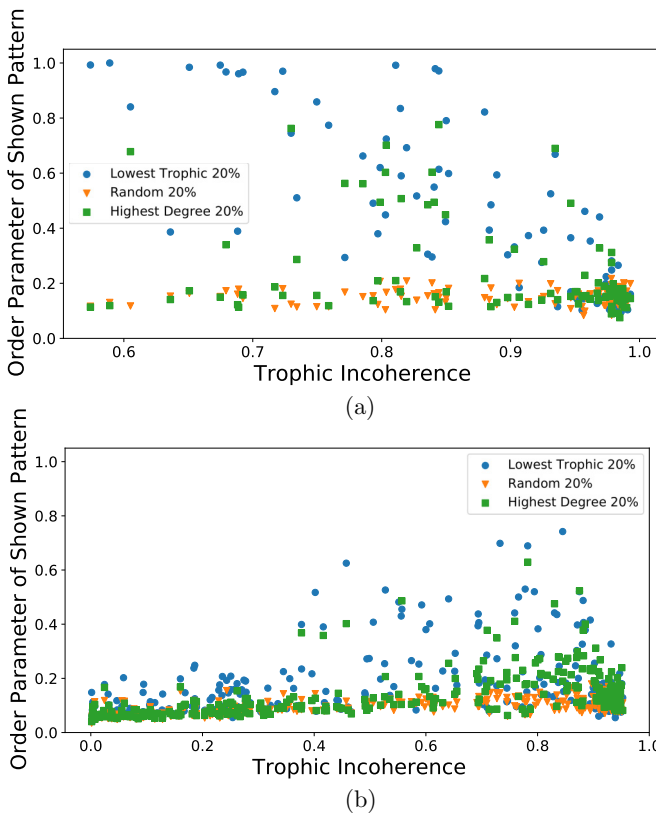


FIG. 8. Distribution of performance for networks showing ten patterns to the lowest trophic level, highest degree, and random 20% of nodes for  $N = 500$  and (a)  $\langle k \rangle = 100$  and (b)  $\langle k \rangle = 20$ .

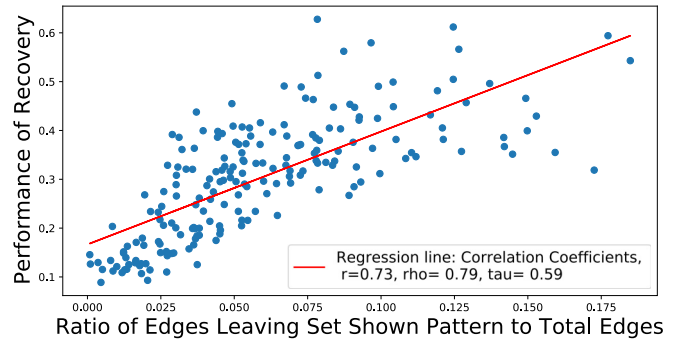


FIG. 9. Relationship between network performance and the ratio between the number of edges leaving the set shown the pattern and total edges in the network for  $N = 500$  and  $\langle k \rangle = 20$ . Networks are of intermediate incoherence.

nodes do. However, they do perform better than a random set of nodes, as expected, in both networks of average degrees 20 and 100. In the networks of average degree 100 [Fig. 8(a)], the lowest trophic level nodes are better than the highest degree nodes when the network is more coherent and hierarchical, as in this case the system is more strongly controlled by the low-level nodes. When the networks are less hierarchical, the influence of the high out-degree nodes becomes comparable to the influence of the low-level nodes. This highlights a crucial point: In a complex network, the “importance” of nodes can be determined both by their degree-based centrality and by their relative position in the hierarchy, depending on how trophically coherent the overall system is. In a very hierarchical (i.e., coherent) network, even if a node has a high out-degree, the state of the system can still be more controlled by lower out-degree nodes below it in the hierarchy. Our results, due to the generative model, focus on networks where the degree distributions are not extremely heterogeneous. The fact that in very hierarchical networks low-level nodes control the state of the nodes above them would still hold in a very heterogeneous network. However, degree may be a more important factor if the out-degree of a few nodes were so large that they directly affected much of the network. These network properties can interact in a variety of ways and may be the subject of future work.

**C. Structural properties of networks affecting performance**

We hypothesized that some network properties outwith trophic coherence could explain the range and inconsistency of behavior for sparse networks. One possible measure was the number of edges leaving the node set shown the pattern compared to the total number of edges. When very few edges connect the nodes shown the pattern to the rest of the network, it is unlikely for the pattern to be successfully recovered, as when the pattern is updated it cannot be properly transmitted outside the set shown the pattern. The results of this are displayed in Fig. 9. This shows that there is a strong correlation (correlation coefficients in the legend) between the edge ratio and performance, but it does not exactly determine the behavior of the system. However, it is very clear that the worse performing networks have very small values of this parameter and it can be used to identify the failing networks, if not precisely to select the very best networks.

