Predicting and Controlling Complex Networks

by

Yuzhong Chen

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2016 by the
Graduate Supervisory Committee:

Ying-Cheng Lai, Chair
Andreas Spanias
Cihan Tepedelenlioglu
Lei Ying

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

The research on the topology and dynamics of complex networks is one of the most focused area in complex system science. The goals are to structure our understanding of the real-world social, economical, technological, and biological systems in the aspect of networks consisting a large number of interacting units and to develop corresponding detection, prediction, and control strategies. In this highly interdisciplinary field, my research mainly concentrates on universal estimation schemes, physical controllability, as well as mechanisms behind extreme events and cascading failure for complex networked systems.

Revealing the underlying structure and dynamics of complex networked systems from observed data without of any specific prior information is of fundamental importance to science, engineering, and society. We articulate a Markov network based model, the sparse dynamical Boltzmann machine (SDBM), as a universal network structural estimator and dynamics approximator based on techniques including compressive sensing and K-means algorithm. It recovers the network structure of the original system and predicts its short-term or even long-term dynamical behavior for a large variety of representative dynamical processes on model and real-world complex networks.

One of the most challenging problems in complex dynamical systems is to control complex networks. Upon finding that the energy required to approach a target state with reasonable precision is often unbearably large, and the energy of controlling a set of networks with similar structural properties follows a fat-tail distribution, we identify fundamental structural “short boards” that play a dominant role in the enormous energy and offer a theoretical interpretation for the fat-tail distribution and simple strategies to significantly reduce the energy.

Extreme events and cascading failure, a type of collective behavior in complex networked systems, often have catastrophic consequences. Utilizing transportation and evolutionary game dy-

namics as prototypical settings, we investigate the emergence of extreme events in simplex complex networks, mobile ad-hoc networks and multi-layer interdependent networks. A striking resonance-like phenomenon and the emergence of global-scale cascading breakdown are discovered. We derive analytic theories to understand the mechanism of control at a quantitative level and articulate cost-effective control schemes to significantly suppress extreme events and the cascading process.

To my parents

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude to my advisor, Dr. Ying-Cheng Lai, for his guidance, mentoring, and continuous support over the past few years, which I couldn't have achieved so much without. His great personality, diligence and conscientious attitude toward research, and captivating ideas have educated and helped me throughout my Ph.D. studies.

I want to take this opportunity to thank and Dr. Zigang Huang, Dr. Liang Huang, and Dr. Wenxu Wang, the former research scientists of our group as well as sincere friends of mine, who have helped me in so many aspects. Dr. Zigang Huang, an old friend ever since my Master thesis, helped me a lot in my doctorate research works. His kindness and generosity always make the collaboration with him enjoyable. It was Dr. Liang Huang's recommendation that offer me the opportunity to join this research group. I also learned a lot from Dr. Liang Huang and Dr. Wenxu Wang, who taught me and inspired me with their solid physics background, broad interest and knowledge, and deep understanding of physical problems.

I always value the excellent research atmosphere built by our Chaos group members, Lezhi Wang, Junjie Jiang, Riqi Su, Guanglei Wang, Hongya Xu, Lei Ying, Nan Yao, and Zhidan Zhao. The fruitful discussions and collaborations with them made me feel so supportive and encouraged all these years, and the friendship will take a lifetime to cherish.

I also want to thank Dr. Shouhuai Xu for providing inspiration and constructive critiques on many of my projects. Also, I want to thank Dr. Thomas Seager for his helpful suggestions and collaboration which made the great topics possible.

I'm grateful to Dr. Yanchao Zhang, Dr. Lingfei Wu, Dr. Pan Zhang, and Dr. Xuan Ni for their insightful knowledge and helpful discussions on many interesting research problems.

My special thanks to the committee members, Dr. Andreas Spanias, Dr. Cihan Tepedelenlioglu,

and Dr. Lei Ying, for their precious time as well as invaluable comments and advices.

Finally, I would like to express deepest thankfulness to my family for all their love and support, especially for my parents who raised me with a love of science and supported me in all my pursuits. The completion of this thesis also means a lot to them. Here I dedicate my works to my beloved parents, without whose affection and encouragement none of my accomplishments would have been possible.

TABLE OF CONTENTS

	Page
LIST OF TABLES	xi
LIST OF FIGURES	xv
CHAPTER	
1 INTRODUCTION	1
1.1. Structure and Dynamics Estimation for Complex Networks	1
1.2. The Paradox of Controlling Complex Networks	2
1.3. Extreme Events and Cascading Failure on Complex Networks	3
1.4. Spatiotemporal Patterns and Predictability of Cyberattacks	4
2 UNIVERSAL STRUCTURAL ESTIMATOR AND DYNAMICS APPROXIMATOR FOR COMPLEX NETWORKS	6
2.1. Introduction	6
2.2. Results	11
2.2.1. SDBM as a Network Structural Estimator	11
2.2.2. Universality of SDBM as a Network Structural Estimator	17
2.2.3. Parameter Estimation Scheme	26
2.2.4. Universal Dynamics Approximator	28
2.2.5. Model Complexity and Representation Power	34
2.2.6. Potential for SDBM as a Generative Model?	35
2.3. Discussion	37
2.4. Materials and Methods	40
2.4.1. Structural Estimation Details.	40
2.4.2. Conflict Resolution.	41

	Page
2.4.3. Implementation of the K-Means Algorithm.	41
2.4.4. Link Weights and Node Biases Estimation.	44
2.4.5. Degree Guided Solution Substitution Operation.	45
3 RECOVERING INTERDEPENDENT STRUCTURE OF MULTI-LAYER COMPLEX NETWORKS	47
3.1. Introduction	47
3.2. Results	48
3.3. Discussion	57
3.4. Methods	59
4 THE PARADOX OF CONTROLLING COMPLEX NETWORKS: CONTROL INPUTS VERSUS ENERGY REQUIREMENT	61
4.1. Introduction	61
4.2. Resolution of Control Paradox	64
4.2.1. Gramian Matrix and Quantification of Control Energy.	64
4.2.2. Resolution of Control Paradox.	65
4.3. Distribution of Control Energy and Concept of Control Chains	68
4.4. Control Energy Optimization of Modeled and Real-World Networks	74
4.4.1. Practical Controllability of An Electrical Circuit Network.	74
4.4.2. Strategies to Balance Control Energy and Extra Inputs.	75
4.4.3. Control Energy Optimization of Modeled Complex Networks.	76
4.4.4. Practical Controllability of Real-World Networks.	77
4.5. Discussion	79
5 CONTROLLING EXTREME EVENTS ON COMPLEX NETWORKS	86

	Page
5.1. Introduction	86
5.2. Results	89
5.2.1. The Existence of Optimal Mobility and Its Robustness	89
5.2.2. Explanation and Boundary Effect	92
5.2.3. The Effect of Wormholes	98
5.2.4. Suppression of Extreme Events in Intermittently Mobile Networks	98
5.2.5. Degree Bias and Scaling	101
5.3. Discussion	101
5.4. Methods	102
5.4.1. Relation Between $\xi(V)$ and $\eta(V)$	102
5.4.2. Detailed Analytical Explanation to the Non-Monotonous Behavior	104
5.4.3. Self-Consistent Equation to Estimate the Number of Extreme Events in ER Random Networks	108
5.4.4. Effect of Degree-Degree Correlation	109
6 RESILIENCE OF INTERDEPENDENT COMPLEX NETWORKS - EMERGENCE OF EXTREME EVENTS AND CONTROL	111
6.1. Introduction	111
6.2. Results	113
6.2.1. Global Extreme Events and Two Absorption States	116
6.2.2. First-Order Phase Transition	117
6.2.3. Time Evolution and Packet Lifetime	118
6.2.4. Positive Feedback Loop	120
6.2.5. Betweenness and Effective Betweenness	121

	Page
6.2.6. Control Strategies to Suppress Extreme Events	126
6.3. Discussion	127
6.4. Methods	129
6.4.1. Model Details	129
6.4.2. Total Number of Packets in a Layer	130
6.4.3. Effect of Rerouting Via Random Walk	132
6.4.4. Effective Betweenness Calculation and Flux Distribution	136
7 OPTIMIZING COOPERATION ON COMPLEX NETWORKS IN THE PRESENCE OF FAILURE	137
7.1. Introduction	137
7.2. Results	138
7.3. Discussion	146
8 SPATIOTEMPORAL PATTERNS AND PREDICTABILITY OF CYBERATTACKS	147
8.1. Introduction	147
8.2. Results	150
8.2.1. Deterministic Attacks	153
8.2.2. Flux-Fluctuation Relation and Stochastic Attack	155
8.2.3. Spatial Concentration of Attacks	159
8.2.4. Inference Probability of Attack Patterns	161
8.2.5. Predictability of Cyberattacks	165
8.3. Discussion	168
8.3.1. How Do Attackers Choose Their Targets?	168
8.3.2. Interplay Between Different Attack Patterns	169

	Page
8.3.3. Attack Pattern Inference and Prediction	169
8.3.4. Possible Improvement for Future IDS	169
8.3.5. About the IP Information of the Attackers	170
8.4. Materials and Methods	170
REFERENCES	173
APPENDIX	
A UNIVERSAL STRUCTURAL ESTIMATOR AND DYNAMICS APPROXIMATOR FOR COMPLEX NETWORKS	194
REFERENCES	208
B THE PARADOX OF CONTROLLING COMPLEX NETWORKS: CONTROL INPUTS VERSUS ENERGY REQUIREMENT	211
REFERENCES	242

LIST OF TABLES

Table	Page
1. Description of the 14 Dynamical Processes on Model Networks and the Conditional Probabilities. The quantity $P_i^{0 \rightarrow 1}$ or $P_i^{1 \rightarrow 0}$ is the probability that the state of node i (of degree k_i) becomes 1 at the next time step when its current state is 0, or vice versa, conditioned on its current neighbor state configuration. The number of i 's neighbors in state 1 is n_i . (See Supplementary Note 1 for a detailed description of the dynamics and parameters for estimating the conditional probabilities $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$.)	21
2. Reconstruction Error Rates. For the same dynamical processes in Tab. 1, the error rates (in percentages) in uncovering the existent and non-existent links, R_e^1 and R_e^0 , respectively. ER-random and BA scale-free of size $N = 100$ and average degree $\langle k \rangle = 4$ are used as the underlining network supporting the various dynamical processes. The normalized number of linear measurements used in compressive sensing is $M/N = 0.4$ for all cases.	22

3.	Structure Reconstruction Error Rates of 5 Real-World Networks Associated with 11 Typical Complex Network Dynamics. The percentage error rates of the non-existent and the existent links detection (R_e^0 / R_e^1 , %) are listed for all combinations of the network structure and dynamics. Each entry denotes the error rates R_e^0 / R_e^1 corresponding to a particular type of dynamics (shown in the column) for a real-world network topology (shown in the row). The result for each combination is obtained with $M = 0.4N$ linear measurements used in each implementation. For some combinations, there exists a global absorption state in which the system state configuration freezes, or a vibration state where the system only switches between a small number of configurations. In such a case, not sufficient number of linearly independent measurements can be obtained to implement compressive sensing. The entries associated with these cases are marked by “* / *”. A detailed description of the real world networks can be found in Supplementary Note 2.	24
4.	Structure Reconstruction Error Rates of 4 Additional Real-World Networks Associated with 11 Types of Complex Network Dynamics. Table legends are the same as for Tab. 3.	25

