


# Influencer identification of dynamical networks based on an information entropy dimension reduction method

Dong-Li Duan(段东立), Si-Yuan Ji(纪思源), and Zi-Wei Yuan(袁紫薇)

**Citation:**Chin. Phys. B, 2024, 33 (**4**): 040502. DOI: 10.1088/1674-1056/ad102e
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

**What follows is a list of articles you may be interested in**

---

## Parametric instability in the pure-quartic nonlinear Schrödinger equation

Yun-Hong Zhang(张云红) and Chong Liu(刘冲)
Chin. Phys. B, 2024, 33 (**3**): 030506. DOI: 10.1088/1674-1056/ad11e7

## Semi-analytical steady-state response prediction for multi-dimensional quasi-Hamiltonian systems

Wen-Wei Ye(叶文伟), Lin-Cong Chen(陈林聪), Zi Yuan(原子), Jia-Min Qian(钱佳敏), and Jian-Qiao Sun(孙建桥)
Chin. Phys. B, 2023, 32 (**6**): 060506. DOI: 10.1088/1674-1056/acae7c

## Generalized projective synchronization of two coupled complex networks with different sizes

Li Ke-Zan (李科赞), He En (何恩), Zeng Zhao-Rong (曾朝蓉), Chi K. Tse (谢智刚)
Chin. Phys. B, 2013, 22 (**7**): 070504. DOI: 10.1088/1674-1056/22/7/070504

## Adaptive function projective synchronization of uncertain complex dynamical networks with disturbance

Wang Shu-Guo (王树国), Zheng Song (郑松)
Chin. Phys. B, 2013, 22 (**7**): 070503. DOI: 10.1088/1674-1056/22/7/070503

## Hash function construction using weighted complex dynamical networks

Song Yu-Rong (宋玉蓉), Jiang Guo-Ping (蒋国平)
Chin. Phys. B, 2013, 22 (**4**): 040506. DOI: 10.1088/1674-1056/22/4/040506

---

# Influencer identification of dynamical networks based on an information entropy dimension reduction method

Dong-Li Duan(段东立)[1,†], Si-Yuan Ji(纪思源)[1], and Zi-Wei Yuan(袁紫薇)[2]

[1] *School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an* 710311, *China*
[2] *School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an* 710072, *China*

(Received 11 September 2023; revised manuscript received 17 November 2023; accepted manuscript online 28 November 2023)

Identifying critical nodes or sets in large-scale networks is a fundamental scientific problem and one of the key research directions in the fields of data mining and network science when implementing network attacks, defense, repair and control. Traditional methods usually begin from the centrality, node location or the impact on the largest connected component after node destruction, mainly based on the network structure. However, these algorithms do not consider network state changes. We applied a model that combines a random connectivity matrix and minimal low-dimensional structures to represent network connectivity. By using mean field theory and information entropy to calculate node activity, we calculated the overlap between the random parts and fixed low-dimensional parts to quantify the influence of node impact on network state changes and ranked them by importance. We applied this algorithm and the proposed importance algorithm to the overall analysis and stratified analysis of the *C. elegans* neural network. We observed a change in the critical entropy of the network state and by utilizing the proposed method we can calculate the nodes that indirectly affect muscle cells through neural layers.

**Keywords:** dynamical networks, network influencer, low-dimensional dynamics, network disintegration

**PACS:** 05.10.–a, 89.75.Fb, 87.55.kd, 89.75.Hc **DOI:** 10.1088/1674-1056/ad102e

## 1. Introduction

From random networks to scale-free networks,[1–5] the connections between objects are inherently complex. In the real world, a range of complex systems, including power grids, road transportation networks and water supply pipelines, can be modeled as complex networks. Classic network models rely on highly structured connections[10–12] and these networks exhibit internal activity with spontaneous, often high-dimensional, dynamic behavior.[7–9] However, extensive research has confirmed that in large-scale networks, the activity of individual nodes resembles a mixture of fixed patterns and disorderly behavior.[13–16] The dynamics of such networks can be inferred from connectivity using geometric methods. For convenience, Sompolinsky *et al.*[6] defined the connectivity of these networks as the sum of their random components and the minimum low-dimensional structure required to achieve specific computations.[17]

In recent years, the research hotspot of complex networks has gradually shifted to the role played by individual nodes and links in the entire network. One group of nodes in the network is much smaller in size than the overall network but plays an important role in the entire network. If these nodes are activated or removed, it affects the structure and status of the entire network, making the originally stable network chaotic and causing some functions in the network system to be disordered.[43] It is crucial to enhance the most influential nodes with the least amount possible, as the entire network may even collapse due to the absence of these important nodes.[18] In the network model mentioned earlier, we have studied an important node recognition algorithm based on entropy[19,41] to identify nodes in the network that can control the network's behavior and status. Researchers generally believe that entropy can be used to measure the degree of disorder and we use this characteristic and combine it with the network model to calculate the overlap of a node's activity level with its random and fixed low-dimensional parts through dynamic mean field theory. We use the influence of the node on the network state change based on the overlap to quantify entropy value and finally rank the importance of each node according to its influence on the network entropy value. In this way, we find nodes that have a greater impact on the network state. We calculate the overlap of node activity with a random matrix and minimal low-dimensional structures using dynamic mean field theory, representing the current state of disorder or stability in the network and quantifying it using entropy values.[20–23] Finally, we rank the importance of each node according to its influence on the network entropy value, thus finding nodes that have a greater impact on the network state.

## 2. Model and method

### 2.1. The network dynamic model

Francesca Mastrogiuseppe and Srdjan Ostojic[17] employed a network model to record the relationship between

†Corresponding author. E-mail: mineduan@163.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd 

neuronal connectivity, low-dimensional dynamics and computational features. We follow this network model, in which each node is associated with a continuous variable, $x_i(t)$, which characterizes the network's dynamic behavior. The output of every node is a non-linear function, known as the sigmoidal function $\phi(x)$, applied consistently with previous findings,[6,24,25] where $\phi(x) = \tanh x$. In this case, the transformed $\phi(x_i(t))$ is recognized as the activity variable of node $i$ and $J_{ij}$ is the connectivity matrix. As a result, we deduce the ensuing dynamic model as

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi\left(x_j(t)\right), \tag{1}$$

no experiments considering external input perturbations are performed in this article. However, from a theoretical point of view, if the network is given an external input, Eq. (1) is altered to $\mathrm{d}x_i/\mathrm{d}t = -x_i(t) + \sum_{j=1}^{N} J_{ij}\phi\left(x_j(t)\right) + I_i$.