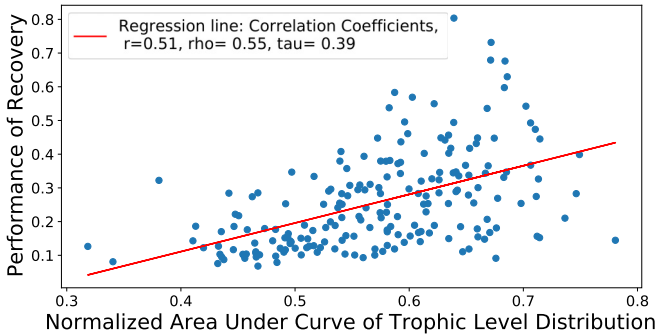


FIG. 10. Plot of distribution of performance of 200 intermediate incoherence networks vs the integral from 0 to 1 over a parameter  $\alpha$  of the curve of the number of nodes of trophic level less than  $\alpha h_{\max}$  for  $N = 500$  and  $\langle k \rangle = 20$ . Networks are of intermediate incoherence.

Another factor which we thought may influence the performance was the distribution of trophic levels among the nodes. In networks generated with the model used here (see Sec. IV), edges tend only to span a small difference in trophic level. We would therefore like the level distribution to be peaked towards lower levels so that more nodes have a lower level and are more likely to be densely interconnected with the set of nodes shown the pattern. This is shown in Fig. 10, where we sum the cumulative distribution of the number of nodes of trophic level less than  $\alpha h_{\max}$  for  $\alpha$  in the range 0 – 1. This function is maximized when the level distribution peaks towards lower-level nodes and so provides a good measure of where the peak in trophic level lies, while being normalized so different networks can be compared. It shows a similar profile to the result of Fig. 9, where the correlation is again strong but does not precisely predict the performance of the network. We therefore surmise that the performance of a network at this task depends on several topological features, including but not limited to trophic coherence, mean degree, mean degree of the lowest-level nodes, and trophic level distribution.

#### D. Time series of pattern recovery in sparse network components

In this section we review the time series of the dynamics of pattern recovery in a network with average degree 20 and highlight some of the reasons for the inconsistency in per-

formance between similar networks. In all of the following example time series the pattern is presented to the 20% of nodes with the lowest trophic level.

We find that the network structure can induce quite heterogeneous dynamics. This is something that is not noticeable when the recovery is working well. Let us consider first the case of a dense network, with mean degree 100, as shown in Fig. 11. In this time series each color represents the pattern which has been most recently presented to the network, while the y axis represents the order parameter corresponding to that pattern. For a well-performing network, the order parameter quickly returns close to 1 whenever a new pattern is presented. This is the case for the network shown in Fig. 11. Due to the recovery being this good and the edge density being high, heterogeneous dynamics is not observed. Patterns are recovered to the same extent in all parts of the network hierarchy and additionally the whole network is strongly connected, so there is no difference in dynamics inside or outside that component.

This is very different from the dynamics exhibited by the sparse network used in Fig. 12, which stores four patterns. This network is a specific example of a network which performs reasonably well, but it should be borne in mind that many sparse networks fail very badly. The dynamics is analyzed by considering four different network components (subgraphs): the whole system, the largest strongly connected component, the bottom 20% of the nodes by trophic level, and the top 20% of nodes by level. Note that these components are simply where the data were collected and the presentation location was unchanged. In the full system [Fig. 12(a)], recovery is good for some patterns but fails badly for others. The behavior of each pattern is roughly consistent, and if a pattern fails or succeeds at one presentation it will repeat the same behavior at subsequent presentations. The order parameter dips when a new pattern is presented and then moves to its new stable value. Additionally, there are fluctuations around the stationary states and updates to the system do not stop (i.e., some neurons continue to change state in subsequent time steps). This is different from the dynamics inside the largest strongly connected component [Fig. 12(b)], where for the fully recovered stable patterns updates stop and there are no fluctuations. This highlights the stabilizing effects of feedback associated with being strongly connected. Among the low-level nodes [Fig. 12(c)], for those patterns which are correctly recalled, the order parameter goes to 1 when the new

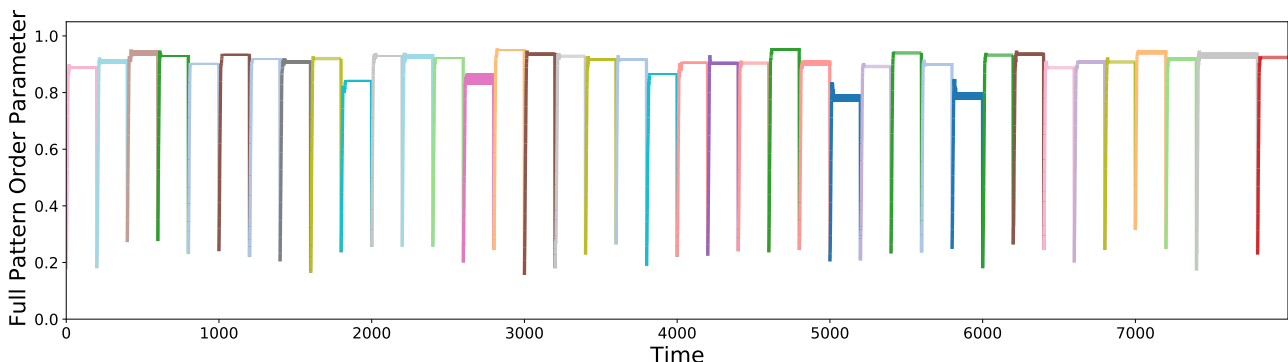


FIG. 11. Time series of the pattern order parameter for an average degree 100 network with 500 nodes storing 20 patterns. Each color represents the pattern most recently shown to the network. At this edge density recovery is very consistent.