5.	The Interdependency Reconstruction and Parameter Estimation Performance on 6 Multi-Layer Networks Structured Via 11 Real-World Complex Networks. Four categories of real-world networks are adopted: (1) three social networks [4, 5], (2) three food webs [9, 10], (3) three protein interaction networks [5], and (4) two logic circuits [6]. $10\% \times N_{\max}$ randomly selected nodes in each layer are chosen as bridge nodes, where $N_{\max}$ denote the total number of nodes in the layer with the largest size. The relative error rate ϵ_C of detecting interdependent connections for nonzero terms (NZ, the existent links) and zero terms (Z, the non-existent links) and the parameter estimation error e_W for both evolutionary game dynamics and nonlinear oscillator dynamics are presented in the table. $R_M = 60\%$ is adopted for all calculations.	58
6.	Description of the 9 real-world networks used in our study (N - number of nodes; L - number of edges).	199
7.	Description of the 18 real-world networks used in the paper (N - number of nodes; M - number of edges).	238
8.	Practical controllability of real-world networks studied in Ref. [2], where N_D denotes the number of controllers as determined by the structural controllability theory and M^* is the number of augmented driver nodes needed to make the network practically controllable. The densities of the original driver nodes and with M^* augmented drivers included are $n_D = N_D/N$ and $n_D^* = N_D^* / N$, respectively, where $N_D^* = N_D + M^*$; n_D^* is the new measure of controllability; and E^* denotes the new energy.	239

9. In addition to Table 8, when an additional control signal is added to the middle of each LCC [strategy (I)] so that M_{mid}^* extra control inputs are used, the control energy is E_{mid}^* . For comparison, we use M_{mid}^* randomly chosen extra driver nodes and calculate the required control energy $E_{\text{R-mid}}^*$. We also test another strategy in which an extra control signal is applied to each of the m convergent nodes of all LCCs [strategy (II)], in which M_{end}^* additional control inputs are used. The control energy required is denoted as E_{end}^* , with $E_{\text{R-end}}^*$ being the control energy when the same number of additional controllers are chosen randomly. All energies are result of averaging over 1000 combinations of initial and final states. The control diameter D_C of each network is listed in the last column. (See Supplementary Table 7 for detailed description of the 18 networks.) 240

LIST OF FIGURES

Figure		Page
1.	<p>A Schematic Illustration of SDBM Structure Reconstruction. (a) Connection reconstruction of the red node in a complex network with 20 nodes. The connections of the network is completely unknown, thus the red node could be potentially connected with any other node, as shown by the orange dashed links. After the compressive sensing process [see (c)] and the K-means clustering [see Fig.8] for this node with the dynamical process data of the system, its true connections are detected, marked by the red links. The link weights and the node bias are represented by vector $\mathbf{V}_{N \times 1}$ in (c). (b) All the network connections are recovered by running the process shown in (a) for each node. The Markov blanket of the red node in (a) consists all its nearest neighbors, shown as the nodes with dashed red circles. (c) A schematic illustration of the compressive sensing framework for connection detection, formulated by Eq. (2.9).</p>	9

2. Solution Examples of Compressive Sensing and the Corresponding K-Means Clustering Results. The compressive sensing solutions (green squares) are compared, element-wise, with the true element values of the original connection vectors (dark purple triangles), where the x-axis is the index of nodes with different degree values for different network topologies and dynamics types. The target nodes and the correctly detected existent connections via the K-means are marked as light blue and light purple bars, respectively. The false negative and the false positive estimations are marked as red and gray bars, respectively. For all cases, the amount of linear measurements is $M = 0.4N$. The restriction on Hamming distances is set to be $\Gamma = 0.35$ for all examples, and a small change in Γ would not affect the qualitative results. (a) A node with the smallest degree $k = 2$ in a BA scale-free network of size $N = 100$ and average degree $\langle k \rangle = 4$ with SDBM dynamics. (b) The node of the largest degree $k = 18$ in the same network. (c) A node of degree $k = 8$ in an ER random network of size $N = 100$ and average degree $\langle k \rangle = 4$ with SDBM dynamics. (d) The node with the largest degree $k = 19$ in a BA scale-free network of size $N = 100$ and average degree $\langle k \rangle = 4$ with prisoner's dilemma game (PDG) dynamics. (e) A node of degree $k = 11$ in a real-world social network of size $N = 67$ subject to CP dynamics. (f) A node of degree $k = 10$ in a real-world electrical circuit network of size $N = 122$ with Language model dynamics. In (d-f), the element values of the original connection vector are set to unity since no true weight values are given. For cases (a,b,c,f), there are no false negative detections, i.e., all existent links have been successfully detected. For cases (a,c,e), there are no false positive detections, i.e., no non-existent links are mistaken as existent ones. A detailed description of the dynamical processes is given in Tab. 1. 19

3. Structural Estimation Success Rates and Parameter Estimation Errors Versus the Normalized Data Amount. For SDBM dynamics on (a) ER-random and (b) BA scale-free networks, the success rates of detection of existent links and identification of nonexistent links, $1 - R_e^1$ (green circles) and $1 - R_e^0$ (white circles), respectively, versus the normalized data amount M/N with error bars. The absolute values of the errors in estimating the link weights and the nodal biases for the existent links (purple squares) and non-existent links (white squares) are also shown. Most data points have quite small error bars, indicating the performance stability of our method. High performance structural and parameter estimations are achieved insofar as M/N exceeds about 10%. 20
4. A Schematic Illustration of SDBM Parameter Estimation. (a) The parameter estimation framework for the corresponding SDBM of the same system shown in Fig. 1 (a,b). It is formulated by Eq. (2.11). The connections of the network are already known via the structure reconstruction step, and thus the entries are only obtained from a node's Markov blanket [shown in Fig. 1(b)]. This calculation is implemented for each node in the system. (b) The values of the link weights and the node biases (shown as different colors) for the corresponding SDBM are calculated via the parameter estimation scheme in (a) and the degree guided solution substitution operation (see Materials and Methods). 27

5. Comparison Between the Conditional Probability of the Original System and That Generated Via the Reconstructed SDBM. (a) Conditional probability $P[x_i(t+1) = 1|\mathbf{X}_i^R(t)]$ versus time t for the original system with SDBM dynamics for a BA-scale free network and (b) the reconstructed approximator using the configuration time series of the original system, where $1 \leq i \leq N$ (the y-axis) and $1 \leq t \leq T$ (the x-axis). (c) Time series of the conditional probability $P[x_i(t+1) = 1|\mathbf{X}_i^R(t)]$ for the original system with SIS dynamics for a BA-scale free network and (d) the reconstructed approximator. 29
6. Distributions of the Conditional Probability Estimation Errors for Various Complex Network Dynamics. The distributions of $\delta = |P[x_i(t+1) = 1|\mathbf{X}_i^R(t)] - P^{\text{est}}[x_i(t+1) = 1|\mathbf{X}_i^R(t)]|$ for the dynamical processes in categories I to V from Table 1 for ER random (green bars) and BA scale-free (blue bars) networks, where $P^{\text{est}}[x_i(t+1) = 1|\mathbf{X}_i^R(t)]$ denotes the conditional probability estimated from the corresponding SDBM approximator through the original state configuration times series, $1 \leq i \leq N$, and $1 \leq t \leq T$ 33

7. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using the Self-Generated State Configuration Time Series. (a,d), (b,e), and (c,f): the conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ time series generated from the original state configuration time series (upper panel) and from the SDBM approximator using the state configuration time series generated by itself (lower panel) for SQ-SG, SIS, and Voter dynamics on ER random networks, respectively. (g,h,i) The mean and standard deviations of $p = P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ for each node i for all time steps for SQ-SG (g), SIS (h), and Voter dynamics (i), respectively, on ER random networks, calculated from the original system (red circles) and the approximator (blue squares), where $1 \leq i \leq N$ (the x-axis). 38

8. A Schematic Illustration of the K-Means Algorithm for a 2-Dimensional Data Clustering Task. (a) The data points (solid blue circles) in a 2-D feature space to be clustered. It can be visually distinguished that there are three clusters, thus $K = 3$. The K-means algorithm will assign each data point into one of the three clusters. (b) Randomly guess 3 cluster center locations (aqua, green, and red hollow circles), and each data point finds out which center it's closet to. (c) The 2-D space is divided into three regions by three decision boundaries (black dashed lines), each containing the corresponding data points whose closest center is within this region. The data points currently in the regions with the corresponding aqua, green, and red centers are assigned to the aqua, green, and red clusters, respectively. At this stage, data points may be assigned to a cluster that it visually does not belong to. (d) Each center moves to the centroid of the data points currently assigned to it. (e) The updated cluster assignments of the data points are obtained according to the new center locations (movements are shown by the black arrows). The steps in (c) and (d) are repeated until convergence. (f) The final cluster assignments. Different from this illustration, the compressive sensing solutions of the link weights w_{ij} are 1-dimensional data points, which form two clusters in 1-D space, corresponding to the existent and non-existent links, respectively. (We use the 2-D case to illustrate the K-means algorithm instead of 1-D, due to the representation difficulty in the 1-D case.)

9. An Example of Our Proposed Degree Guided Solution Substitution Operation and Final Equation Group Construction. The target node 1 has only two neighbors, nodes 2 and 3. Without loss of generality, we set $k_3 < k_1 = 2 < k_2$ so that the link weight between nodes 1 and 3, w_{13} , is already obtained. The quantity $P_s^{\text{MB}} = \ln(1/\langle x_1(t+1) \rangle_s - 1)$ ($s = 1, \dots, 4$, as $2^{k_1} = 4$ possible (x_2, x_3) configurations exist in total) denotes the estimated probability for $x_1 = 1$ under configuration s at the next time step. The occurring probabilities of the corresponding (x_2, x_3) configurations are listed as percentages in the left column of each panel. **(a)** All the four (x_2, x_3) configurations are shown by the first two columns of the 4×3 matrix in the equation. **(b)** The contribution of the known w_{13} and the configurations of x_3 as marked in **(a)** are moved to the left-hand side of the equation. **(c)** In order to solve for $(w_{12}, b_1)^T$, the two most precise P_s^{MB} values ($s = 2$ and 4) corresponding to the most frequent (x_2, x_3) configurations from **(b)** are chosen to build a linear 2-equation group. However, the 2×2 matrix on the right-hand side does not have full rank and, hence, this equation group is ill defined **(d)** Configurations $s = 1$ and 2 are selected to construct a 2-equation group with a full rank 2×2 matrix on the right-hand side with relatively precise probability estimations, and this equation group is used to solve $(w_{12}, b_1)^T \dots \dots \dots$ 43

10. Schematic Illustrations of a Multi-Layer, Interdependent Network and the Compressing Sensing Approach of Solving the Interdependent Connection Structure.

(a) A 3-layer interdependent network with 8 nodes in each layer. Nodes i_{11} and i_{12} are the bridge nodes in layer 1; nodes i_{21} , i_{22} , and i_{23} are the bridge nodes in layer 2; and nodes i_{31} , i_{32} , and i_{33} are the bridge nodes in layer 3. The interdependent node pairs are i_{11} and i_{32} , i_{12} and i_{22} , i_{21} and i_{31} , and i_{23} and i_{33} . (b) A schematic illustration of Eq. 3.2. The numbers of rows and columns only have schematic meaning.