We consider a specific class of connectivity matrices that can be written as a sum of two terms

$$J_{ij} = g\chi_{ij} + P_{ij}. \tag{2}$$

Similar to the research done by Sompolinsky *et al.*,[6] $\chi_{ij}$ is a Gaussian all-to-all random matrix in which each element is extracted from a central normal distribution with a variance of $1/N$. The parameter $g$ represents the random connection strength scale in the network, which we refer to as random strength. The second term $P_{ij}$ is a low-rank matrix. In this network model, we assume that the low-rank part of the connection is fixed, while the random part varies between different implementations of the connection. This result depends on two simplified assumptions.[17] The first assumption is that the low-rank term is statistically uncorrelated. The second assumption is that the structured connectivity, as shown in Eq. (3), is weak in the limit of large $N$, i.e., its scale is $1/N$, and the scale of the random connection part $\chi_{ij}$ is $1/\sqrt{N}$. Our main focus is on the case where $P_{ij}$ is a rank-one matrix. In this scenario, $P_{ij}$ can be written as the outer product of two one-dimensional vectors $\boldsymbol{m}$ and $\boldsymbol{n}$

$$P_{ij} = \frac{m_i n_j}{N}. \tag{3}$$

According to the first assumption, the entries of vectors $\boldsymbol{m}$ and $\boldsymbol{n}$ are independent of the random connection matrix $\chi_{ij}$. In this case, the only non-zero eigenvalue of $P$ is given by $\boldsymbol{m}^{\mathrm{T}}\boldsymbol{n}/N$, with corresponding right and left eigenvectors being $\boldsymbol{m}$ and $\boldsymbol{n}$, respectively.

### 2.2. The entropy-based ranking algorithm based on overlap $\kappa$

Assessing the importance of nodes is essentially examining the extent to which network changes are caused by perturbations of the nodes in complex coupling. Most node-importance algorithms start from the network structure and define the importance of nodes by the size of the remaining maximum connected component in the network. We define the overlap $\kappa$ of the random and fixed low-rank parts of the network and, based on this, we propose a node-importance algorithm that considers the impact of overlap degree on network entropy. Specifically, we provide the specific steps of the algorithm

$$H_\kappa(i) = -\kappa_i \log_2(\kappa_i) - (1-\kappa_i)\log_2(1-\kappa_i), \tag{4}$$

where $\kappa_i$ is the overlap $\kappa$ of $i$th perturbation. The entropy formula we use is based on a variant of binary entropy also known as Shannon entropy. Shannon entropy is a fundamental concept in information theory used to measure the uncertainty of a random variable or the amount of information.[45] The original Shannon entropy formula is $H_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$. $x$ is a random variable that represents the uncertainty we need to measure or the amount of information itself, according to Shannon's entropy theory, and we turned $x$ to the overlap $\kappa$. Next, we describe $\kappa$ and the dynamic mean-field theory used.

### 2.3. Mean-field theory for dynamical networks

Our results rely on the mathematical analysis of network dynamics based on dynamic mean field theory (DMFT),[6,26,27] which averages the disorder source of the random part of connectivity to obtain a dynamic and effective description.

According to the last term in Eq. (1), the long-time properties of the solutions of Eq. (1) for large $N$ have been studied by using the DMFT originally developed for spin-glasses.[44] This theory is exact in the limit of $N \to \infty$. The dynamics of the system at long times can be reduced to a self-consistent equation of a single unit. Based on this theory, in different implementations of the Gaussian random matrix $\chi_{ij}$, it will generate a time-dependent Gaussian field by random inputs from the other units,[6] so that the input of node $i$ can be approximated as a Gaussian random process $\eta_i(t)$. We obtain an approximation as

$$\sum_{j=1}^{N} J_{ij}\phi\left(x_j(t)\right) \approx \eta_i(t). \tag{5}$$

Making each node $i$ obey the class Langevin equation,[44] obtained from Eqs. (1) and (5),

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -x_i(t) + \eta_i(t), \tag{6}$$

Gaussian random process $\eta_i(t)$ can, in principle, have different first-order statistics and second-order statistics for each unit, but they are statistically independent between different units. Therefore, $x_i$ of different units is also an independent Gaussian random process and can only be coupled through first- and second-order statistics.

The core of the DMFT is the self-consistent equation of the mean $\mu_i$ and the autocorrelation equation $\Delta_i(t)$. In the equilibrium state (without transient dynamics), the average $\mu_i$ for $x_i$ is obtained by averaging the random part of the connectivity in Eq. (1) over the entire sphere. For unit connections, it can be obtained as follows:

$$\mu_i = \kappa m_i, \tag{7}$$

with

$$\kappa = \frac{1}{N}\sum_{j=1}^{N} n_j \left[\phi_j\right], \tag{8}$$

where $\phi_j$ is a short form of $\phi(x_j(t))$ and $[\phi_j]$ represents the mean activation rate of the Gaussian transformation $x_j$ at node $j$. From a geometrical perspective, $\kappa_i$ represents the overlap between the left connection vector $\boldsymbol{n}$ and the average firing rate in $i$th perturbation, which is also given by the group mean value $n_j[\phi_j]$. At the same time, it can be represented as follows:

$$\kappa = \int \mathrm{d}\boldsymbol{m}\mathrm{d}\boldsymbol{n} p(\boldsymbol{m},\boldsymbol{n})\boldsymbol{n}\int \mathrm{D}z\phi\left(\boldsymbol{m}\kappa + \sqrt{\Delta_0}z\right), \tag{9}$$

where $p(\boldsymbol{m},\boldsymbol{n})$ represents the joint distribution of vectors $\boldsymbol{m}$ and $\boldsymbol{n}$, $\int \mathrm{D}z = \int_{-\infty}^{+\infty} \mathrm{e}^{-\frac{z^2}{2}}/\sqrt{2\pi}\mathrm{d}z$,[6] $\Delta_0$ is the variance of $x_i$ and the autocorrelation function $\Delta_i(t)$ quantifies the fluctuation of the activity quantity $x_i$ around the expected average value. The results of its calculation show that the fluctuations of the activation $x_i$ around the expected mean for all units in the network are same, that is, it is independent of unit $i$ (see Eq. (18)), so the auto-correlation function $\Delta_i(t)$ can be decomposed into a static variance that quantifies the balance of the random component fluctuations of $x_i$ in different realizations when the network is in a state of temporal chaos. An additional temporal variance can be written in static form as

$$\Delta_0 = g^2\frac{1}{N}\sum_{j=1}^{N}\left[\phi_i^2\right], \tag{10}$$

where $\left[\phi_i^2\right]$ is the mean value of $\phi_i^2(x)$ of Gaussian variable $x_i$.

Equations (8) and (10) indicate that both $\mu_i$ and $\Delta_0$ depend on the average and macroscopic quantities of the population.