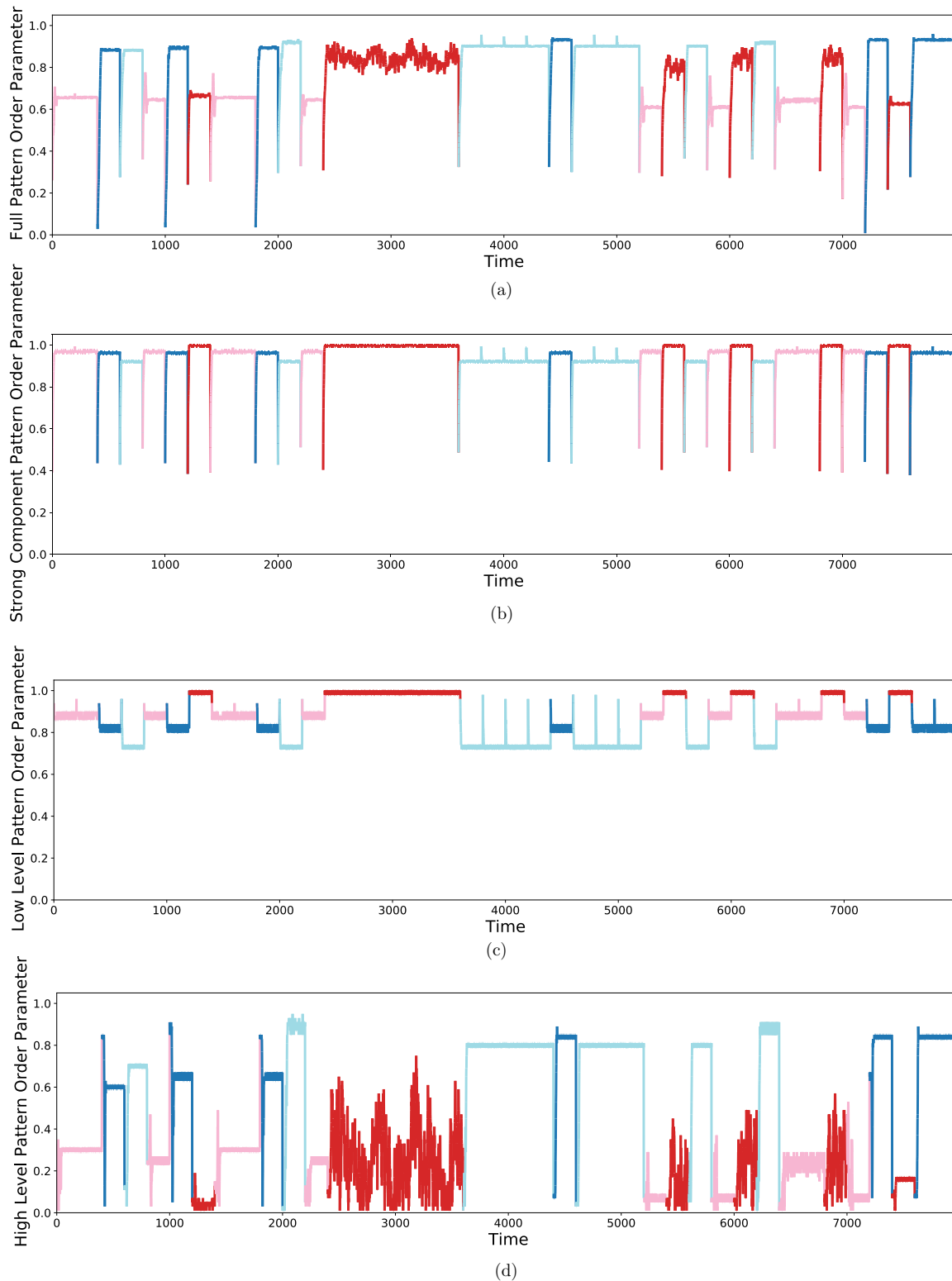


FIG. 12. Time series for different network components for an average degree 20 network with 500 nodes storing 4 patterns: (a) full pattern order parameter, (b) strongly connected component, (c) bottom 20% of nodes by trophic level, and (d) top 20% of nodes by trophic level. Each color represents the pattern most recently shown to the network.

pattern is presented. However, if a pattern is not recovered by the low-level nodes then this precludes the possibility of that pattern being successfully transmitted through the network. This means that if a pattern is not recovered by the low-level

nodes [Fig. 12(c)], then it will also fail to be recovered by the high-level nodes [Fig. 12(d)]. Recovery by the high-level nodes is the least consistent and fluctuates the most, since these nodes are furthest from where the patterns are presented.

In addition, the order parameter initially drops to zero whenever a new pattern is presented as it is not shown to any of the nodes contained in this set. These results might be different if the network included basal nodes (those with no in neighbors) and would depend on what update rule we chose for these, e.g., maintain their state indefinitely, update randomly, etc.