The number of time instances selected (the number of rows of $\hat{\mathbf{U}}_i$ and $\mathbf{W}'_i$) is usually much smaller than the length of vector in a typical compressive sensing setting. The colored parts of matrix $\mathbf{W}'_i$ and vector $\mathbf{A}'_i$ corresponds to the inner-layer connections of layer m , while the black-and-white parts corresponds to the interdependent connections between layer m and the bridge nodes in all other layers. Using (a) as an example, if $m = 2$ and $i = i_{21}$, then the black-and-white components in (b) are corresponding to the possible interdependencies between node $i = i_{21}$ and nodes $q = i_{11}, i_{12}, i_{31}, i_{32}$, and i_{33} 50

11. Bridge Node Identification and Interdependencies Reconstruction. (a1,a2,a3) The standard deviations of the compressive sensing solution $\mathbf{A}_i$ for each node i within layer 1 (a1), 2 (a2), layer 3 (a3), respectively. The interdependent multiplex structure adopted in this example contains 3 layers, each of which is built in ER-random topology with $N_m = 50$ nodes ($m = 1, 2$, and 3) and average degree $\langle k \rangle = 4$. The x-axis denotes the node index, and the y-axis denotes the standard deviation σ for each corresponding node. The threshold $\sigma_c = 10^{-3}$ is used in our calculation, and the light purple bars marks identified the bridge nodes (BN). (b1,b2,b3) The colormap of the augmented connection vector $\mathbf{A}'_i$ for each node i in layer 1 (a1), 2 (a2), and 3 (a3). The x-axis show the index i , and the entry in the j 's (or the q 's) row and the i 's column stands for the connection between node i and j (or q), where j ($1 \leq j \leq N_m$) denotes an inner-layer node, and q ($q > N_m$) denotes an bridge node in another layer (below the dashed line). The dark blue back ground stands for the non-existent connections, while the all other colored entries implies an existent connection. Only the bridge nodes have nonzero entries $\eta_{iq}c_{iq}$ ($q > N_m$) in the augmented part, while in the non-augmented part, a nonzero entry of an ordinary node has the value $a_{ij} = 1$ (dark red), while that of a bridge node has the value $(1 - \eta_{iq})c_{iq} < 1$ (green, yellow, orange, or light red). 54

12. The Interdependent Connection Structure Recovery and the Payoff Exchange Rate Estimation Performance Under Evolutionary Game Dynamics. (a,b,c) The relative interdependent connection reconstruction error rate ϵ_C for the nonzero terms (“NZ”, i.e., the existent links, shown in purple) and the zero terms (“Z”, i.e., the non-existent links, shown in green) versus the normalized number of measurements $R_M = l/N_m$ under noise level $\mu_N = 0$ (triangles pointing up), 10^{-4} (circles), 10^{-3} (squares), and 10^{-2} (triangles pointing left). (d,e,f) The absolute estimation error $e_W = |\eta^{\text{estimation}} - \eta^{\text{true}}|$ of the interdependent payoff exchange rate η_{iq} averaged over all the interdependent node pairs versus R_M under the four different noise levels as in (a,b,c). The results are obtained on three 2-layer multiplex networks: (a,d) both layers have ER-random topology; (b,e) both layers have BA scale-free topology; (c,f) one layer has ER-random topology, while another one has BA scale-free topology. All layers in all cases are in size $N_m = 100$ with average degree $\langle k \rangle = 4$. 10% randomly selected nodes in each layer are bridge nodes, each of which owns an interdependent partner in another layer, and there are accordingly 10 interdependent links in each multiplex system. Each data point is averaged over 10 network realizations, each with 10 compressive sensing implementations. 56

13. Structural and Practical Controllability Measures in Directed Networks. Structural controllability measure n_D versus directional edge probability P_b for (a) ER random networks and (b) BA scale-free networks of size $N = 1000$ and three values of the average degree ($\langle k \rangle = 4, 6$, and 8). The dash-dotted lines represent the results obtained by the cavity method [2, 2], and the squares, triangles, and circles are the simulation results from the maximum matching algorithm [2]. (c, d) Measure of practical controllability $P(\bar{C}_W)$ for ER random and BA scale-free networks of size $N = 100$, respectively, where $P(\bar{C}_W)$ is the probability that the condition number of the Gramian matrix is less than some physically reasonable threshold value versus P_b . Comparing a with c, or b with d, we observe the striking phenomenon that, in the parameter regime where the number of driver nodes is minimized so that the corresponding networks are deemed to be most structurally controllable, they are practically uncontrollable. The phenomenon persists regardless of the network size and type. All nodes are self-loop free. The qualitative behavior is robust against the value of $\bar{C}_W$ 67
14. Distributions of Control Energy for Practically Controllable Networks. (a, c) Energy distributions under different values of the threshold condition number $\bar{C}_W$ for $P_b = 0.1$ and $t_f = 1$. $\langle k \rangle = 6$ for random networks (a) and $\langle k \rangle = 8$ for scale-free networks (c). (b, c) Energy distributions for different values of the average degree $\langle k \rangle$ and the directional connection probability P_b under $\bar{C}_W = 10^{12}$ and $t_f = 1$ for ER random and BA scale-free networks, respectively. In all cases, we observe an algebraic (power-law) scaling behavior. 69

15. Relationship Among 1D Chain Energy, the Smallest Eigenvalue of H -Matrix, and Network Control Energy. (a) For an 1D chain network of length l , energy E_l and $\lambda_{H_l}^{-1}$ versus l , providing support for the analytic result equation (4.4). (b) Correlation between $\langle E \rangle$, the average of control energy for networks with the same LCC length, and E_L , the energy of a LCC with length $D_C = L$ ($L = 3, 4, 5, 6$, and 7 for ER and $L = 3, 4, 5$, and 6 for BA networks), calculated from ensembles of 10000 networks. 71

16. Schematic Illustration of Various Concepts to Characterize and Understand the Practical Controllability of a Network. (a) Control-signal paths (CSPs) of a random network obtained from maximum matching in structural controllability theory, where a control signal enters a CSP via the corresponding driver node (yellow), the starting node of the path, and goes through each matched node (blue) along the path. In this example, the network has $n_D = 10$ CSPs of different lengths. Non-path links, links that are irrelevant to matching, are displayed in green. (b) All possible LCCs in the network. Typically there are multiple LCCs of the same length. In this example, the length of the LCCs is 4, which is defined as the control diameter of the network. Two LCCs sharing no common nodes are marked by red nodes and solid red arrows. Links belonging to other LCCs are marked by red dashed arrows. CSPs are denoted using letters a to j from the left to the right. Each node is specified using its path number and its position along the path sequentially from top to bottom. For example, node $e1$ is the driver node of path e , and node $h2$ is the node right after the driver node on path h . The two LCCs with solid arrows are listed in bold. Eight LCCs in the network converge to only three end-nodes, $e6$, $f5$, and $f6$ (marked by red dashed circles), leading to LCC degeneracy $m = 3$. The control energy is determined by any randomly chosen m LCCs among all existent ones. . . 81

17. Cascade Parallel R-C Circuit and Its Corresponding Network Presentation. (a) A cascade parallel R-C circuit with $L = 7$ resistors ($R_1, R_2, \dots, R_L$, each of resistance 1Ω) and 7 capacitors ($C_1, C_2, \dots, C_L$, each of capacitance $1F$). External voltage input $u(t)$ is applied onto the left side of the circuit, and the voltage of capacitor C_i is $u_i(t)$ ($1 \leq i \leq L$). (b) Network representation of the circuit in a as a bidirectional 1D chain network of seven nodes, where the external voltage input $u(t)$ is injected into node 1 (yellow driver node, the controller). The dynamical state of node i is described by the voltage on its capacitor, $u_i(t)$. Links (blue) between nodes are bidirectional and have uniform weight 1 in either direction. Each node has a self-link (red) of weight -2 , except the ending node (node 7) whose self-link has weight -1 . (c) The circuit network in b with an extra external current input $i_e(t)$ into the capacitor C_3 , where i_3 and i_4 denote the currents through resistors R_3 and R_4 , respectively. In the absence of the extra current input, $i_3(t) - i_4(t)$ is the current through the branch of C_3 . (d) The extra external current input $i_e(t)$ serves as a redundant control input injected into node 3 of the network in b. Now there are two driver nodes (yellow) in the network, nodes 1 and 3. 82
18. Control Energy and Optimization for 1D Chain and Cascade Parallel R-C Circuit. (a) Energy required for controlling a unidirectional chain (red) and the corresponding circuit (blue) as well as the dissipated energy of the circuit calculated from equation (4.5) versus the chain length L . (b) Control and dissipated energies in the presence of a redundant control signal to node i ($i > 1$), which breaks the chain into two subchains of lengths i and $L - i$, respectively. 83

19. Effects of Redundant Control Inputs. (a) For control diameter $D_C = 2$, distribution of the energy ratio E_x/E under optimization strategies (I) (Mid, red circles) and (II) (End, blue squares), where E_x and E are the required energies with and without redundant control, respectively. Results from two randomized optimization strategies are marked by R-Mid (black triangles) and R-End (gray diamonds), corresponding to strategies (I) and (II), respectively. The values of E_x/E are collected from practically controllable networks from an ensemble of 10000 ER random networks ($\langle k \rangle = 6$, $P_b = 0.1$). For each network, if strategy (I) [or (II)] requires r redundant controls, r additional random control inputs are applied to the system 10 times to average out the random fluctuations. Panels (b-d) show the E_x/E distributions for networks with control diameter $D_C = 3, 4$, and 5 , respectively. 84

20. Augmented Control Inputs and Energy Optimization. (a) Densities of the original driver nodes n_D (dark purple) and of the augmented controls n_D^* (light purple). (b) Normalized energy reduction $\Delta E_{\text{mid}}^*/E^* = (E^* - E_{\text{mid}}^*)/E^*$ (light blue) when an additional control signal is added to the middle of each LCC [strategy (I)] and normalized energy reduction $\Delta E_{\text{R-mid}}^*/E^* = (E^* - E_{\text{R-mid}}^*)/E^*$ (dark blue) when the same number of control signals are randomly added into each network. (c) Normalized energy reduction $\Delta E_{\text{end}}^*/E^* = (E^* - E_{\text{end}}^*)/E^*$ (light green) when an additional control signal is added to the end node of each LCC [strategy (II)] and normalized energy reduction $\Delta E_{\text{R-end}}^*/E^* = (E^* - E_{\text{R-end}}^*)/E^*$ (dark green) when the same number of control signals are randomly added into each network. All colored bars (light/dark purple, light/dark blue, and light/dark green) start from 0, and the darkly colored bars overlap with the corresponding portions of the lightly colored bars. For the bars with only dark colors, the lightly colored bars are in the same height but covered by the dark ones in front of them. In majority cases, our optimization strategies work better than random control signal augmentation, and in 7 out of 17 networks, the energy reduction under our strategies is one or several order of magnitude greater than under random augmentation. 85

21. The Number of Extreme Events Versus Velocity and Mobility. For a network of $N = 1200$ nodes in a square domain of size $L = 10$ (arbitrary unit), **a**, The number of extreme events n_{ex} versus the agent velocity v for $W = 50N, 80N$, and $100N$; **b**, n_{ex} of extreme events versus velocity v when nodes execute random walk (R) or deterministic motion (D) within square, circular, and stadium-shaped domains. The areas of the circle and the stadium are chosen to be equal to the area of the square $L \times L$. **c**, Robustness of control with respect to heterogeneity in the nodal communication range: n_{ex} versus v for $\beta = -1, 0$, and 1 , where β is a parameter characterizing the distribution of nodal communication range. **d** and **e**, robustness of control with respect to packet-transportation protocol: n_{ex} versus velocity v when packet generation and annihilation are taken into account for two situations where **d** the total number of packets is fixed at W and **e** the number of newly generated packets at each time step is fixed to be W/N . **f**, *Generalization of mobility*: n_{ex} versus mobility, defined as the probability that a node moves with velocity v , for a number of v values. All simulation results are obtained using 100 realizations in $T = 1000$ time steps. The error bars in n_{ex} from different realizations have relatively small and similar magnitude in (**a-e**), and it monotonously increases with velocity v , as shown in panel **a** [the error bars in (**b-e**) are omitted for clear visualization]. Here, the definition of an extreme event on a node is that its number of packets is at least four standard-deviations above the average. 88