### 2.4. Dynamic behavior of a single node

The DMFT relies on the assumption that when the network is sufficiently large, the disordered components of the coupling structure (represented here by $\chi_{ij}$) can effectively decouple the activities of individual nodes. Based on our study, we use the mean field theory applied to networks with a primary structure that has no external input. Firstly, according to Eqs. (2), (3) and (5), we will give the spontaneous dynamics equation of a single node to determine the effective noise $\eta_i$ for node $i$,

$$\eta_i(t) = g\sum_{j=1}^{N}\chi_{ij}\phi\left(x_j(t)\right) + \frac{m_i}{N}\sum_{j=1}^{N} n_j\phi\left(x_j(t)\right). \tag{11}$$

We will show that this assumption of de-correlated activity is self-consistent for the particular network structure we study, described in the standard DMFT by averaging over different realizations of the random matrix $\chi_{ij}$ for $\eta_i$.[6,28] $[\cdot]$ denotes the average over the random matrix $\chi_{ij}$ and $\langle\cdot\rangle$ denotes the average of the different units in the network. The network activity can be equivalently described using the variables $x_i(t)$ or their nonlinear transformation $\phi(x_i(t))$. Since these two quantities are not independent, the statistics of the distribution of the latter can be expressed in terms of the statistics of the former, and the average value of the effective noise received by unit $i$ is

$$\left[\eta_i(t)\right] = g\sum_{j=1}^{N}\left[\chi_{ij}\phi\left(x_j(t)\right)\right] + \frac{m_i}{N}\sum_{j=1}^{N} n_j\left[\phi\left(x_j(t)\right)\right]. \tag{12}$$

Under the hypothesis that, in large networks, neural activity decorrelates, more specifically, that activity $\phi(x_j(t))$ is independent of its outgoing weights, we have

$$\left[\eta_i(t)\right] = g\sum_{j=1}^{N}\left[\chi_{ij}\right]\left[\phi\left(x_j(t)\right)\right] + \frac{m_i}{N}\sum_{j=1}^{N} n_j\left[\phi\left(x_j(t)\right)\right] = m_i\kappa, \tag{13}$$

when $[\chi_{ij}] = 0$ we can get

$$\kappa := \frac{1}{N}\sum_{j=1}^{N} n_j\left[\phi\left(x_j(t)\right)\right] = \left\langle n_j\left[\phi\left(x_j(t)\right)\right]\right\rangle, \tag{14}$$

which quantifies the overlap between the mean population activity vector and the left-connectivity vector $\boldsymbol{n}$. Similarly, the noise correlation function is given by

$$\begin{aligned}&\left[\eta_i(t)\eta_j(t+\tau)\right]\\ =\ & g^2\sum_{k=1}^{N}\sum_{l=1}^{N}\left[\chi_{ik}\chi_{jl}\right]\left[\phi\left(x_k(t)\right)\phi\left(x_l(t+\tau)\right)\right]\\ &+\frac{m_i m_j}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N} n_k n_l\left[\phi\left(x_k(t)\right)\phi\left(x_l(t+\tau)\right)\right].\end{aligned} \tag{15}$$

The parameter $\tau$ represents a time lag representing the temporal separation between the observations of two nodes $i$ and $j$. Each cross term vanishes when $[\chi_{ij}] = 0$ at this point, which then indicates that there is no direct connection between $i$ and $j$. When we expand $[\eta_i(t)\eta_j(t+\tau)]$, we get four terms where the cross terms about $i$ and $j$, such as $\frac{gm_i}{N}\sum_{k=1}^{N}\sum_{l=1}^{N}[\chi_{ik}n_k\phi(x_k(t))\phi(x_l(t+\tau))]$ and $\frac{gm_j}{N}\sum_{k=1}^{N}\sum_{l=1}^{N}[\chi_{jl}n_l\phi(x_k(t))\phi(x_l(t+\tau))]$ vanish. However, in a similar way to the standard DMFT derivation,[6] the first term on the right-hand side vanishes in the cross correlation $(i \neq j)$ and is present in the autocorrelation function $(i = j)$ when $[\chi_{ik}\chi_{jl}] = \delta_{ij}\delta_{kl}/N$. We then obtain

$$\begin{aligned}&\left[\eta_i(t)\eta_j(t+\tau)\right]\\ =\ & \delta_{ij}g^2\left\langle\left[\phi_i(t)\phi_i(t+\tau)\right]\right\rangle\end{aligned}$$

$$+\frac{m_i m_j}{N^2}\sum_{k=1}^{N}\sum_{l=1}^{N} n_k n_l \left[\phi\left(x_k(t)\right)\phi\left(x_l(t+\tau)\right)\right]. \quad (16)$$

The second term on the right-hand side of the above equation corresponds to a sum containing $N$ terms, where $k=l$; the effect of which vanishes under massive $N$ due to the $1/N^2$ scaling. According to the assumptions we started with, when $k\neq l$, the activity is related to $[\phi(x_k(t))\phi(x_l(t+\tau))] = [\phi(x_k(t))][\phi(x_l(t+\tau))]$ and in the order of the $N$ priors, we get

$$\begin{aligned}
&[\eta_i(t)\eta_j(t+\tau)]\\
&= \delta_{ij}g^2\langle[\phi_i(t)\phi_i(t+\tau)]\rangle\\
&\quad+\frac{m_i m_j}{N^2}\sum_{k} n_k\left[\phi\left(x_k(t)\right)\right]\sum_{k\neq l} n_l\left[\phi\left(x_l(t+\tau)\right)\right]\\
&= \delta_{ij}g^2\langle[\phi_i(t)\phi_i(t+\tau)]\rangle + m_i m_j k^2, \quad (17)
\end{aligned}$$

so that

$$[\eta_i(t)\eta_j(t+\tau)] - [\eta_i(t)][\eta_j(t)] = \delta_{ij}g^2\langle[\phi_i(t)\phi_i(t+\tau)]\rangle. \quad (18)$$

Up to this point, we find that the effective input statistics are uncorrelated across units, so our initial assumptions are self-consistent. Finally, for each unit $i$, we compute the first- and second-order statistics of the effective input $\eta_i(t)$. The obtained expressions show that the individual noise statistics depend on the statistics of the whole network activity.

In particular, the average value of the effective inputs depends on the average overlap $\kappa$, but varies between units through the components of the right connectivity vector $\boldsymbol{m}$. On the other hand, the autocorrelation of the effective inputs is the same for all units and is determined by the overall average firing rate autocorrelation $\langle[\phi_i(t)\phi_i(t+\tau)]\rangle$. Once the statistic of $\eta_i(t)$ is determined, the Langevin-like stochastic process can be solved by Eq. (6) to obtain the autocorrelation solution of the activation variable $x_i(t)$.