### E. Search for improvements to network structure

The results relating the distribution of trophic levels to neural-network performance open the possibility of biasing the network generation process so that it preferentially leads to networks with topology better suited to the task. A simple way to accomplish this is to generate the networks in the same way as previously, but modify the probability of adding edges so that it is biased towards adding edges to lower-level nodes. This can be accomplished by modifying the probability of placing an edge so that

$$P_{ij} = \exp\left(-\frac{(\tilde{h}_j - \tilde{h}_i - 1)^2}{2T_{\text{gen}}} + \gamma\tilde{h}_i\right), \quad (15)$$

where the  $\gamma\tilde{h}_i$  modification in the exponential acts to bias the distribution towards high or low levels, depending on the sign of  $\gamma$ . In what follows we choose  $\gamma = -0.5$  in order to add more edges to nodes with lower trophic level. One downside of this method is that it is harder to control precisely the trophic incoherence of a network and to span the full range of incoherence.

The broad effects of biasing the network generation and performance are demonstrated in Fig. 13. This shows that when edges are more likely to connect to low-level nodes [Fig. 13(a)], the very worst performing networks are essentially eliminated and all the sparse networks recall at least a fraction of the pattern. This biasing has no effect on performance when presenting the pattern to the random or higher-level nodes, as they still fail to force the system to change state when the perturbation is applied to these nodes.

To demonstrate the importance of where feedback is placed in the hierarchy we change the sign of the biasing factor and make it more likely that edges are added to the higher-level nodes [Fig. 13(b)]. This creates networks which are not suited to the recovery task and perform badly in all cases. This is due to the fact that the edges connecting to high-level nodes do not allow both for the pattern to be stable and for the information to be transmitted across the system. One issue with biasing the network generative process is that it becomes more difficult to control precisely the trophic coherence of the network, which is why the range of trophic incoherence is restricted in Fig. 13.

The time series of pattern recovery for sparse biased networks clearly demonstrate how this biasing procedure modifies the dynamics of the system. Pattern recovery across the whole system [Fig. 14(a)] is very consistent compared to the unbiased networks (Fig. 6), which fully recover some patterns and fail to recover others. This time series is a representative example of the behavior seen in biased networks and comes from a single network. The consistent level which they reach however is below 1, so the patterns are not fully recovered and the recovery is not as high as the maximum seen in some specific unbiased networks (Fig. 12). Whether

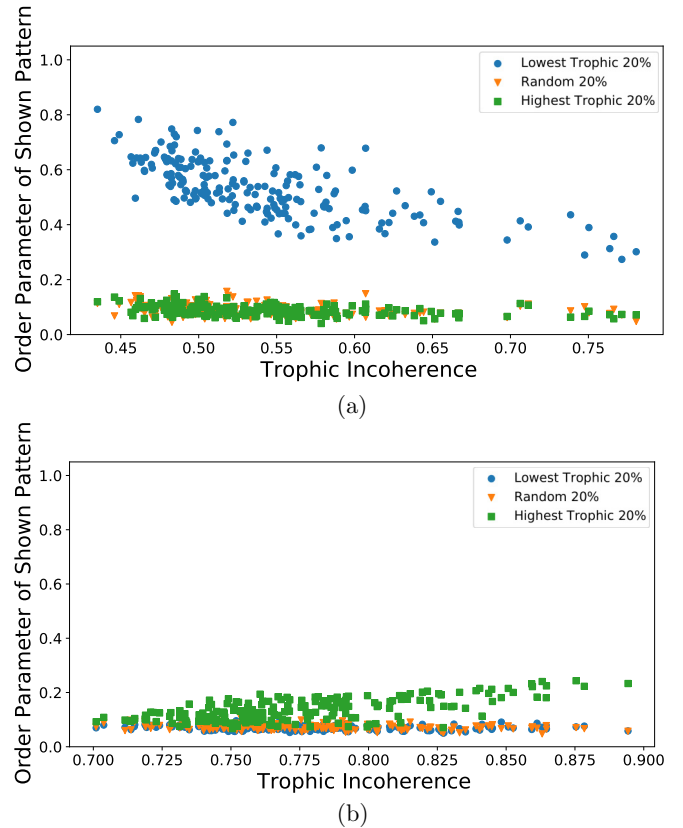


FIG. 13. Distribution of performance for biased networks biased towards (a) low-level nodes with  $\gamma = -0.5$  and (b) high-level nodes with  $\gamma = 0.5$ , for  $N = 500$  and  $\langle k \rangle = 20$ .

this is better may depend on the context: Remembering part of every pattern so it can be identified may be preferable to recalling some patterns perfectly but not recovering others at all. Additionally, for biased networks there are large fluctuations and updates continue when the system has reached the new state. This can be explained by looking at the dynamics inside the largest strongly connected component only [Fig. 14(b)]. In this component recovery is very consistent and all patterns are fully recovered, so the network does much better when this component is larger. It also explains why, in the time series for the dynamics of the full network, fluctuations around a stable point are observed, since updates to neuron states stop in the strongly connected component but continue outside it. The fact that the recovery is very good inside the largest strongly connected component opens up the possibility of selectively generating networks which both are biased towards lower-level nodes and have large strongly connected components.