22. Geometrical Setting and the Theory Result. **a**, Illustration of various geometrical quantities used in the analysis: communication radius a , distance b from the middle of the square to the target node, distance b_n from the middle of the square to a neighbor of the target node, the width Δb_n of the bar region, and the cutoff region of the communication circle. **b**, Results of Eq. 5.2 for $db = 0.01$ and $\Delta b_n = 0.002$ (the main features do not depend on the specific values of db and Δb_n). 90
23. Boundary Effect. **a1, b1, c1**, Behavior of $\langle w(t) \rangle_b$ versus b from direct simulation (blue squares) and predicted by Eq. (5.8) (solid black lines) for $v = 0, 0.5$ and 3.0 , respectively. Note that $v \approx 0.5$ is the optimal value of the velocity that minimizes n_{ex} in Fig. 21, and $v = 3.0$ represents a relatively high velocity value for the system of size $L = 10$ and communication radius $a = 1$. For **a1**, we have $\eta(v = 0) = 0$. Since Eq. (5.8) indicates a smooth dependence of $\langle w(t) \rangle_b$ on b and since a detailed interdependence between $\eta(v)$ and v is unknown, we find $\eta(v) = 15$ best fits the case in **b1**. For **c1**, we have $\eta(v = 3.0) \rightarrow \eta(v \rightarrow \infty) = 50$. The red dashed line indicates the position of the peak in the curve of $\langle w(t) \rangle_b$ versus b . **a2, b2, c2**, Positions of all extreme events occurred in square domain for $v = 0, 0.5$, and 3.0 , respectively. **a3, b3, c3**, Positions of all extreme events occurred in circular domain for $v = 0, 0.5$, and 3.0 , respectively. **a4, b4, c4**, Positions of all extreme events occurred in stadium domain for $v = 0, 0.5$, and 3.0 , respectively. All results are obtained from 20 independent statistical realizations. 93

Figure	Page
24. Effect of Long Range Links. a , effect of periodic boundary condition: n_{ex} versus v when 0% (green circles, no PBC), 30% (aqua triangles), 50% (pink triangles), and 100% (purple triangles, complete PBC) randomly selected cross-boundary links are applied. b , <i>Effect of communication wormhole on control</i> : n_{ex} versus v when a wormhole link is applied to any pair of nodes closer than $d_{HW} = 13.0, 12.5, 12.3$, and 12.0 for $p_{HW} = 0.4\%, 2\%, 3\%$, and 5% , respectively.	97
25. Number of Extreme Events in Intermittently Mobile Networks. a and b , For a mobile network with hard-wall boundary conditions, n_{ex} versus v and for a number of fixed values of the frozen time ΔT and versus ΔT for three values of v . c and d , Corresponding results for periodic boundary conditions. e and f , For a classic ER-network under intermittent random rewiring (a kind of generalized mobility), n_{ex} versus the rewiring probability P_R (for a number of fixed values of ΔT) and versus ΔT (for $P_R = 0.5, 0.8$, and 1.0), respectively.	99
26. Degree Bias and Scaling. For a mobile network in a square domain with hard-wall boundary conditions, a , number n_{ex} of extreme events occurring on small (red circles) and large (blue squares) degree nodes versus v , and b , n_{ex} versus the total number of packets W for various velocity values.	101
27. Linear Relation Between $\langle W(T - 1) \rangle$ and $k(T)$. a-c , Linear relation between $\langle w(t - 1) \rangle$ and $k(t)$ for mobile networks with hard-wall and periodic boundary conditions, and for ER-random networks, respectively. d , Dependence of $\eta(v)$ on v for mobile networks with hard-wall and periodic boundary conditions, and $\eta(P_R)$ versus P_R for ER-random networks (inlets). All results are averaged over 20 statistical realizations.	103

28. PDF of Flow $P(W)$ in Simulation and in Different Theory Assumptions. **a**, For the classic ER random networks with rewiring probability $P_R = 1$, PDF of flow $P(w)$ obtained from direct simulation (red squares) and self-consistent equation (solid blue line). **b**, PDF of flow $P(w)$ from direct simulation (red squares), from theory without taking into account spatial degree correlation (black circles), and from theory with degree-degree correlation (solid blue line). 107
29. Schematic Illustration of a Multilayer, Interdependent Network Subject to Internal (Interlayer) Resource Competitions. Layers 1 (G_1) and 2 (G_2) are two different subsets of Layer 0 (G , or the global network), which is solely used for model generate and does not participate in the transportation dynamics. In this schematic example, $N = 10$ (number of node in Layer 0), $M = 2$ (the number of interdependent layers), and $P_m = 0.5$ ($m = 1$ and 2). Accordingly, $N_m = N \cdot P_m = 5$, and correspondingly, the node degrees in Layers 1 and 2 are 0.5 time of their degrees in Layer 0 (fluctuation exists due to randomness). The combination of Layers 1 and 2 forms a multilayer interdependent network. Nodes 5 and 6 exist in both Layers 1 and 2, and thus, the two layers would compete for resources within each of these two overlapped nodes. Nodes 4 and 9 exist in neither of the two layers. Each of all the other nodes exist in only one of the two layers. (See the Methods section for model details.) 114

30. Extreme Event Occurrence and Packet Arrival Rates. (a) Extreme event occurrence rate R_{EE} and (b) packet arrival rate R_A versus the number of layers, M , for systems with $P_m = 0.25$ (circles), 0.5 (squares), 0.75 (diamonds), and 1 (triangles), respectively. (c) R_{EE} and (d) R_A versus P_m for $M = 3$ (diamonds), 5 (circles), and 10 (squares), respectively. The results are obtained through 20 simulation realizations for each of the 10 network realizations. Each network has the average degree $\langle k \rangle = 4$ and size $N = 1000$. Each realization runs for 500 time steps. The quantities R_{EE} and R_A are averaged over the last 300 steps (typically, the system evolves into equilibrium within 200 time steps). 115
31. System's Time Evolution Towards a Catastrophic State. For a multilayer network of $N = 1000$ nodes, $M = 3$ layers, and $P_m = 1$, (a) the transition between a free (blue) and a catastrophic states (red) in time for all nodes. (b) The ratio of node i 's load f_i to its capacity C_i versus time for all nodes. (c) Time traces of the total number of packets in the layer (denoted as W , thick solid line), of the number of "stuck" packets due to the surrounding extreme events (W_{stop} , triangles), and of the number of movable packets (W_{jump} , squares). Among the W_{jump} movable packets, the number of those that effectively execute random walks (W_{rand} , red dashed line) and the number of the remaining packets moving along their respective shortest paths (W_{short} , blue dot line) are also plotted versus time. 118

32. Numerically Obtained and Theoretical Predicted Probabilities of Extreme Events. For $M = 1$ (circles), 3 (squares), 5 (diamonds), and 10 (triangles), (a) numerically obtained probability $q_{EE}(k)$ for an extreme event to occur on a node of degree k and (b) theoretical prediction based on the concept of betweenness. (c) For $P_m = 1$, the probability of extreme events versus M , where the degree distribution function $P(k)$ used in the calculation of q_{EE} is $P(k) = k^{-3} / \sum_{k=k_{\min}}^{k_{\max}} k^{-3}$ for $k_{\min} = 2$ and $k_{\max} = 75$ (typical values from network realizations). In fact, small changes in the values of $k_{\min}$ and $k_{\max}$ do not affect the main features of the results. 121
33. Probability of Extreme Events Calculated From Our Theory of Effective Betweenness. (a-d) The probability $q_{EE}(k)$ for an extreme event to occur on a node of degree k obtained from simulation (circles), the theory of betweenness (diamonds), and the theory of effective betweenness (squares) for $M = 1, 2, 3$, and 5, respectively. In (e), the extreme-event probability versus M is shown, obtained from simulation (circles), the theory of betweenness (diamonds), and the theory of effective betweenness (squares), for $M = 10$ and $P_m = 1$. The degree distribution function $P(k)$ used to calculate q_{EE} is obtained using 10 network realizations. The betweenness and effective betweenness for each node of degree value k are calculated from the corresponding network topology and averaged over all such nodes among the 10 network realizations. 122

34. Occurrence Rate of Extreme Events and Packet Arrival Rate Under Control. In the control parameter space, n_{top} denotes the number of top-degree hubs whose capacities are augmented, and r_1 is the ratio of the enhanced capacity to the original capacity. The extreme event occurrence rate R_{EE} and the packet arrival rate R_A in the parameter space are shown, respectively in (a) and (b). The simulation parameters are the same as in Fig. 30. 126
35. Number of Packets L_d Steps Away From Their Destinations. For a single layer network of size $N = 1000$, the number of packets in an equilibrium state that are L_d steps away from their respective destinations from theory (triangles) and simulation (circles). The total number of packets that the system can accommodate in the equilibrium state is about $W = 558$, which includes the newly generated packets within the current time step (diamonds) and all the remaining old packets (squares). 131
36. Normalized Effective Betweenness and Free Flux Distribution Function. For a single layer network of size $N = 1000$, (a) the normalized effective betweenness of a node of degree k , p_E , versus k , from simulation (squares) and theory (line). The simulation results are obtained from 20 network realizations. (b) The free flux distribution function $P(w)$ for a node of degree 75 versus w obtained from simulation (squares) and theory (line), where the distribution function is meaningful only when the system is set to be free of extreme events, i.e., each node is set to have infinite capacity so that packets can travel freely via the shortest path towards their destinations. This way the theoretical probability for the node to be visited can be calculated. 133

37. (Color online.) Death rate S_d (red circles) and frequencies of cooperation ρ_C (blue squares) and defection ρ_D (green triangles) versus the tolerance parameter α for six combinations of three values of the control parameter β (-1 , 0 , and 1) and two values of T_0 (20 and 100) for $b = 1.5$ 141
38. (Color online.) For four combinations of the parameter values of α and T_0 , S_d (red circles), ρ_C (blue squares), and ρ_D (green triangles) versus the control parameter β for $b = 1.5$ 142
39. (Color online.) Time tolerance $T(k)$ (left) and its derivative $dT(k)/d\beta$ (right) versus control parameter β for $T_0 = 20$ 144
40. Time Series of Attack Frequency $w(T)$ for All IP Addresses. (a) Spatiotemporal representation of attack frequencies $w(t)$ of all IP addresses on a logarithmic scale, where the x -axis and y -axis are, respectively, time t and IP address index from 1 to 491 (top to bottom). The IP region 1 - 246 is denoted as the aqua background, where each of the four IP regions (19 - 31 , 35 - 47 , 50 - 130 , and 131 - 191) exhibits a particular attack pattern that is overlayed on the background. The IP region 247 - 363 possesses an attack pattern that is overlayed on the dark-blue background lying under the entire IP-space. (b) Three-dimensional presentation of the attack frequency $w(t)$ on a logarithmic scale. The “walls” of unity height are not visible on the logarithmic scale, but higher “walls” are visible in the IP region 364 - 491 [corresponding to the vertical light-blue lines on the bottom dark-blue background in (a)], which actually occur in the entire IP-space but are mixed with other patterns. The time resolution is $\Delta t = 1000$ seconds. 151