## 3. Experiment and results

### 3.1. Network behavior perturbation experiment

According to the generated network connectivity matrix, we conducted network behavioral perturbation experiments on a fully connected network,[29] a random network,[30] a scale-free network[31] and a real network (Chicago arterial road network[32,33]). Based on the generated connectivity, we calculated the influence of nodes on network entropy value and removed nodes sorted by importance. We then compared three methods, namely, entropy ranking, random removal and degree[34] removal. In fully connected networks, we replaced degree ranking with sequential ranking. The experimental results showed that removing 20% to 40% of the important nodes sorted by entropy ranking could minimize the network's behavioral value. This method was significantly better than others in fully connected networks, ER networks ($K=40$, $N=200$), BA networks ($\gamma=30$, $N=200$) and the Chicago arterial road network ($N=933$), see Fig. 1 for details.

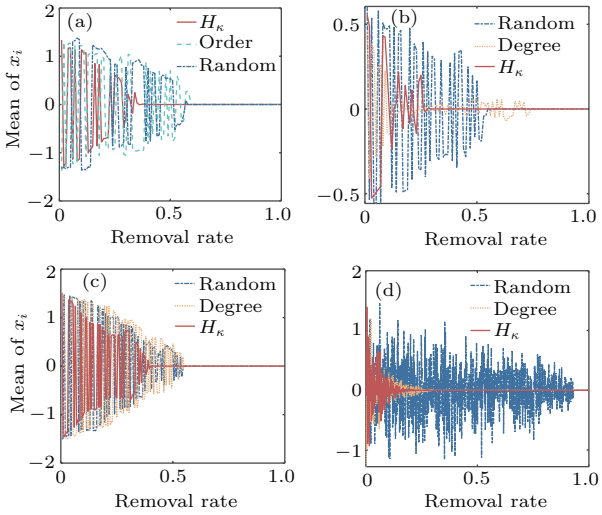



**Fig. 1.** Performance of different algorithms for the behavior perturbation experiment. Due to the stochastic nature of network models, except for the real road network (d), we conducted 1000 experiments and obtained the average values of the following results: (a) fully connected network, $N=1000$, with no degree sorting and added sequential sorting for comparison; (b) scale-free network, $\gamma=30$, $N=200$; (c) ER network, $K=40$, $N=200$; (d) Chicago arterial road network.[32,33] The horizontal axis represents the proportion of removed nodes from the network according to the sorting result and the vertical axis represents the average behavioral value of the remaining network.

### 3.2. Correlation matrix

Based on the results of network behavior perturbation, we compared the entropy-ranking method with degree ranking, random ranking and sequential ranking, and observed the node selection situation. We used Pearson correlation analysis[35] to compare the sorting results from all of the algorithms. It can be seen from Fig. 2(a) that their correlation is low, indicating that the entropy-ranking method has richer information compared to other methods. We believe that this is a very important but easily overlooked indicator, because the low correlation between proposed and known algorithm suggests that proposed algorithm is more likely to provide insights beyond the known algorithm.

To further illustrate the differences between the methods, we performed a visual analysis of the network selection. We do not provide any additional illustration of random and sequential ordering here, but only compare the difference in selection between the degree method and our entropy sorting, which illustrates that entropy sorting method has more informative information as compared to the other methods. Regarding the additional information, we have visualized the network based on the experiments in Fig. 1(c) and marked the difference between the points selected by the degree method and those selected by our method. The size of the circle represents the importance of the nodes and the red nodes are the nodes selected by our algorithm. It can be seen that it does not present a clear aggregation state like the degree algorithm, which suggests that our algorithm can more comprehensively capture the information about the network nodes.

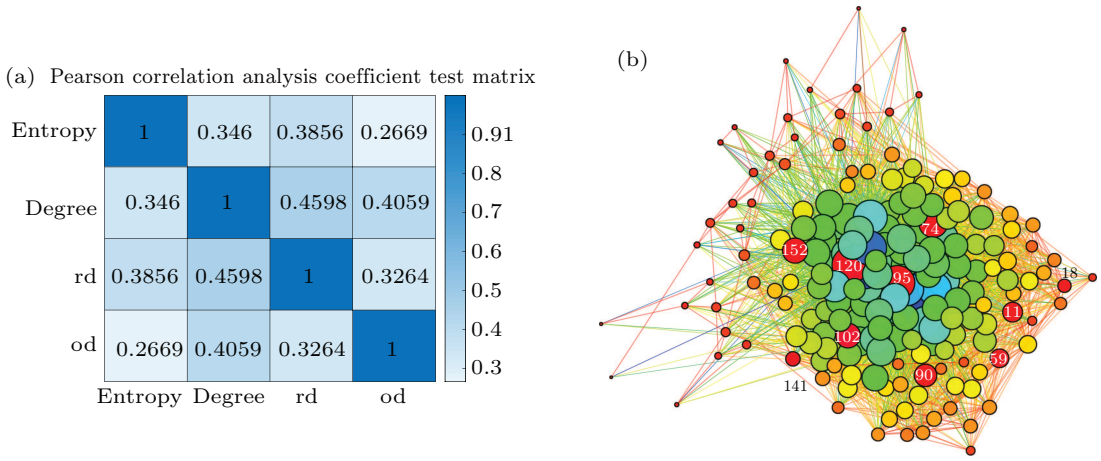



**Fig. 2.** Correlation analysis and point selection visualization. (a) Pearson correlation matrix. The table shows that $H_\kappa$ represents the entropy ranking, 'degree' represents the degree ranking, 'rd' represents random ranking and 'od' represents sequential ranking. Pearson correlation analysis was used to compare the correlation between the methods. (b) Network critical node visualization. This visualization showcases the application of the degree algorithm in selecting crucial nodes within a network. Node radius indicates the ranking, with larger radii denoting higher rankings (i.e., larger degrees). We performed node ranking based on entropy in the same network, highlighting the top ten nodes in red.

### 3.3. System state disturbance experiment

In a systemic evaluation, we gauge the system's state by monitoring the temporal evolution of individual behaviors across the entire network. Previous studies commonly assert that when the behavioral values of the majority of nodes exhibit random fluctuations, the system is considered chaotic. Conversely, if individual behaviors display consistent and non-random fluctuations over time, the system is deemed stable. Typically, a well-functioning system tends to be stable. However, if the intention is to induce an abnormal operational state, a straightforward approach is to introduce chaos into an initially stable system. In this context, we employ entropy to quantify the degree of chaos within the system and introduce a node importance algorithm to rank network nodes. Perturbations are then applied to nodes based on their importance and subsequent alterations in the network state are observed.

To validate our algorithm, we conducted state perturbation experiments on random networks. Initially, the connection matrix of the random network was generated according to the network model. The temporal fluctuation of each node's initial state was then computed and documented. Subsequently, based on this initial state, networks were categorized into chaotic and stable states. Leveraging our proposed node importance algorithm, nodes were prioritized under the initial conditions and the activity of nodes was recalculated after each perturbation according to their respective importance. The pseudo-code of procedure is given in Algorithm 1.