This is demonstrated in Fig. 15, which shows the performance of biased networks where the largest strongly connected component comprises more than 60% of the nodes. These networks are simply generated by repeating the generative process and discarding networks which do not meet this requirement. The higher this threshold, the more inefficient the process but the more likely we are to keep only highly performing networks. At a threshold of 60% all very poorly performing networks are eliminated and the recovery performance is consistently around 0.6. These results demonstrate

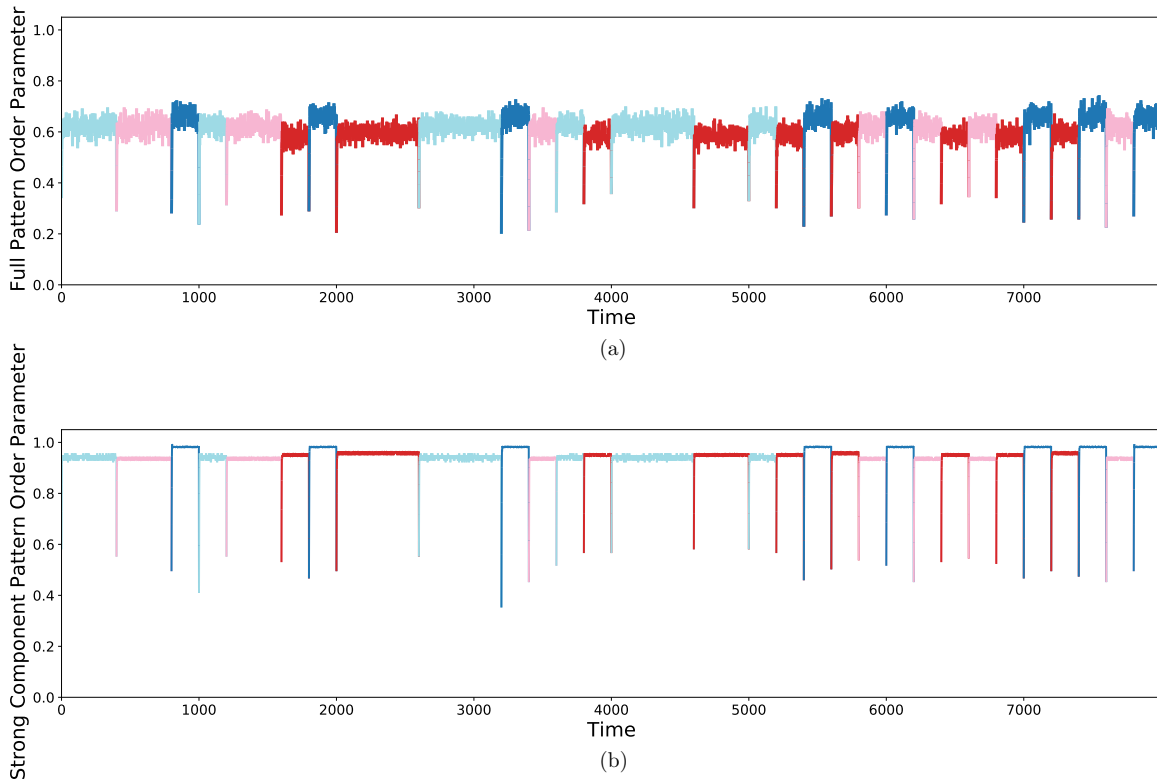


FIG. 14. Time series of the pattern order parameter calculated inside the different components for an average degree 20 network with 500 nodes storing 4 patterns, generated with a bias towards adding edges to lower-level nodes: (a) order parameter of the whole network and (b) largest strongly connected component only. Each color represents the pattern most recently shown to the network.

that despite the variability in the dynamics of directed sparse Hopfield networks, it is possible to generate structures which perform well consistently by tuning a few parameters:  $T_{gen}$  to set the trophic coherence,  $\gamma$  to place edges preferentially at lower level nodes, and the threshold for the minimum size of the strongly connected component.

There are many possible ways to modify network structure to maintain performance and we just give a small sample here. Biasing is limited by the fact that it reduces the control of the trophic structure and that strongly biasing may decrease the size of the strongly connected component which is needed for recovery. The best way to improve network structure depends

on the constraints, on whether edges can be added or removed, and on how success is defined. Biasing makes the recovery more consistent, but the performance of unbiased networks may peak higher for certain networks and patterns, which may be preferred in some situations.

VI. CONCLUSION

We have shown that neural networks based on sparse trophically coherent graphs have a much wider range of possible behavior than ones based on either fully random or complete graphs [10,40,41], where all nodes necessarily have very similar dynamical roles. This symmetry is broken in a coherent network, as different nodes can have very different abilities to affect the dynamics of the system. The interplay between trophic structure and dynamics has already been observed across a range of systems in the literature [18,22,26]. It has also been shown that the coherence of a network is linked to the non-normality of the adjacency matrix [3,8]. Non-normality in networks has in turn been linked to sensitivity to perturbations and to the stability of the system across a wide range of dynamics [43–47], which is consistent with our results that more coherent networks are more sensitive to targeted perturbations and less stable.