Figure	Page
41. Time Series of Attack Frequency with Higher Time Resolution. Time range: $t = 86200$ to 86600 . Time resolution: $\Delta t = 10$ seconds. (a,b) Two- and three-dimensional representations of the attack frequency $w(t)$ for the entire IP-space on a linear scale, respectively.	152
42. Deterministic Attack Pattern on Different Time Scales. For time resolution $\Delta t = 10$ seconds and IP region 1-250, (a) deterministic attack pattern in a relatively large time scale: the sloped light-green lines in the IP region 19-31, where the time range is $t = 38000$ to 42000 , (b) enlarged section between the two vertical pink lines in (a) in the time range from $t = 39500$ to 40000 , which provides a time-scale comparison between two types of deterministic attack patterns.	153
43. Deterministic Attack Pattern of “Walls”. For time resolution $\Delta t = 10^{-3}$ second, (a) one type of “wall” attacks over the entire IP space, with each IP receiving one attack. The time delays for the attacks to reach the targeted IP addresses exhibit three distinct values, separating the dots in this panel into three lines. (b) A different type of “wall” attack with different delay, and (c) enlarged section in the small red square in (a). Five consecutive attacks on each IP occurred before the next IP is attacked.	155
44. Flux-Fluctuation Relation. For time resolution $\Delta t = 10$ seconds, $\langle w \rangle$ - σ relations (a) for each IP in the IP space, (b) for IP regions 247-257 and 364-491, and for each IP in the dark-blue background with walls (red circles) and without walls (blue squares), (c) for IP region 258-363 with (red circles) and without (blue squares) walls. The blue dashed lines in (b) and (c) have the slope $1/2$	156

Figure	Page
45. Attack Frequency Deviation Within An IP Region Versus the Average At Each Time Unit for (a) the the dark-blue background and (b) IP region 258-363. The time resolution is $\Delta t = 10$ seconds.	158
46. Correlation Coefficients. (a) Correlation coefficient matrix associated with the IP space, (b) total number of attacks on an IP address, M (blue), the average time interval between consecutive attacks on an IP, $\langle \tau \rangle$ (red), and the cluster size to which each IP belongs (green). The clustering threshold is set to be 0.7 (somewhat arbitrary). (c) Time series of attack frequency for IP 155 (red), 160 (blue), and 165 (green). All three IP addresses belong to the region of the aqua background without any other overlayed attack patterns. (d) Time series of attack frequency for IP 405 (red), 409 (blue), and 420 (green), which belong to the region of the dark-blue background without any other overlayed attack patterns. One abrupt peak occurs at IP 409, and another at IP 420. The time resolution is $\Delta t = 10000$ seconds for all four panels.	160
47. Markov State Transition Probability Matrix (MSTPMs). For time resolution $\Delta t = 10$ seconds, MSTPMs obtained from two randomly selected IP addresses within the corresponding IP region. See the legend of Fig. 45(a) for the eight IP regions under different attack patterns: IP addresses (a1,a2) 10 and 33, (b1,b2) 20 and 30, (c1,c2) 40 and 45, (d1,d2) 60 and 120, (e1,e2) 140 and 190, (f1,f2) 200 and 240, (g1,g2) 260 and 360, and (h1,h2) 420 and 490.	162
48. Correlation Coefficient Matrix Associated with the MSTPMs for (a) $\Delta t = 10$ seconds and (b) $\Delta t = 100$ seconds.	165

49. Predictability Calculated From Coarse-Grained Time Series. (a,b) Predictability upper bound $\Pi^{\max}$ for all 491 IP addresses for various section lengths under time resolutions $\Delta t = 100$ seconds and 1000 seconds, respectively, where $\Pi^{\max}$ values are averaged over all sections. (c) Average $\Pi^{\max}$ versus section length h for $\Delta t = 100$ seconds (red circles) and 1000 seconds (blue squares). (d) Average $\Pi^{\max}$ of the deterministic (open symbols) and stochastic (closed) attacks versus the section length h for $\Delta t = 100$ seconds (red circles) and 1000 seconds (blue squares). In the figure, “D” denotes deterministic and “R” stands for random or stochastic. See text for explanation on why, in the case of $\Delta t = 100$, stochastic attacks can have a higher predictability upper bound than deterministic attacks. 167
50. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for SDBM Dynamics. The conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ time series generated by the original system from the original state configuration time series (a,c) and by the SDBM approximator from the state configuration time series produced by itself (b,d) for ER random (a,b) and BA scale-free (c,d) topology. (e,f) The mean and standard deviations of $p = P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ for each node i for all time steps for ER random (e) and BA scale-free (f) networks obtained from the original system (red circles) and the approximator (blue squares), where $1 \leq i \leq N$ (the x-axis). 200
51. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Ising Glauber (IS) Dynamics. Legends are identical to that in Fig. 50. 201

Figure	Page
52. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Self-Questioning Snowdrift Game (SQ-SG) Dynamics. Legends are identical to that in Fig. 50.	201
53. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Self-Questioning Prisoner's Dilemma Game (SQ-PDG) Dynamics. Legends are identical to that in Fig. 50.	202
54. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Minority Game (MG) Dynamics. Legends are identical to that in Fig. 50.	202
55. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Voter (VT) Dynamics. Legends are identical to that in Fig. 50.	203
56. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Majority Vote (MV) Dynamics. Legends are identical to that in Fig. 50.	203
57. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Link-Update Voter (LV) Dynamics. Legends are identical to that in Fig. 50.	204
58. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Language Model (LM). Legends are identical to that in Fig. 50.	204

Figure	Page
59. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Kirman (KM) Dynamics. Legends are identical to that in Fig. 50.	205
60. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Contact Process (CP) Model. Legends are identical to that in Fig. 50.	205
61. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Susceptible-Infected-Susceptible (SIS) Disease-Spread Model. Legends are identical to that in Fig. 50.	206
62. Condition Number C_W Versus Control Error e_X for Random and Scale-Free Networks. Network size is $N = 100$ for (a-d) and 200 for (e-h), average degree is $\langle k \rangle = 6$ for ER random networks [(a),(b),(e), and (f)] and 8 for BA scale-free networks [(c), (d), (g), and (h)]. Directional link probability between any pair of nodes is $P_b = 0.1$. Panels (a),(c),(e), and (g) show the scaling relation between the condition number C_W and control error e_X . Panels (b), (d), (f), and (h) show the fraction R_{CW} of the networks with a certain C_W number. The scaling relation holds within some C_W - e_X region with boundaries specified as the black dashed lines. The e_X values are not physically meaningful outside the boundaries that are defined according to the precision limit of computation. The thresholds of C_W and e_X used in the computations are 10^{12} and 10^{-4} , respectively, which are indicated as the blue dashed lines. The threshold values are chosen to lie within the physical boundaries so that the calculations for all C_W values are meaningful.	217

63. Practical Controllability Measures for Directed Networks. Measure of the practical controllability, $P(\bar{C}_W)$, for (a) ER random networks and (b) BA scale-free networks of size $N = 200$ 218
64. Distributions of Control Energy for Practically Controllable Networks Under Different Network Sizes. Panels (a) and (b) are for random networks with $\langle k \rangle = 6$ and scale-free networks with $\langle k \rangle = 8$, respectively, for two values of the network size ($N = 100$ and $N = 200$), where we set $\bar{C}_W = 10^{12}$ and $t_f = 1$. In all cases, we observe an algebraic (power-law) scaling behavior. 218
65. Comparison Between Energies Required to Control a Unidirectional and a Bidirectional 1D Chains: (a) energies required to control a unidirectional chain E_{uni} (red) and a bidirectional one E_{bi} (blue) versus chain length L , and (b) the relative energy difference $|E_{\text{uni}} - E_{\text{bi}}|/E_{\text{uni}}$ versus chain length L 222
66. Correlation Between Network Control Energy and the Smallest Eigenvalue of H -Matrix. Network size is $N = 100$, directional link probability between any pair of nodes is $P_b = 0.1$, and average degree is (a) $\langle k \rangle = 6$ for ER random networks and (b) $\langle k \rangle = 8$ for BA scale-free networks. 225
67. Probability Distribution of Control Diameter and Its Degeneracy. (a) Distribution of D_C (blue) and the probability that an LCC of length D_C appears in the random network ensemble versus D_C . (b) Probability distribution of LCC-degeneracy m for networks with $D_C > 2$ 228

68. Distribution of control energy in a double-chain interaction models. (a) Two chains of $D_C = 5$ (red) and $D_C = 6$ (black), where the upper panel is a schematic illustration of two LCCs of identical length $D_C = 5$ interacting with each other via some random links (green) between them. (b) Two chains with their lengths randomly chosen from 3 – 6, where the upper panel shows the case of two interacting chains of length 5 and 3. The longer chain (red) plays the role of LCC, while the shorter chain is a non-LCC (orange). 231
69. Controlling and Optimizing a Cascade Parallel R-C Circuit and the Corresponding Network Presentation. (a) A cascade parallel R-C circuit with 3 resistors (R_1 , R_2 , and R_3 , each of resistance 1Ω) and 3 capacitors (C_1 , C_2 , and C_3 , each of capacitance $1F$), where $u(t)$ is the external input voltage, $u_1(t)$, $u_2(t)$, and $u_3(t)$ are the voltages on the capacitors C_1 , C_2 , and C_3 , respectively, $i_1(t)$, $i_2(t)$, and $i_3(t)$ are the currents through the resistors R_1 , R_2 , and R_3 , respectively. (b) Network representation of the circuit in (a). (c) Circuit with an extra external current input $i_e(t)$ into the capacitor C_2 . (d) The extra external current input $i_e(t)$ serves as a redundant control input injected into node 2 of the network in (b). There are two driver nodes (yellow) in the network: 1 and 3. 234

1 . INTRODUCTION

1.1. Structure and Dynamics Estimation for Complex Networks

Revealing the underlying structure and dynamics of complex networked systems from observed data is of fundamental importance to science, engineering, and society. Successful recovery is usually achieved with the aid of certain specific prior information about the particular mathematical rules governing the system dynamics. Is it possible to develop a universal framework capable of reverse engineering different types of dynamical processes on complex networks, regardless of their details? In Chapter 2, I articulate a Markov network based model, the sparse dynamical Boltzmann machine (SDBM), as a universal network structural estimator and dynamics approximator. The SDBM can establish its topology according to that of the original system and is capable of simulating the original dynamical process by reproducing the time series data. Given a networked dynamical system, it is possible to obtain its SDBM equivalent with identical topology and dynamical behaviors as the original system. I develop a fully automated method based on techniques including compressive sensing and K-means algorithm to find the SDBM. My method is completely data driven in that it requires neither the original system equations nor any subjective parameter choice. I demonstrate, for a large variety of representative dynamical processes on model and real-world complex networks, that the equivalent SDBM can recover the network structure of the original system and predicts its short-term or even long-term dynamical behavior with high precision.

Real-world phenomena often indicate that the behaviors of complex systems of similar or different natures can be coupled in certain ways, and the interdependencies between those system usually play crucial roles in determining their dynamical behavior both microscopically and macroscopically. Identifying the interdependent interactions between different complex networked systems thus becomes a critical issue in science and engineering. In Chapter 3, I propose a two-stage com-

pressive sensing based method that recovers the connection topology and the related parameter values of the interdependent components of multi-layer complex networked infrastructures with high precision, given only the dynamical process time series. The performance effectiveness of the method is demonstrated by various types of modeled and real-world complex networks with both discrete and continuous dynamical processes taking places on them, and high robustness against noise is simultaneously achieved in all cases. Systems with large sizes that hinders the previous compressive sensing based approaches may also become applicable due to the layer-wise reconstruction scheme within our method.

1.2. The Paradox of Controlling Complex Networks

One of the most challenging problems in complex dynamical systems is to control complex networks. In previous frameworks based on the structural or the exact controllability theories, the ability to steer a complex network toward any desired state is measured by the minimum number of required driver nodes. However, if I implement actual control by imposing input signals on the minimum set of driver nodes as determined, e.g., by the structural controllability theory, an unexpected phenomenon arises: the energy required to approach a target state with reasonable precision is often unbearably large, precluding us from achieving actual control, i.e., the designated state can not be reached in effect, especially for networks with a small number of drivers. In particular, the energy of controlling a set of networks with similar structural properties follows a fat-tail distribution, indicating the existence of networks with practically divergent energy. In Chapter 4, I aim to reconcile the paradox of controlling complex networks: optimal structural controllability versus unrealistic energy required for control. I identify fundamental structural “short boards” in complex networks that play a dominant role in the enormous energy, and offer a theoretical interpretation for the fat-tail energy distribution and simple strategies to significantly reduce the energy by imposing slightly

augmented set of input signals on properly chosen nodes. Our findings indicate that, although full control can be guaranteed by the prevailing structural controllability theory, it is necessary to balance the number of driver nodes and the control energy to achieve actual control, and our results provide a framework to address this outstanding issue.