We also use degree sorting to perform the same operation on the same network to compare the differences in terms of the effect of degree sorting and entropy sorting under the same perturbation ratio. In Figs. 3(a)–3(d), we carried out experiments comparing two states and four methods on ER network and scale-free network respectively. After many experiments, it can be clearly seen that our entropy-sorting method is very effective in changing the state of the network. The state of the

**Algorithm 1** Node-importance algorithm pseudo-code

1: function CREATEMATRIX($G, Mm, Mn, Sim, Sin, g$)
2: $R \leftarrow g \cdot \text{normrnd}(0, \sqrt{\frac{1}{N}}, N, N)$ ▷ Connectivity randomized partial matrix
3: $\text{normrnd}(Mm, Sim, 1, N) \rightarrow m, \text{normrnd}(Mn, Sin, N, 1) \rightarrow n$
4: $(m \cdot n)/N, R + M \rightarrow M, J$ ▷ $M$ is the smallest low-dimensional part of the connectivity matrix
5: **end function**
6: **for all** Node $i$ in $G$
7: SimulateActivity($J$)
8: ode45(@Integrate, $t, x_0$, options, $J$)
9: **function** INTEGRATE($t, x, J$)
10: **return** $x$
11: **end function**
12: $\text{overlap}(x, m, n, Mm, Mn) \rightarrow \kappa$
13: $\text{Entropy}(N, \kappa) \rightarrow H(\kappa)$
14: **end for**
15: Record Rank List ▷ Record the change in network entropy after each perturbation and use it to rank the nodes in terms of importance

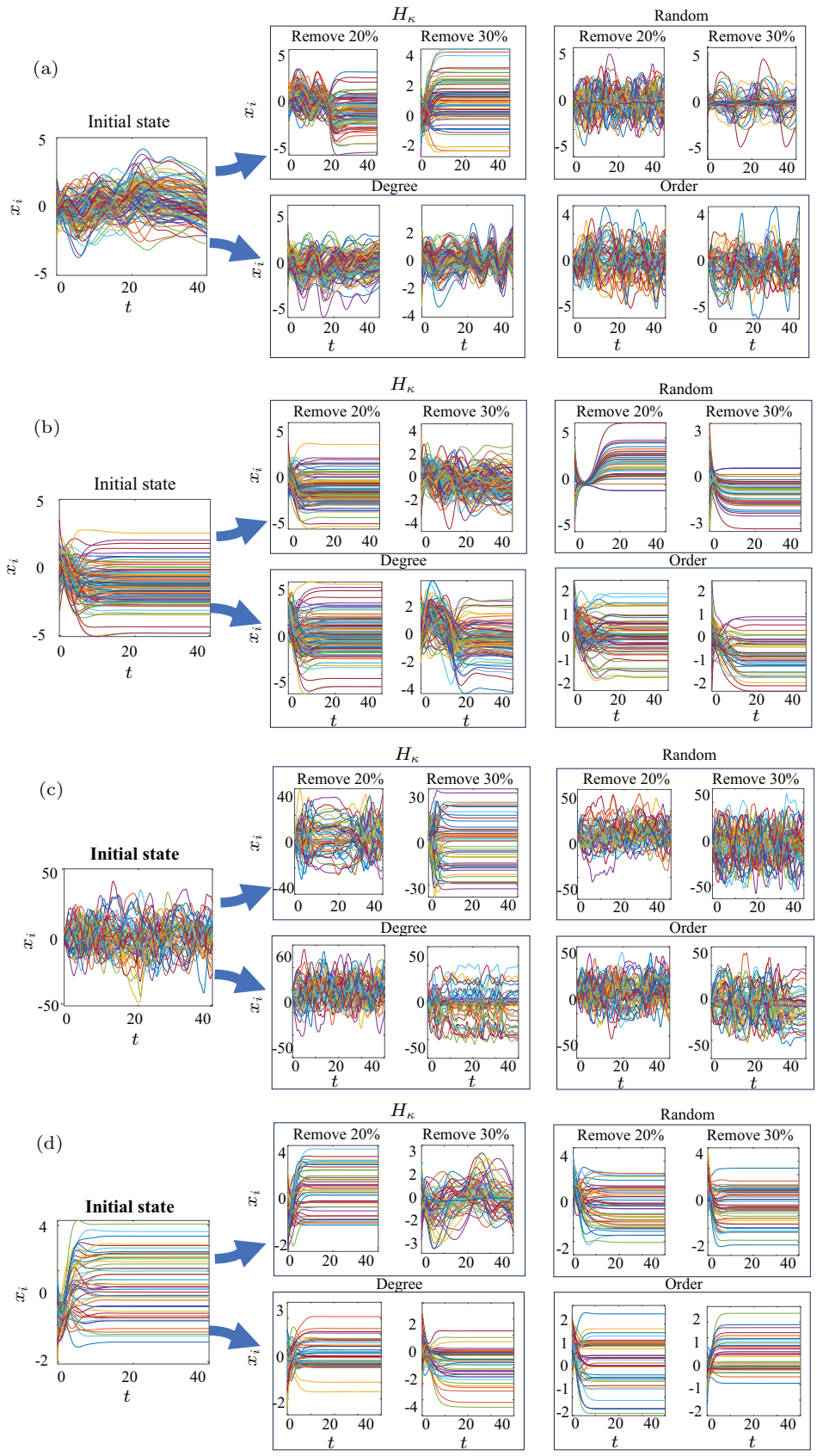



**Fig. 3.** Comparison of system perturbation effects. The horizontal axis represents time and the vertical axis represents the activity of the nodes as they fluctuate over time. Each line represents a different node in the network. (a) The initial state is a chaotic ER network ($N = 50$, $k = 10$) and the change in remaining node activity after removing 20% and 30% of nodes sorted by entropy and degree is compared. (b) The initial state is a stable ER network ($N = 50$, $k = 10$) and the change in remaining node activity after removing 20% and 30% of nodes sorted by entropy and degree is compared. (c) The initial state is a chaotic BA network ($N = 50$, $\gamma = 10$) and the change in remaining node activity after removing 20% and 30% of nodes sorted by entropy and degree is compared. (d) The initial state is a stable BA network ($N = 50$, $\gamma = 10$) and the change in remaining node activity after removing 20% and 30% of nodes sorted by entropy and degree is compared.

network using entropy sorting changes significantly after removing the same proportion of important nodes given according to each algorithm. For a chaotic network as its initial state (see Figs. 3(a) and 3(c)), we found that when the proportion of perturbed nodes according to entropy sorting accounts for 20% of all nodes, the network state gradually tends to be stable. When the number of perturbed nodes increases to 30%, the network state reaches stability from the original chaotic state. With degree sorting, the same proportion of node perturbation still keeps the system in a chaotic state. For the case of initial state stabilization, entropy sorting is still better than other methods for the change of network state, see Figs. 3(b) and 3(d). This observation suggests that the network system is more likely to transition from a chaotic state to a stable state under perturbed nodes ordered by $H_K(i)$, whereas the degree-ordered perturbations fail to achieve the same stabilizing effect. This may imply that nodes sorted by entropy play a more important role in the network structure and state evolution, and their perturbations have a more significant effect on the overall stabilization of the system. This also highlights that the choice of node-ordering method has an important impact on the network state evolution and that ordering by entropy may be more effective in steering the network towards a stable state.