The behavior observed in the system studied here relies on two key facts: that the networks are sparse and that the sets of input nodes are small. If the networks are too dense then hierarchical structure is destroyed and the asymmetry between nodes does not exist (there is a limit to how coherent a dense network can be). Moreover, it is due to the

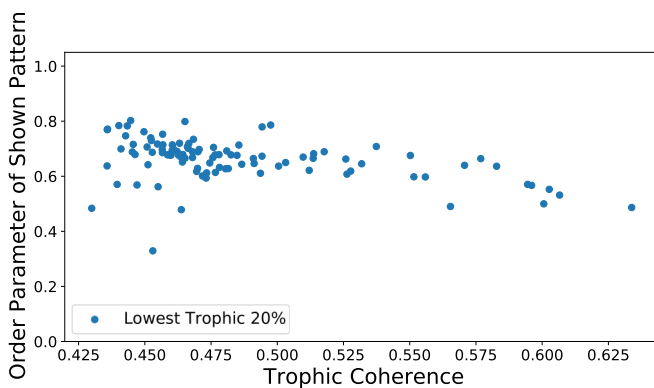


FIG. 15. Performance of low-level-biased networks with nodes of average degrees 20 and 500 where the largest strongly connected component contains at least 60% of the neurons.

network's trophic coherence that a small subset of nodes is able to drive the dynamics of the whole system. Many real-world systems display both of these properties. Additionally, they are often neither highly coherent nor incoherent, but have trophic coherence in the intermediate range which allows for a balance between stability and sensitivity to stimuli [3,7,8]. Therefore, we believe the principles studied here for the case of Hopfield-like neural networks may be broadly applicable to a range of real-world systems. The limitations of these methods are that since trophic incoherence is an average global network property it lacks the precise detail to characterize fully the behavior of the system in all cases. It is challenging to control precisely both the trophic incoherence and another aspect of network structure, since one may restrict the other, as with the biasing method. In the future this work could be extended by looking at a time series of patterns which are correlated with each other [37], patterns correlated with the structure, or networks with heterogeneous degree distributions and varying in- and out-degree correlations.

In conclusion, we have demonstrated, through numerical analysis, that trophic structure strongly shapes pattern recovery in directed Hopfield-like networks. In particular, on a sparse directed network a small number of input neurons, which can be identified by their trophic levels even in the absence of basal nodes, are able to drive the system in such a way that it recovers patterns. This would not be possible on either a complete or fully random network, which requires at least about 50% of the nodes to receive the input in order to change state. In order for such networks to recover patterns successfully, they must have the correct balance between feedback and directionality, a feature which is determined by the trophic coherence. However, we observed that setting the appropriate trophic coherence was not enough to guarantee good performance. We found that by biasing the network generation process so as to add edges preferentially to lower-level nodes and then discarding networks with strongly connected components below a minimum size, we could reliably produce architectures that perform the task well.

#### ACKNOWLEDGMENTS

The authors would like to thank the Centre for Doctoral Training in Topological Design and Engineering and Physical Sciences Research Council (EPSRC) for funding this research. S.J. also acknowledges support from the Alan Turing Institute under EPSRC Grant No. EP/N510129/1.

#### APPENDIX A: SOFTWARE TOOLS USED

Various software packages were used to manipulate the networks and perform the simulations. The PYTHON package GRAPH TOOL [48] was used for some of the network manipulation. NETWORKX [5] was used for network drawing and some network manipulation and analysis. The JULIA package LIGHTGRAPHS.JL was used for the spectral radius results [49]. All the updating and training of the Hopfield-like networks was done with the aid of the CYTHON package [50] to convert

PYTHON code to C as pure PYTHON was found to be too slow to allow efficient study.

#### APPENDIX B: NETWORK GENERATION

The detailed steps to the network generative process are as follows:

(1) Create the  $N$  nodes of the network and assign to each node one incoming edge in each case from a randomly chosen other node. After this, each node has in-degree 1.

(2) Compute the initial trophic levels  $\tilde{h}$  using Eq. (2). This is best solved iteratively, since this method is fast and works even if there are small disconnected components.

(3) Add edges up to the desired edge number with probability dependent on the trophic level difference between the nodes minus 1. The edge probability used is Gaussian and defined as

$$P_{ij} = \exp\left(-\frac{(\tilde{h}_j - \tilde{h}_i - 1)^2}{2T_{\text{gen}}}\right), \quad (\text{B1})$$

where  $T_{\text{gen}}$  is a temperaturelike parameter used to control how coherent the network is: Small  $T_{\text{gen}}$  generates networks which are highly coherent.

(4) Recompute the trophic levels  $h$  including the newly added edges. Then compute the incoherence parameter  $F$  of the generated network.

This method works best for reasonably sparse networks, since when the edge density is too large it becomes difficult to find configurations of high trophic coherence, if they exist at all. On the other hand, if the edge density is very low the resulting network may not be even weakly connected. However, for a large range of densities it will encounter no issues. Due to the stochastic nature of the method it is not possible to predict precisely the incoherence of a generated graph. For example, 1000 networks generated with 500 nodes and  $30 \times 500$  edges, at temperature  $T_{\text{gen}} = 1.3$ , cluster around  $F \approx 0.59$ , with most networks in the interval  $F \in (0.56, 0.65)$ . However this level of precision is sufficient for analyzing general regions of behavior with no issues.