1.3. Extreme Events and Cascading Failure on Complex Networks

Extreme events, a type of collective behavior in complex networked dynamical systems, often can have catastrophic consequences. To develop effective strategies to control extreme events is of fundamental importance and practical interest. In Chapter 5, utilizing transportation dynamics on complex networks as a prototypical setting, I find that making the network “mobile” can effectively suppress extreme events. A striking, resonance-like phenomenon is uncovered, where an optimal degree of mobility exists for which the probability of extreme events is minimized. I derive an analytic theory to understand the mechanism of control at a detailed and quantitative level, and validate the theory numerically. Implications of our finding to current areas such as cybersecurity are discussed.

Multilayer, interdependent complex networks are ubiquitous in modern infrastructures. The resilience of such networks is of considerable interest. Extreme events, a type of collective dynamical behaviors, can compromise the network resilience dramatically. In Chapter 6, utilizing transportation as a prototypical type of dynamical process to model, e.g., the interplay among public transportation, power, information, or telecommunication service facilities, I investigate the emergence of extreme events in interdependent networks. I introduce an inter-layer traffic resource competing mechanism to take into account the limited capacity associated with distinct network layers. A striking finding is that, when the number of network layers and/or the overlap among the layers are increased, extreme events can emerge in a cascading manner on a global scale, even if the capacity

is sufficiently large so that extreme events on any of the same number of independent layers can be ruled out. Asymptotically, there are two stable absorption states: a state free of extreme events and a state of extreme events, and the transition between them is abrupt in both time and parameter domains. Our results indicate that internal interactions in the multiplex system can yield qualitatively distinct phenomena that do not occur for independent network layers. An implication is that, public resource competitions among different service providers can lead to a higher resource requirement than naively expected. I derive an analytical theory to understand the emergence of global-scale extreme events based on the concept of effective betweenness. I also articulate a cost-effective control scheme through increasing the capacity of very few hubs to suppress the cascading process of extreme events so as to protect the entire multi-layer infrastructure against global-scale breakdown.

Cooperation has been recognized as a fundamental driving force in many natural, social, and economical systems. Given a complex networked system in which agents (nodes) interact with one another according to the rules of evolutionary games and are subject to failure or death, can cooperation prevail and be optimized? In Chapter 7, I articulate a control scheme to maximize cooperation by introducing a time tolerance, a time duration that sustains an agent even if its payoff falls below a threshold. Strikingly, I find that a significant cooperation cluster can emerge when the time tolerance is approximately uniformly distributed over the network. A heuristic theory is derived to understand the optimization mechanism, which emphasizes the role played by medium-degree nodes. Implications to policy making to prevent/mitigate large-scale cascading breakdown are pointed out.

1.4. Spatiotemporal Patterns and Predictability of Cyberattacks

A relatively unexplored issue in cybersecurity science and engineering is whether there exist intrinsic patterns of cyberattacks. Conventional wisdom favors absence of such patterns due to the

overwhelming complexity of the modern cyberspace. Surprisingly, in Chapter 8, through a detailed analysis of an extensive data set that records the time-dependent frequencies of attacks over a relatively wide range of consecutive IP addresses, I successfully uncover intrinsic spatiotemporal patterns underlying cyberattacks, where the term “spatio” refers to the IP address space. In particular, I focus on analyzing *macroscopic* properties of the attack traffic flows and identify two main patterns with distinct spatiotemporal characteristics: deterministic and stochastic. Strikingly, there are very few sets of major attackers committing almost all the attacks, since their attack “fingerprints” and target selection scheme can be unequivocally identified according to the very limited number of unique spatiotemporal characteristics, each of which only exists on a consecutive IP region and differs significantly from the others. I utilize a number of quantitative measures, including the flux-fluctuation law, the Markov state transition probability matrix, and predictability measures, to characterize the attack patterns in a comprehensive manner. A general finding is that the attack patterns possess high degrees of predictability, potentially paving the way to anticipating and, consequently, mitigating or even preventing large-scale cyberattacks using macroscopic approaches.

2 . UNIVERSAL STRUCTURAL ESTIMATOR AND DYNAMICS APPROXIMATOR FOR COMPLEX NETWORKS

2.1. Introduction

A central issue in complexity science and engineering is systems identification and dynamical behavior prediction based on experimental or observational data. For a complex networked system, often the network structure and the nodal dynamical processes are unknown but only time series measured from various nodes in the network can be obtained. The challenging task is to infer the detailed network topology and the nodal dynamical systems from the available data. This line of pursuit started in biomedical science for problems such as identification of gene regulatory networks from expression data in systems biology [3–6] and uncovering various functional networks in the human brain from activation data in neuroscience [7–10]. The inverse problem has also been an area of research in statistical physics where, for example, the inverse Ising problem in static [11–15] and kinetic [16–22] situations has attracted continuous interest. Recent years have witnessed the emergence and growth of a subfield of research in complex networks: data based network identification (or reverse engineering of complex networks) [23–38]. In these works, the success of mapping out the entire network structure and estimating the nodal dynamical equations partly relies on taking advantage of the particular properties of the system dynamics in terms of the specific types and rules. For example, depending on the detailed dynamical processes such as continuous-time oscillations [29, 30, 32], evolutionary games [33], or epidemic spreading [37], appropriate mathematical frameworks uniquely tailored at the specific underlying dynamical process can be formulated to solve the inverse problem.

In this paper, we address the following challenging question: is it possible to develop a universal and completely data-driven framework for extracting network topology and identifying the

dynamical processes, without the need to know a priori the specific types of network dynamics? An answer to this question would be of significant value not only to complexity science and engineering but also to modern data science where the goal is to unearth the hidden structural information and to predict the future evolution of the system. Here we propose the concept of *universal structural estimator and dynamics approximator* for complex networked systems and demonstrate that such a framework or “machine” can indeed be developed for a large number of distinct types of network dynamical processes. Our approach will be a combination of numerical computation and physical reasoning. Since we are yet able to develop a rigorous mathematical framework for the universal machine, the present work should be regarded as an initial attempt towards the development of a universal framework for network reconstruction and dynamics prediction.

The key principle underlying our work is the following. In spite of the dramatic difference in the types of dynamics in terms of, e.g., the interaction setting and state updating rules, there are two common features shared by many dynamical processes on complex networks: (1) they are stochastic, first-order Markovian processes, i.e., only the current states of the systems determine their immediate future; and (2) the nodal interactions are local. The two features are characteristic of a *Markov network* (or a *Markov random field*) [39, 40]. In particular, a Markov network is an undirected and weighted probabilistic graphical model that is effective at determining the complex probabilistic interdependencies in situations where directionality in the interaction between connected nodes cannot be naturally assigned, in contrast to the directed Bayesian networks [39, 40]. A Markov network has two types of parameters: a node bias parameter that controls its preference of state choice, and a weight parameter characterizing the interaction strength of each undirected link. The joint probability distribution of the state variables $\mathbf{X} = (x_1, x_2, \dots, x_N)^T$ is given by $P(\mathbf{X}) = \prod_C \phi(\mathbf{X}_C) / \sum_{\mathbf{X}} \prod_C \phi(\mathbf{X}_C)$, where $\phi(\mathbf{X}_C)$ is the potential function for a well-connected

network clique C , and the summation in the denominator is over all possible system state $\mathbf{X}$. If this joint probability distribution is available, literally all conditional probability interdependencies can be obtained. The way to define a clique and to determine its potential function plays a key role in the Markov network's representation power of modeling the interdependencies within a particular system. To be concrete, in this work we pursue the possibility of modeling the conditional probability interdependence of a variety of dynamical processes on complex networked systems via a binary Ising Markov network with its potential function in the form of the Boltzmann factor, $\exp(-E)$, where E is the energy determined by the local states and their interactions along with the network parameters (the link weights and node biases) in a *log-linear* form [41]. This is effectively a sparse Boltzmann machine [41] adopted to complex network topologies without hidden units. (Note that hidden units usually play a critical role in ordinary Boltzmann machines [41]). We introduce a temporal evolution mechanism as a persistent sampling process for such a machine based on the conditional probabilities obtained via the joint probability, and generate a Markov chain of persistently sampled state configurations to form state transition time series for each node. For convenience, we call our model a *sparse dynamical Boltzmann machine* (SDBM).

For a dynamical process on complex networks, such as epidemic spreading or evolutionary game dynamics, the state of each node at the next time step is determined by the probability conditioned on the current states of its neighbors (and its own state in some cases). There is freedom to manipulate the conditional probabilities that dictate the system behavior in the immediate future by changing the values of its parameters, i.e., the weights and biases. A basic question is then, for an SDBM, is it possible to properly choose these parameters so that the conditional probabilities so produced are identical or nearly identical to those of a typical dynamical process with each given observed system state configuration? If the answer is affirmative, the SDBM can serve as a dynamics approximator

of the original system, and the approximated conditional probabilities possess predictive power for the system state at the next time step. When such an SDBM is found for many types of dynamical process on complex networks, it effectively serves as a universal dynamics approximator. Moreover, if the detailed statistical properties of the state configurations can be reproduced in the long time limit, that is, if the time series generated by this SDBM are statistically identical or nearly identical to those from the original system, then the SDBM is a generative model of the observed data (in the language of machine learning), which is potentially capable of long term prediction.

When a dynamics approximator exists for each type of dynamics on a complex network, the topology of the SDBM is nothing but that of the original network, providing a simultaneous solution to the problem of network structure reconstruction. Previous works on the inverse static or kinetic Ising problems led to methods of reconstruction for Ising dynamics by maximizing the data likelihood (or pseudo-likelihood) function via the gradient descent approaches [11–22]. Instead of adopting these approaches, as a part of our methodology to extract the network structure, we articulate a compressive sensing [42–47] based approach, whose working power has been demonstrated for a variety types of non-Ising type of dynamics on complex networks [32–38]. By incorporating a K-means clustering algorithm into the sparse solution obtained from compressive sensing, we demonstrate that nearly perfect reconstruction of the complex network topology can be achieved. Using 14 different types of dynamical processes on complex networks, we find that, if the time series data generated by these dynamical processes are assumed to be generated from its equivalent SDBMs, the universal reconstruction framework is capable of recovering the underlining network structure of each original dynamics with almost zero error. This represents solid and concrete evidence that SDBM is capable of serving as a universal structural estimator for complex networks. In addition to being able to precisely reconstruct the network topologies, the SDBM also allows the

link weights and node biases to be calculated with high accuracy.

Our method is fully automated and does not require any subjective parameter choice.