### 3.4. Comparing the entropy values during state changes

As we use entropy as a metric to perturb nodes, it can cause originally chaotic or stable systems to become stable or chaotic. Therefore, we recorded the entropy value of the system in different states. When we changed the system from stable to chaotic and then back to stable (i.e., stable → chaotic → stable), we found that the entropy value of the same stable state network was different, but relatively close, see results in Fig. 4.

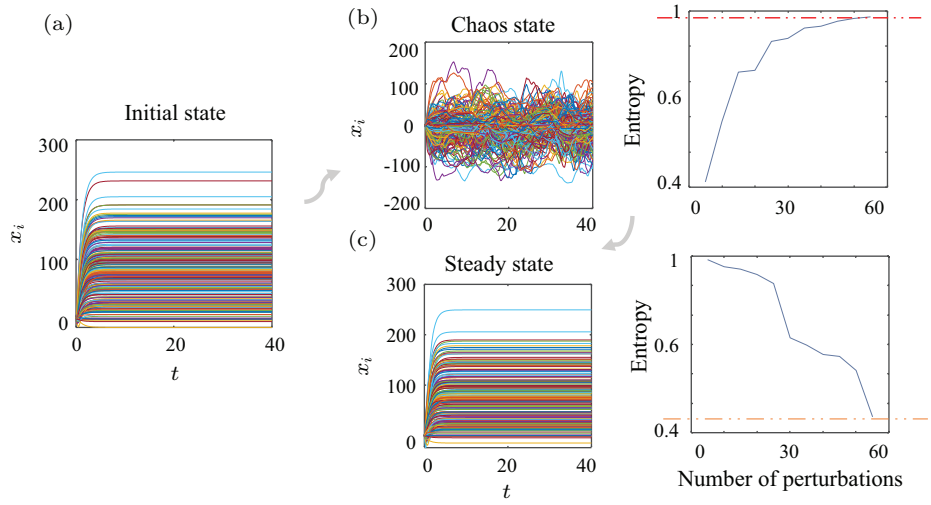



**Fig. 4.** State changes and critical entropy values. (a) The change of the node behavior values over time in the initial state of the network. On the left side of (b), the initial network is disturbed to reach a chaotic state, while on the right side, the change of entropy value is shown. On the left side of (c), a chaotic network from (b) is used as the initial state to control the nodes and restore the network to a stable state, with the change of entropy value shown on the right.

### 3.5. Disturbance of the *C. elegans* network

The nervous system of an animal changes as its body grows and its behavior matures from birth to adulthood.[24–27,36–39,42] In the study by Witvliet *et al.*,[40] eight homologous *C. elegans* whole-brain reconstructions were created using consecutive sections of electron microscopy to investigate how it changes with age and to compare the connectomes of different developmental stages. It was found that the consistent wiring changes between different neurons reveal continual changes. They divided the neurons into four categories (see Fig. 5 for details), namely, sensory, inter, motor and modulatory, with feedforward and feedback connections between these neurons that ultimately act on muscles together. We used the network model in Subsection 2.1 to process the *C. elegans* network and used the entropy-ranking method to analyze the entire network and its functional layers. We selected the nodes that have the greatest impact on the network state and observed the degree to which each layer's nodes influence the muscle layer.

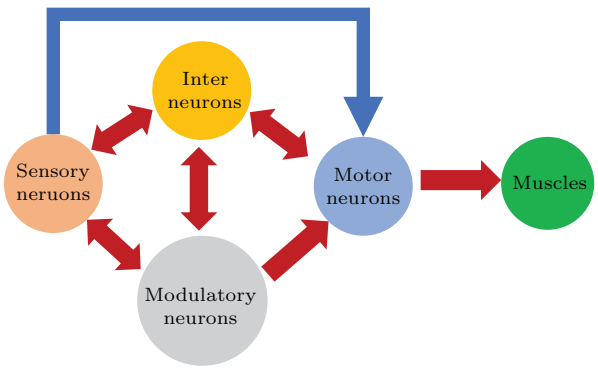



**Fig. 5.** The neural layer relationships of *C. elegans*. The circles represent the corresponding neural layers and the direction of the arrows indicates a connection between two layers of neurons.

### 3.5.1. Overall disturbance

Firstly, the overall control of the *C. elegans* neural network was carried out, with the initial state of each node obtained. Then, the importance of each node was ranked using entropy and degree ranking, and perturbation was carried out. Experimental results show that, without perturbation, the activity level of each node in Fig. 6 did not change over time, indicating that the network was in a stable state.

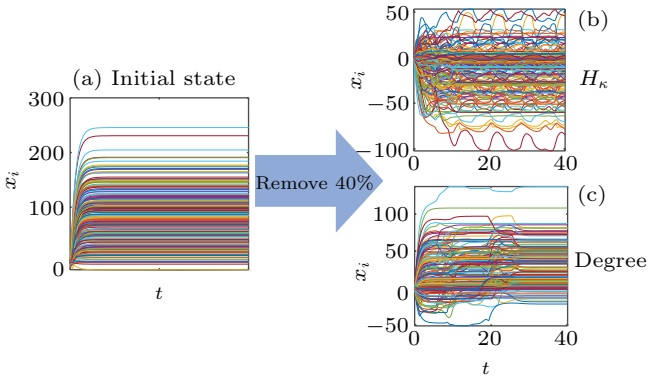



**Fig. 6.** The overall perturbation of the *C. elegans* neural network. (a) The activity of each node over time when the network is not perturbed. (b) The state of remaining network nodes over time after removing nodes based on the $H_\kappa(i)$ method. (c) The state of the remaining network over time after removing nodes based on the degree-ranking method. The horizontal axis represents time and the vertical axis represents the behavioral value of nodes.