The third step can be quite computationally inefficient for large networks with many possible edges as the probabilities for adding an edge at most locations are very close to zero. This can be improved by more efficiently sampling the probability distribution using the method outlined below.

The goal of this sampling method is to set up the sampling so that each time a random number is drawn it results in an edge. This avoids repeatedly drawing numbers for the majority of edges which are unlikely to be added. The steps are as follows:

(1) Label all the possible edges and probabilities with an integer  $l$  and  $P_l$ , respectively.

(2) Compute the sum of all these probabilities

$$S = \sum_l P_l.$$

(3) Draw a random number  $r$  between 0 and  $S$ .

(4) Sum the probabilities one at a time until the random number  $r$  is reached.

(5) Add an edge at the space  $l$  corresponding to the probability  $P_l$  which made the sum greater than  $r$ .

(6) Set  $P_l = 0$  and repeat steps 2–6 until the required edge number is reached.

This method is much more efficient: The sums can be computed quickly as it avoids the many repeated random number draws for every single missed edge that would otherwise be

necessary. It is possible to also create variants of this method by modifying the initial structure to which subsequent edges are added or to recast the model so as to start from a dense network and prune edges with a similarly defined probability to generate networks of the desired trophic incoherence.

- 
- [1] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner, The structure of the nervous system of the nematode *Caenorhabditis elegans*, *Philos. Trans. R. Soc. London Ser. B* **314**, 1 (1986).
- [2] D. J. Watts and S. H. Strogatz, Collective dynamics of ‘small-world’ networks, *Nature (London)* **393**, 440 (1998).
- [3] R. S. MacKay, S. Johnson, and B. Sansom, How directed is a directed network?, *R. Soc. Open Sci.* **7**, 201138 (2020).
- [4] S. Johnson, [www.samuel-johnson.org](http://www.samuel-johnson.org) data repository, <https://www.samuel-johnson.org/data> (accessed October 2020).
- [5] A. A. Hagberg, D. A. Schult, and P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, in *Proceedings of the 7th Python in Science Conference (SciPy 2008), Pasadena, 2008*, edited by G. Varoquaux, T. Vaught, and J. Millman (SciPy, 2008), pp. 11–15, [https://conference.scipy.org/proceedings/SciPy2008/paper\\_2/](https://conference.scipy.org/proceedings/SciPy2008/paper_2/).
- [6] P. Erdős and A. Rényi, On random graphs I, *Publ. Math.* **6**, 290 (1959).
- [7] S. Johnson and N. S. Jones, Looplessness in networks is linked to trophic coherence, *Proc. Natl. Acad. Sci. USA* **114**, 5618 (2017).
- [8] S. Johnson, Digraphs are different: Why directionality matters in complex systems, *J. Phys.* **1**, 015003 (2020).
- [9] P. Kale, A. Zalesky, and L. L. Gollo, Estimating the impact of structural directionality: How reliable are undirected connectomes?, *Netw. Neurosci.* **2**, 259 (2018).
- [10] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA* **79**, 2554 (1982).
- [11] P. Zheng, J. Zhang, and W. Tang, Analysis and design of asymmetric Hopfield networks with discrete-time dynamics, *Biol. Cybern.* **103**, 79 (2010).
- [12] L. Cantini and M. Caselle, Hope4Genes: A Hopfield-like class prediction algorithm for transcriptomic data, *Sci. Rep.* **9**, 337 (2019).
- [13] A. Szedlak, G. Paternostro, and C. Piermarocchi, Control of asymmetric Hopfield networks and application to cancer attractors, *PLoS ONE* **9**, e105842 (2014).
- [14] G. Baggio, V. Rutten, G. Hennequin, and S. Zampieri, Efficient communication over complex dynamical networks: The role of matrix non-normality, *Sci. Adv.* **6**, eaba2282 (2020).
- [15] Y. Y. Liu and A. L. Barabási, Control principles of complex systems, *Rev. Mod. Phys.* **88**, 035006 (2016).
- [16] C. W. Lynn and D. S. Bassett, The physics of brain network structure, function and control, *Nat. Rev. Phys.* **1**, 318 (2019).
- [17] M. Leonetti, V. Folli, E. Milanetti, G. Ruocco, and G. Gosti, Network dilution and asymmetry in an efficient brain, *Philos. Mag.* **100**, 2544 (2020).
- [18] J. Klaise and S. Johnson, From neurons to epidemics: How trophic coherence affects spreading processes, *Chaos* **26**, 065310 (2016).
- [19] M. Chau and H. Chen, Incorporating web analysis into neural networks: An example in hopfield net searching, *IEEE Trans. Syst. Man Cy. C* **37**, 352 (2007).
- [20] A. Szedlak, S. Sims, N. Smith, G. Paternostro, and C. Piermarocchi, Cell cycle time series gene expression data encoded as cyclic attractors in Hopfield systems, *PLoS Comput. Biol.* **13**, e1005849 (2017).
- [21] R. C. Rockne, P. Ao, C. Espinosa-Soto, F. Alves Barbosa Da Silva, A. J. Conforte, L. Alves, F. C. Coelho, and N. Carels, Modeling basins of attraction for breast cancer using Hopfield networks, *Front. Genet.* **1**, 314 (2020).
- [22] S. Johnson, V. Domínguez-García, L. Donetti, and M. A. Muñoz, Trophic coherence determines food-web stability, *Proc. Natl. Acad. Sci. USA* **111**, 17923 (2014).
- [23] S. Levine, Several measures of trophic structure applicable to complex food webs, *J. Theor. Biol.* **83**, 195 (1980).
- [24] A. Pagani, G. Mosquera, A. Alturki, S. Johnson, S. Jarvis, A. Wilson, W. Guo, and L. Varga, Resilience or robustness: Identifying topological vulnerabilities in rail networks, *R. Soc. Open Sci.* **6**, 181301 (2019).
- [25] A. Pagani, F. Meng, G. Fu, M. Musolesi, and W. Guo, Quantifying resilience via multiscale feedback loops in water distribution networks, *J. Water Res. Plan. Man.* **146**, 04020039 (2020).
- [26] C. Pilgrim, W. Guo, and S. Johnson, Organisational social influence on directed hierarchical graphs, from tyranny to anarchy, *Sci. Rep.* **10**, 4388 (2020).
- [27] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Spin-glass models of neural networks, *Phys. Rev. A* **32**, 1007 (1985).
- [28] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**, 97 (1970).
- [29] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, *Cogn. Sci.* **9**, 147 (1985).
- [30] A. Crisanti, M. Falcioni, and A. Vulpiani, Transition from regular to complex behaviour in a discrete deterministic asymmetric neural network model, *J. Phys. A: Math. Gen.* **26**, 3441 (1993).
- [31] D. O. Hebb, *The Organization of Behavior; A Neuropsychological Theory*, 1st ed. (Psychology Press, New York, 2002), p. 378.
- [32] G. Tanaka, R. Nakane, T. Takeuchi, T. Yamane, D. Nakano, Y. Katayama, and A. Hirose, Spatially arranged sparse recurrent neural networks for energy efficient associative memory, *IEEE Trans. Neural Netw. Learn. Syst.* **31**, 24 (2020).
- [33] S. Diederich and M. Opper, Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules, *Phys. Rev. Lett.* **58**, 949 (1987).
- [34] E. Gardner, The space of interactions in neural network models, *J. Phys. A: Gen. Phys.* **21**, 257 (1988).
- [35] N. Davey and R. Adams, High capacity associative memories and connection constraints, *Connect. Sci.* **16**, 47 (2004).
- [36] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann, Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs, *Science* **275**, 213 (1997).