2.2. Results

2.2.1. SDBM as a Network Structural Estimator

For an SDBM of size N , the probability that the system is in a particular binary state configuration $\mathbf{X}_{N \times 1} = (x_1, x_2, \dots, x_i, \dots, x_N)^T$ is given by

$$P(\mathbf{X}) = \frac{\exp(-E_{\mathbf{X}})}{\sum_{\mathbf{X}} \exp(-E_{\mathbf{X}})}, \quad (2.1)$$

with $E_{\mathbf{X}}$ being the total energy of the network in $\mathbf{X}$:

$$E_{\mathbf{X}} = \mathbf{X}^T \cdot \mathbf{W} \cdot \mathbf{X} = \sum_{i=1, i \neq j}^N \sum_{j=1, j \neq i}^N w_{ij} x_i x_j - \sum_{i=1}^N b_i x_i, \quad (2.2)$$

where x_i and x_j are binary variables (0 or 1) characterizing the state of nodes i and j , respectively, $\mathbf{W}$ is a weighted matrix with its off diagonal elements $w_{ij} = w_{ji}$ ($i, j = 1, \dots, N$, $i \neq j$) specifying the weight associated with the link between nodes i and j . The i th diagonal element of $\mathbf{W}$ is chosen as the bias parameter b_i for node i ($i = 1, \dots, N$), which determines node i 's preference to state 0 or 1. The total energy $E_{\mathbf{X}}$ includes the interaction energies (the sum of all $w_{ij} x_i x_j$ terms) and the nodes' self energies (the various $b_i x_i$ terms). The partition function of the system is given by

$$\mathcal{Z} = \sum_{\mathbf{X}} \exp(-E_{\mathbf{X}}), \quad (2.3)$$

where the summation is over all possible $\mathbf{X}$ configurations. The state of x_i at the next time step is determined by the states of all the other nodes at the present time step, $\mathbf{X}_i^R$, via the following conditional probability

$$P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)] = \frac{P[x_i(t+1) = 1, \mathbf{X}_i^R(t)]}{P[x_i(t+1) = 1, \mathbf{X}_i^R(t)] + P[x_i(t+1) = 0, \mathbf{X}_i^R(t)]}, \quad (2.4)$$

where the two joint probabilities are given by

$$\begin{aligned}
P[x_i(t+1) = 1, \mathbf{X}_i^R(t)] &= \\
&\frac{1}{\mathcal{Z}} \exp \left[- \sum_{j=1, j \neq i}^N w_{ij} x_j(t) - b_i - \sum_{s=1, s \neq i}^N \sum_{j=1, j \neq i}^N w_{sj} x_s(t) x_j(t) - \sum_{s=1, s \neq i}^N b_s x_s(t) \right], \\
P[x_i(t+1) = 0, \mathbf{X}_i^R(t)] &= \frac{1}{\mathcal{Z}} \exp \left[- \sum_{s=1, s \neq i}^N \sum_{j=1, j \neq i}^N w_{sj} x_s(t) x_j(t) - \sum_{s=1, s \neq i}^N b_s x_s(t) \right].
\end{aligned}$$

A Markov network defined in this fashion is also called a kinetic Ising model [16–22]. With the joint probabilities, the conditional probability in Eq. (2.4) becomes

$$P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)] = \frac{1}{1 + \exp \left[\sum_{j=1, j \neq i}^N w_{ij} x_j(t) + b_i \right]}. \quad (2.5)$$

We thus have

$$\ln \left(\frac{1}{P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]} - 1 \right) = \sum_{j=1, j \neq i}^N w_{ij} x_j(t) + b_i.$$

Letting $Q_i(t) \equiv (P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)])^{-1} - 1$, we have

$$\ln Q_i(t) = \begin{pmatrix} x_1(t), & \dots, & x_{i-1}(t), & x_{i+1}(t), & \dots, & x_N(t), & 1 \end{pmatrix} \begin{pmatrix} w_{i1} \\ \vdots \\ w_{i(i-1)} \\ w_{i(i+1)} \\ \vdots \\ w_{iN} \\ b_i \end{pmatrix} \quad (2.6)$$

For M distinct time steps $t_1, t_2, \dots, t_M$, we obtain the following matrix form:

$$\begin{pmatrix} \ln Q_i(t_1) \\ \ln Q_i(t_2) \\ \vdots \\ \ln Q_i(t_M) \end{pmatrix} = \begin{pmatrix} x_1(t_1), & \dots, & x_{i-1}(t_1), & x_{i+1}(t_1), & \dots, & x_N(t_1), & 1 \\ x_1(t_2), & \dots, & x_{i-1}(t_2), & x_{i+1}(t_2), & \dots, & x_N(t_2), & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1(t_M), & \dots, & x_{i-1}(t_M), & x_{i+1}(t_M), & \dots, & x_N(t_M), & 1 \end{pmatrix} \begin{pmatrix} w_{i1} \\ \vdots \\ w_{i(i-1)} \\ w_{i(i+1)} \\ \vdots \\ w_{iN} \\ b_i \end{pmatrix}, \quad (2.7)$$

which can be written concisely as

$$\mathbf{Y}_{M \times 1} = \Theta_{M \times N} \cdot \mathbf{V}_{N \times 1}, \quad (2.8)$$

where the vector $\mathbf{Y}_{M \times 1} \in R^M$ contains the values of $\ln Q_i(t)$ for M different time steps, the $M \times N$ matrix $\Theta_{M \times N}$ is determined by the states of all the nodes except i , and the first $(N - 1)$ components of the vector $\mathbf{V}_{N \times 1} \in R^N$ are the link weights between node i and all other nodes in the network [illustrated in Fig. 1 (a)] with its last entry being node i 's intrinsic bias.

Since the conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ depends solely on the state configuration of i 's nearest neighbors, or i 's *Markov blanket* [39,40] at time t , as shown in Fig. 1(b) identical configurations at other time steps imply identical conditional probabilities. Thus, given time series data of the dynamical process, the conditional probability can be estimated according to the law of large numbers by averaging over the states of i at all the time steps slightly before the neighboring

state configurations become identical. Note, however, that this probability needs to be conditioned on the state configuration of the entire system except node i , i.e., on $\mathbf{X}_i^R(t)$, and the average of x_i is calculated over all the time steps $t_m + 1$ satisfying $\mathbf{X}_i^R(t) = \mathbf{X}_i^R(t_m)$. This means that there is a dramatic increase in the configuration size, i.e., from k_i (the degree of node i) to $N - 1$ (the size of the vector $\mathbf{X}_i^R$), which makes the number of exactly identical configurations too small to give any meaningful statistics. To overcome this difficulty, we allow a small amount of dissimilarity between $\mathbf{X}_i^R(t)$ and $\mathbf{X}_i^R(t_m)$ by introducing a tolerance parameter, Γ , to confine the corresponding Hamming distances normalized by N (we assume $\mathbf{X}_i^R(t) \approx \mathbf{X}_i^R(t_m)$ if the relative difference between them is no larger than Γ/N). This averaging process leads to

$$\begin{pmatrix} \ln q_i(t_1) \\ \ln q_i(t_2) \\ \vdots \\ \ln q_i(t_M) \end{pmatrix} \quad (2.9)$$

$$= \begin{pmatrix} \langle x_1(t_1) \rangle, & \dots, & \langle x_{i-1}(t_1) \rangle, & \langle x_{i+1}(t_1) \rangle, & \dots, & \langle x_N(t_1) \rangle, & 1 \\ \langle x_1(t_2) \rangle, & \dots, & \langle x_{i-1}(t_2) \rangle, & \langle x_{i+1}(t_2) \rangle, & \dots, & \langle x_N(t_2) \rangle, & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \langle x_i(t_M) \rangle, & \dots, & \langle x_{i-1}(t_M) \rangle, & \langle x_{i+1}(t_M) \rangle, & \dots, & \langle x_N(t_M) \rangle, & 1 \end{pmatrix} \begin{pmatrix} w_{i1} \\ \vdots \\ w_{i(i-1)} \\ w_{i(i+1)} \\ \vdots \\ w_{iN} \\ b_i \end{pmatrix},$$

where $q_i(t) \equiv \langle x_i(t+1) \rangle^{-1} - 1$, and $\langle \cdot \rangle$ stands for the averaging over all instants of time at which the condition $\mathbf{X}_i^R(t) = \mathbf{X}_i^R(t_m)$ is satisfied. A schematic illustration of whole process is presented in Fig. 1, with Eq. 2.9 shown in panel (c).

For a complex network, the degree of a node is typically small compared with the network size. For node i , the link weights are nonzero only for the connections with the immediate neighbors. The vector $\mathbf{V}_{N \times 1}$ is thus typically sparse with the majority of its elements being zero. The sparsity property renders applicable compressive sensing [42–47], through which an N dimensional sparse vector can be reconstructed via a set of M linear measurements, for $M \ll N$. By minimizing the L_1 norm of $\mathbf{V}_{N \times 1}$, i.e., $\|\mathbf{V}_{N \times 1}\|_1 = \sum_{j=1, j \neq i}^N |w_{ij}| + |b_i|$, subject to $\mathbf{Y}_{M \times 1} = \Theta_{M \times N} \cdot \mathbf{V}_{N \times 1}$, we can reconstruct $\mathbf{V}_{N \times 1}$ to obtain the connection weights between node i and all other nodes in the network. One would hope that, applying the procedure to each node would lead to the complete weighted adjacency matrix, $\mathbf{W}$.

To gain insights, we test the structural estimation method for an SDBM itself [Figs. 2(a-c)] and three different types of dynamical processes [Figs. 2(d-f)] by feeding the time series data generated by an SDBM into the framework and comparing the reconstructed SDBM with the original machine. The time instants $t_1, t_2, \dots, t_M$ needed to construct Eq. (2.9) are chosen randomly from T time instants in total. For each node, the compressive sensing algorithm is implemented a multiple of times to obtain the averaged relevant quantities. As shown in Fig. 2, we see that the averaged solutions from the compressive sensing algorithm appear in sharp peaks at places that correspond to the existent links, despite the large differences among the values. This means that, while most existent links can be predicted against the null links, the accuracy of the solution so obtained is not sufficient for the actual element values of $\mathbf{V}_{N \times 1}$ to be determined. For the null links, the corresponding solutions generally appear as a noisy background. For the ideal case where any w_{ij} is zero, the background noise can be quite large especially for the large degree nodes, as shown in Figs. 2(b) and 2(d). Further calculations indicate that averaging a larger number of simulation runs can suppress the background noise to certain extent, but it cannot be eliminated and may become

quite influential for various types of dynamics (Sec. 2.2.2).

In previous works on reconstruction of complex networks based on stochastic dynamical correlations [29, 30] or compressive sensing [33, 37], the existent (real) links can be distinguished from the nonexistent links by setting a single threshold value in certain quantitative measure. The success relies on the fact that the dynamics at various nodes are of the same type, and the reconstruction algorithm is tailored toward the specific type of dynamical process. Our task is significantly more challenging as the goal is to develop a universal system (or machine) to replicate a diverse array of dynamical processes based on data. As can be seen in Figs. 2(a-c), for compressive sensing based reconstruction, the computational criteria to distinguish existent from nonexistent links differ substantially for different dynamics in terms of quantities such as the solution magnitude, peak value distribution, and background noise intensity. As a result, a more elaborate and sophisticated procedure is required for determining the threshold for each particular case, suggesting that a straightforward application of compressive sensing cannot lead to a universal reconstruction algorithm. One may also regard the solutions of the existent links as a kind of *extreme events* [48–50] on top of the random background, but it is difficult to devise a universal criterion to judge if a peak in the distribution of the quantitative measure represents the correct extreme event corresponding to an existent link.

Through extensive testing, we find that a previously developed unsupervised clustering measure, the K-means [39, 40], possesses the desired traits that can be exploited in combination with compressive sensing to develop a universal reconstruction machine (see **Materials and Methods** for details). As we will show, K-means can serve as a base for a highly effective structural estimator for various types of dynamics on networks of distinct topologies. Depending on the specific combination of the network topology and dynamics, the reconstruction accuracies vary to certain extent but

are still acceptable. Since the compressive sensing operation is node specific, the solutions obtained separately from different nodes may give conflicting information as to whether there is actually a link between the two nodes, requiring a proper resolution scheme. We develop such a scheme based on node degree consistency (see **Materials and Methods**). Our universal reconstruction machine thus contains three main components: compressive sensing, K-means, and conflict resolution. We shall demonstrate that the machine can separate the true positive solutions from the noisy background with high success rate, for all combinations of the nodal dynamics and the network topology tested.