To achieve perturbation and disruption of the network, the nodes in the network were ranked according to importance using entropy and degree ranking, and the nodes were perturbed in descending order. After 40% of the nodes were removed, the result shown in the upper right of Fig. 6(b) indicates that, when the network was perturbed by $H_\kappa$ ranking, most of the nodes in the network exhibited significant irregular fluctuations over time, indicating that the network was in a chaotic state. This chaotic state is reflected in the *C. elegans* brain nerve cells, leading to chaotic muscle behavior. Figure 6(c) shows the change of node activity over time when perturbed by degree ranking. The behavioral values of the nodes were different from the initial state, but the behavioral values of the nodes were in a state of irregular fluctuation before $t = 20$. After that, the behavioral values of the nodes no longer fluctuated over time, indicating that the overall network state was still stable. This experiment demonstrated the effectiveness of selecting nodes using entropy ranking to control the *C. elegans* neural network, providing a basis for the subsequent layered control experiments.

### 3.5.2. Layered disturbance

The data used in this study comes from the research of Witvliet *et al.*[40] They divided the *C. elegans* neural network into four layers of neurons and a muscle layer, with the four layers of neurons being sensory, inter, motor and modulatory. We obtained Fig. 5 based on the relationship between different layers of neurons. From Fig. 5, it can be observed that there are feedback and feedforward connections among the four layers of neurons, which affect each other's neural activities. Ultimately, these neural activities act on muscles through the motor layer. The muscle layer is divided into 32 different types of muscle cells, which are impacted by the four layers of neurons. Considering the existence of neural signal transmission between different layers, we applied the entropy-ranking method to perturb the same proportion of nodes in the neural network layers, except for the motor layer which is directly related to muscles. Then, we observed the changes in the activity levels of each muscle cell node in the muscle layer after perturbation.

We recorded the initial state of the muscle layer when the other neuron layers were undisturbed. Then, we sorted the entropy of each neuron node in each layer and disturbed nodes in the same proportion to observe the state changes of each node in the muscle layer. The experimental results in Fig. 7 show that the initial state is stable, indicating that the behavior of the

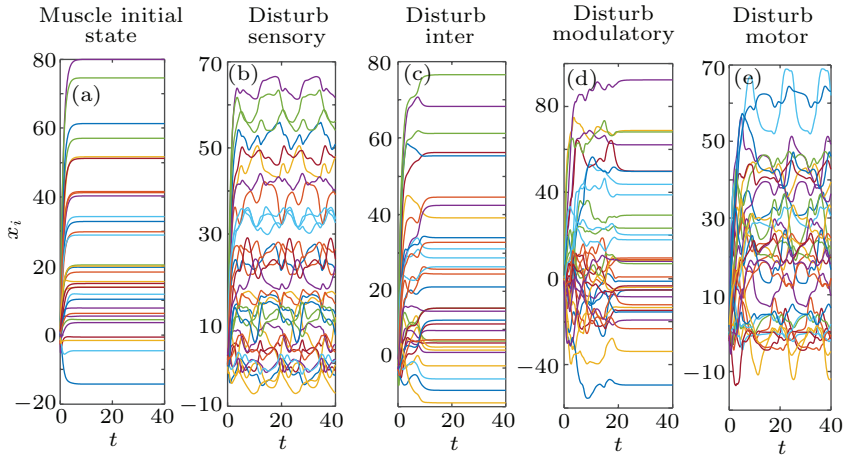



**Fig. 7.** Changes of muscle layer node behavior over time after disturbing each layer. Time is on the horizontal axis and the node behavior value is on the vertical axis. (a) The initial state of each node in the muscle layer. (b)–(e) The state changes of each node in the muscle layer after disturbing the sensory, inter, modulatory and motor layers, respectively, according to $H_\kappa$ ranking.

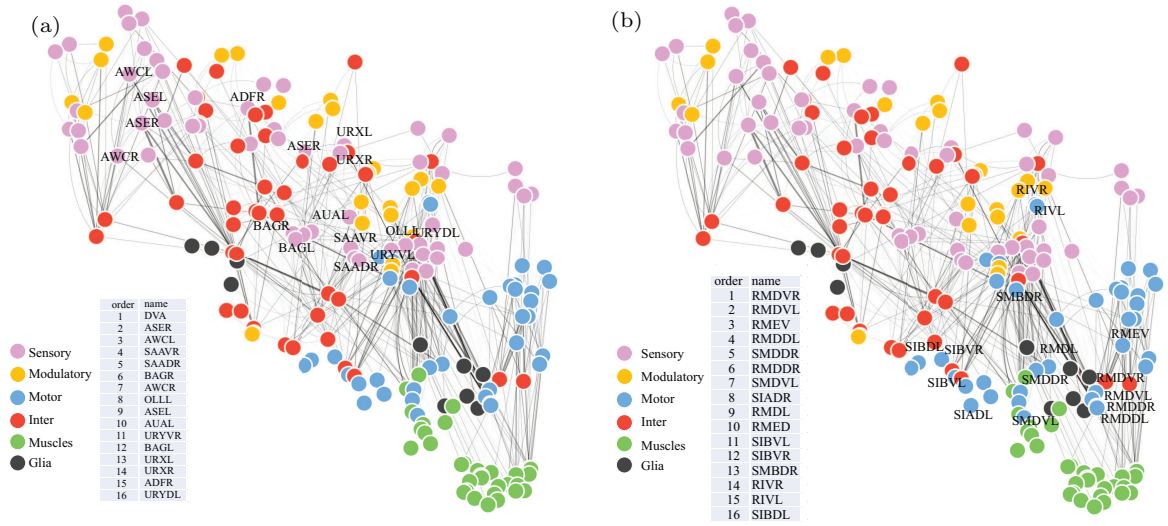



**Fig. 8.** The important nodes in the *C. elegans* network. The color of the circle represents the cell type, each line represents a connection between at least one chemical synapse of the two cells and the line width represents the number of synapses. After multiple experiments, the important nodes that ranked in the top 40% in the sensory and motor layers, respectively, are shown in (a) and (b). The list in the figure shows the corresponding node names and their importance rankings.

*C. elegans* brain cells is normal. We observed that disturbance of the inter and modulatory layers had a certain impact on the performance of muscle cells, but the network state was relatively stable. After disturbing the sensory layer, the changes in the muscle layer were relatively large, and most nodes showed chaotic oscillation. We believe that the reason for this phenomenon is that the sensory layer is connected to each other layer and forms the foundation of the entire cell. Therefore, even if it is not directly connected to muscles, disturbing the sensory layer will have a greater influence on the muscle layer. After disturbing the motor layer, the state of the muscle layer changed from stable to chaotic. This indicates that the muscle behavior of the experimental subject is already chaotic. We believe that this is because the motor layer is directly related to the muscle layer. In Fig. 8, we will represent important nodes in the motor and sensory layers in the *C. elegans* network.