- [37] H. Gutfreund, Neural networks with hierarchically correlated patterns, *Phys. Rev. A* **37**, 570 (1988).
- [38] J. Klaise and S. Johnson, The origin of motif families in food webs, *Sci. Rep.* **7**, 16197 (2017).
- [39] E. A. Wright, S. Yoon, A. L. Ferreira, J. F. Mendes, and A. V. Goltsev, The central role of peripheral nodes in directed network dynamics, *Sci. Rep.* **9**, 13162 (2019).
- [40] I. Pérez Castillo and N. S. Skantzos, The Little-Hopfield model on a sparse random graph, *J. Phys. A: Math. Gen.* **37**, 9087 (2004).
- [41] R. J. McEliece, E. C. Posner, E. R. Rodemich, and S. S. Venkatesh, The capacity of the Hopfield associative memory, *IEEE Trans. Inf. Theory* **33**, 461 (1987).
- [42] H. Sompolinsky, Neural networks with nonlinear synapses and a static noise, *Phys. Rev. A* **34**, 2571 (1986).
- [43] M. Asllani, R. Lambiotte, and T. Carletti, Structure and dynamical behavior of non-normal networks, *Sci. Adv.* **4**, eaau9403 (2018).
- [44] M. Asllani and T. Carletti, Topological resilience in non-normal networked systems, *Phys. Rev. E* **97**, 042302 (2018).
- [45] R. Muolo, M. Asllani, D. Fanelli, P. K. Maini, and T. Carletti, Patterns of non-normality in networked systems, *J. Theor. Biol.* **480**, 81 (2019).
- [46] R. Muolo, T. Carletti, J. P. Gleeson, and M. Asllani, Synchronization dynamics in non-normal networks: The trade-off for optimality, *Entropy* **23**, 36 (2020).
- [47] M. Fruchart, R. Hanai, P. B. Littlewood, and V. Vitelli, Non-reciprocal phase transitions, *Nature (London)* **592**, 363 (2021).
- [48] T. P. Peixoto, The graph-tool python library, [http://figshare.com/articles/graph\\_tool/1164194](http://figshare.com/articles/graph_tool/1164194) (2014).
- [49] S. Bromberger *et al.*, JuliaGraphs/LightGraphs.jl: An optimized graphs package for the Julia programming language, <https://doi.org/10.5281/zenodo.889971> (2017).
- [50] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith, Cython: The best of both worlds, *Comput. Sci. Eng.* **13**, 31 (2011).