## 4. Discussion and conclusion

In this paper, a combination of random components and a minimum low-dimensional structure is used to represent the connectivity model of the network. Furthermore, mean-field theory is used to derive the activity of network nodes. The overlap of the random part and the low-dimensional part is calculated, and the degree of influence of each node on the network state change is calculated and quantified using entropy. Finally, the influence of each node on the network entropy change is ranked according to the importance of this influence.

We conducted network behavior disturbance experiments on fully connected networks, random networks, scale-free networks and the Chicago backbone network using the newly proposed importance algorithm. Our findings show that this algorithm not only has superiority in changing node behavior, but also can effectively control the overall state of the network. Through system state disturbance experiments conducted on random networks, we demonstrated that the important nodes in the network identified by our algorithm can also effectively change the network state. We further applied this method to the *C. elegans* neural network to control the network both globally and locally.

Based on our algorithm and experimental results, this paper draws the following conclusions: (1) The overlapping degree of the random part and the fixed low-dimensional part contains information about the impact on network states. (2) Using entropy, we can determine the degree to which a node affects network states and selectively use these important nodes to change the network state. For example, we can disrupt the system behavior by making a stable network chaotic, or conversely, stabilize a chaotic network to restore normal function.

In conclusion, our work opens up exciting possibilities for the controlled manipulation of network behavior, with applications ranging from critical infrastructure management to biological system control. We look forward to delving into these future endeavors, contributing to the advancement of network science and addressing the complex challenges posed by dynamic networks in our ever-evolving world. Our future work will focus on assessing the robustness of networks under the influence of our importance algorithm. We aim to investigate how the targeted manipulation of specific nodes can enhance or compromise network resilience and how this can be leveraged for better network performance. We plan to delve deeper into the study of biological networks. This includes investigating applications in gene regulatory networks, metabolic networks and protein–protein interaction networks, which can have profound implications for understanding biological processes and diseases.

## Acknowledgements

Project supported by the National Natural Science Foundation of China (Grant Nos. 72071153 and 72231008), Laboratory of Science and Technology on Integrated Logistics Support Foundation (Grant No. 6142003190102), and the Natural Science Foundation of Shannxi Province (Grant No. 2020JM-486).

## References

[1] Strogatz S H 2001 *Nature* **410** 268
[2] Latora V, Nicosia V and Russo G 2017 *Complex networks: principles, methods and applications* (Cambridge University Press) pp. 679–692
[3] Erdős P and Rényi A 1961 *Acta Mathematica Hungarica* **12** 261
[4] Janson S, Rucinski A and Luczak T 2011 *Random graphs*. (John Wiley & Sons) pp. 233–270
[5] Watts D J and Strogatz S H 1998 *Nature* **393** 440
[6] Sompolinsky H, Crisanti A and Sommers H J 1988 *Safety Science* **61** 259
[7] Brunel N 2000 *Journal of Computational Neuroscience* **8** 183
[8] Van V C and Sompolinsky H 1996 *Science* **274** 1724
[9] Williamson R C, Cowley B R, Litwin-Kumar A, Doiron B, Kohn A, Smith M A and Yu B M 2016 *PLoS Computational Biology* **12** e1005141
[10] Wang X J 2002 *Neuron* **36** 955
[11] Amit D J and Brunel N 197 *Phys. Rev. Lett.* **7** 237
[12] Litwin K A and Doiron B 2012 *Nature Neuroscience* **15** 1498
[13] Harris K D and Mrsic F T D 2013 *Nature* **503** 51
[14] Rigotti M, Barak O, Warden M R, Wang X J, Daw N D, Miller E K and Fusi S 2013 *Nature* **497** 585
[15] Mante V, Sussillo D, Shenoy K V and Newsome W T 2013 *Nature* **503** 78
[16] Churchland M M and Shenoy K V 2007 *Journal of Neurophysiology* **97** 4235
[17] Mastrogiuseppe F and Ostojic S 2018 *Neuron* **99** 609
[18] Pei S, Teng X, Shaman J, Morone F and Makse H A 2016 arXiv:0809.0872v1
[19] Rényi A 1961 *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, June20–July 30, 1961, California, US, pp. 547–562
[20] Kapur J N and Kesavan H K 1992 *Entropy optimization principles and their applications* (Springer) pp. 3–20
[21] Gray R M 2011 *Entropy and information theory* (Springer Science & Business Media) pp 61–59
[22] Dehmer M 2008 *Applied Mathematics and Computation* **201** 81
[23] Omar Y M and Plapper P 2020 *Entropy* **22** 1417
[24] Randel N, Shahidi R, Veraszto C, Bezares C L A, Schmidt S and Jekely G 2015 *Elife* **4** e08069
[25] Kornfeld J, Januszewski M, Schubert P, Jain V, Denk W and Fee M S 2020 *BioRxiv* **2020** 2
[26] Schneider M C M, Bodor A L, Collman F and Brittain D 2020 *BioRxiv* **2020** 3
[27] Aicher C, Jacobs A Z and Clauset A 2015 *Journal of Complex Networks* **3** 221
[28] Rajan K, Abbott L F and Sompolinsky H 2008 *Phys. Rev. E* **2010** 011903
[29] Schwing A G and Urtasun R 2015 arXiv:1503.02351
[30] Bollobás B 1998 *Random graphs* (Springer) pp. 34–50
[31] Barabási A and Albert R 1999 *Science* **286** 509
[32] Eash R, Chon K, Lee Y and Boyce D 1979 *Transportation Research* **13** 243
[33] Boyce D E, Chon K S, Ferris M E, Lee Y J, Lin K T and Eash R W 1985 *TRID* **1985-6** 181
[34] Kitsak M, Gallos L K, Havlin S, Liljeros F, Muchnik L, Stanley H E and Makse H A 2010 *Nat. Phys.* **11** 888
[35] Rodgers J L and Nicewander W A 1988 *American Statistician* **42** 59
[36] Jin X, Pokala N and Bargmann C I 2016 *Cell* **164** 632
[37] White J G, Albertson D G and Annes M A R 1978 *Nature* **271** 764
[38] Bargmann C I and Marder E 2013 *Nat. Methods* **10** 483
[39] Varshney L R, Chen B L, Paniagua E, Hall D H and Chklovskii D B 2011 *PLoS Computational Biology* **7** e1001066
[40] Witvliet D, Mulcahy B, Mitchell J K, Meirovitch Y, Parvathala R and Holmyard D 2021 *Nature* **596** 257
[41] Wu C X, Duan D L and Xiao R B 2023 *Physica A* **620** 128727
[42] Duan D L, Wu X X and Si S B 2021 *Frontiers of Engineering Management* **8** 527
[43] Pei S, Teng X, Shaman J, Morone F and Makse H A 2016 arXiv:1606.02739
[44] Sompolinsky H and Zippelius A 2011 *Phys. Rev. B* **25** 6860
[45] Lin J 1991 *IEEE Transactions on Information Theory* **37** 145