2.2.2. Universality of SDBM as a Network Structural Estimator

Figure 3 demonstrates the performance of network structural estimation for SDBM dynamics on random and scale-free networks in terms of the amount of data, characterized by the number of measurements M , normalized by N , for SDBMs built up for systems with different original topologies. We define R_e^1 and R_e^0 as the estimation error rates for the existent and non-existent links, namely, the false positive and the false negative rates, respectively. We see that, for a wide range of the values of M/N , the success rates of the existent and non-existent links, $1 - R_e^1$ and $1 - R_e^0$, respectively, are nearly 100% for homogeneous network topology. For heterogeneous (scale-free) networks, the success rate $1 - R_e^1$ tends to be slightly lower than 100%, due to the violation of the sparsity condition for hub nodes, leading to the difficulty to distinguish the peaks in the distribution of the compressive sensing solutions from the noisy background. For $M/N = 1$, the success rates are reduced slightly, due to the non-zero dissimilarity tolerance Γ that makes the values of $\mathbf{X}_i^R$ at different time steps indistinguishable and introduces linear dependence into Eq. (2.9). As a result, the accuracy of compressive sensing solutions is compromised to certain extent. Generally, high reconstruction precision is guaranteed for almost any value choice of M as long as it is not too

close to 0 and N , and this feature makes our methodology free of subjective parameters.

Our working hypothesis is that, for a networked system with a certain type of nodal dynamics, there exists an equivalent SDBM. Reconstructing the structure of the SDBM would simultaneously give the topology of the original networked dynamical system. Accordingly, the time series data generated from the original system can be used to reveal its underlining interaction structures through the corresponding SDBM. In particular, we directly feed the original time series into the framework of compressive sensing and K-means for SDBM to generate the network structure, and test whether the structure generated by the SDBM based reconstruction mimics that of the original network. The similarity can be quantified by the error rates of the existent and non-existent links. Table 1 list the 14 dynamical processes on model networks and the underlying conditional probabilities, and Tab. 2 shows the performance in terms of the percentage error rates R_e^0 and R_e^1 . The 14 types of dynamical processes are taken from the fields of evolutionary game theory, opinion dynamics, and spreading processes, covering a number of focused research topics in complex networks. Strikingly, for all the dynamics with diverse properties, we find that that, for each and every dynamics-network combination, zero or nearly zero error rates are obtained for both the existent and non-existent links, revealing a strong similarity between the original networks and the ones generated from SDBM, regardless of the type of dynamics. The nonzero error rates in Tab. 2 come mainly from the high degree nodes. Consequently, as indicated in Tab. 2, the reconstruction accuracy for networks of homogeneous topology is generally higher than that for heterogeneous networks.

For certain types of evolutionary game dynamics, especially for the snowdrift game (SG) [24] and the prisoner's dilemma game (PDG) [20–23] with the Fermi updating rule [21], information about the state configuration of the second nearest neighbors is required to calculate the payoffs

Fig. 2. Solution Examples of Compressive Sensing and the Corresponding K-Means Clustering Results. The compressive sensing solutions (green squares) are compared, element-wise, with the true element values of the original connection vectors (dark purple triangles), where the x-axis is the index of nodes with different degree values for different network topologies and dynamics types. The target nodes and the correctly detected existent connections via the K-means are marked as light blue and light purple bars, respectively. The false negative and the false positive estimations are marked as red and gray bars, respectively. For all cases, the amount of linear measurements is $M = 0.4N$. The restriction on Hamming distances is set to be $\Gamma = 0.35$ for all examples, and a small change in Γ would not affect the qualitative results. (a) A node with the smallest degree $k = 2$ in a BA scale-free network of size $N = 100$ and average degree $\langle k \rangle = 4$ with SDBM dynamics. (b) The node of the largest degree $k = 18$ in the same network. (c) A node of degree $k = 8$ in an ER random network of size $N = 100$ and average degree $\langle k \rangle = 4$ with SDBM dynamics. (d) The node with the largest degree $k = 19$ in a BA scale-free network of size $N = 100$ and average degree $\langle k \rangle = 4$ with prisoner's dilemma game (PDG) dynamics. (e) A node of degree $k = 11$ in a real-world social network of size $N = 67$ subject to CP dynamics. (f) A node of degree $k = 10$ in a real-world electrical circuit network of size $N = 122$ with Language model dynamics. In (d-f), the element values of the original connection vector are set to unity since no true weight values are given. For cases (a,b,c,f), there are no false negative detections, i.e., all existent links have been

Fig. 3. Structural Estimation Success Rates and Parameter Estimation Errors Versus the Normalized Data Amount. For SDBM dynamics on (a) ER-random and (b) BA scale-free networks, the success rates of detection of existent links and identification of nonexistent links, $1 - R_e^1$ (green circles) and $1 - R_e^0$ (white circles), respectively, versus the normalized data amount M/N with error bars. The absolute values of the errors in estimating the link weights and the nodal biases for the existent links (purple squares) and non-existent links (white squares) are also shown. Most data points have quite small error bars, indicating the performance stability of our method. High performance structural and parameter estimations are achieved insofar as M/N exceeds about 10%.

TABLE 1

Description of the 14 Dynamical Processes on Model Networks and the Conditional Probabilities.

The quantity $P_i^{0 \rightarrow 1}$ or $P_i^{1 \rightarrow 0}$ is the probability that the state of node i (of degree k_i) becomes 1 at the next time step when its current state is 0, or vice versa, conditioned on its current neighbor state

configuration. The number of i 's neighbors in state 1 is n_i . (See Supplementary Note 1 for a detailed description of the dynamics and parameters for estimating the conditional probabilities

$$P_i^{0 \rightarrow 1} \text{ and } P_i^{1 \rightarrow 0}.)$$

Category	Dynamics Type	$P_i^{0 \rightarrow 1}$	$P_i^{1 \rightarrow 0}$
I	SDBM	$\frac{1}{1 + \exp[\sum_{j=1, j \neq i}^N w_{ij} x_j(t) + b_i]}$	$\frac{\exp[\sum_{j=1, j \neq i}^N w_{ij} x_j(t) + b_i]}{1 + \exp[\sum_{j=1, j \neq i}^N w_{ij} x_j(t) + b_i]}$
II	Ising Glauber [1]	$\frac{1}{1 + \exp[\frac{2J}{\kappa}(k_i - 2n_i)]}$	$\frac{\exp[\frac{2J}{\kappa}(k_i - 2n_i)]}{1 + \exp[\frac{2J}{\kappa}(k_i - 2n_i)]}$
	SQ-SG [25]	$\frac{1}{1 + \exp[(rk_i - n_i)/\kappa]}$	$\frac{\exp[(rk_i - n_i)/\kappa]}{1 + \exp[(rk_i - n_i)/\kappa]}$
	SQ-PDG [25]	$\frac{1}{1 + \exp[(b-1)(k_i - n_i)/\kappa]}$	$\frac{\exp[(b-1)(k_i - n_i)/\kappa]}{1 + \exp[(b-1)(k_i - n_i)/\kappa]}$
III	Minority game [4–8]	$\frac{k_i - n_i}{k_i}$	$\frac{n_i}{k_i}$
	Voter [2, 3]	$\frac{n_i}{k_i}$	$\frac{k_i - n_i}{k_i}$
	Majority vote [2, 3]	$\begin{cases} Q & \text{if } n_i < k_i/2 \\ 1/2 & \text{if } n_i = k_i/2 \\ 1 - Q & \text{if } n_i > k_i/2 \end{cases}$	$\begin{cases} 1 - Q & \text{if } n_i < k_i/2 \\ 1/2 & \text{if } n_i = k_i/2 \\ Q & \text{if } n_i > k_i/2 \end{cases}$
IV	Link-update voter [10, 11]	$\frac{n_i}{\langle k \rangle}$	$\frac{k_i - n_i}{\langle k \rangle}$
	Language model [12, 13]	$S(\frac{n_i}{k_i})^\alpha$	$(1 - S)(\frac{k_i - n_i}{k_i})^\alpha$
	Kirman [14, 15, 15]	$c_1 + dn_i$	$c_2 + d(k_i - n_i)$
V	CP [18, 19]	$\frac{n_i}{k_i} \lambda_i$	μ_i
	SIS [16, 69–71]	$1 - (1 - \lambda_i)^{n_i}$	μ_i
VI	SG [24]	*	*
	PDG [20–23]	*	*

TABLE 2

Reconstruction Error Rates. For the same dynamical processes in Tab. 1, the error rates (in percentages) in uncovering the existent and non-existent links, R_e^1 and R_e^0 , respectively.

ER-random and BA scale-free of size $N = 100$ and average degree $\langle k \rangle = 4$ are used as the underlining network supporting the various dynamical processes. The normalized number of linear measurements used in compressive sensing is $M/N = 0.4$ for all cases.

Category	Dynamics Type	R_e^0 / R_e^1 (% , ER)	R_e^0 / R_e^1 (% , BA)
I	SDBM	0.0 / 0.1	0.1 / 3.4
II	Ising Glauber [1]	0.0 / 0.0	0.0 / 0.0
	SQ-SG [25]	0.0 / 0.0	0.0 / 3.2
	SQ-PDG [25]	0.0 / 0.0	0.3 / 3.3
III	Minority game [4–8]	0.0 / 0.6	0.0 / 0.2
	Voter [2, 3]	0.0 / 0.0	0.0 / 0.0
	Majority vote [2, 3]	0.0 / 0.0	0.0 / 0.1
IV	Link-update voter [10, 11]	0.0 / 0.0	1.8 / 3.6
	Language model [12, 13]	0.0 / 0.0	0.0 / 0.4
	Kirman [14, 15, 15]	0.0 / 2.3	0.0 / 3.8
V	CP [18, 19]	0.0 / 1.4	0.0 / 0.4
	SIS [16, 69–71]	0.3 / 5.2	0.1 / 15.6
VI	SG [24]	0.0 / 1.7	0.0 / 9.4
	PDG [20–23]	0.1 / 9.6	0.1 / 9.8

of the first nearest neighbors. In such a system, the next move of a target node is determined by comparing its payoff from those of its neighbors. This implies that, using solely the state configuration information of the Markov blanket, without the aid of the payoff information that requires the state information of the second nearest neighbors, is insufficient to determine the state of the target node into the immediate future, rendering inapplicable our SDBM based reconstruction. Contrary to this intuition, we find that for both SG and PDG, high reconstruction accuracy can be achieved, as shown in Tab. 2. Recall that our reconstruction method is formulated based on the independence assumption of Markov networks, i.e., in order to reconstruct the local structure of a target node, it should be completely independent of the rest of the system when the configuration of its Markov blanket is given. The results in Tab. 2 indicate that our SDBM based algorithm performs better than expected in terms of the reconstruction accuracy. In fact, the independence assumption can be made to hold by adopting a self-questioning (SQ) based updating rule (see Supplementary Note 1 for details). In this case, excellent reconstruction accuracy is obtained, as can be seen from Tab. 2.

The structural estimation results reported so far are based on model network topologies. Real-world complex networks have also been used to test our framework, as shown in Figs. 2(e,f). Tables 3 and 4 list a number of real world networks and the reconstruction accuracy. For various combinations of network topology and dynamical process, high reconstruction accuracy is achieved, where for a number of cases the error rates are essentially zero. There are a few special cases where the errors are relatively large, corresponding to situations where the globally frozen or oscillating states dominate the dynamical process so that too few linearly independent measurements can be obtained. Overall, the equivalent SDBM correspondence and the effectiveness of our reconstruction scheme for real-world networks can be expected.

TABLE 3

Structure Reconstruction Error Rates of 5 Real-World Networks Associated with 11 Typical Complex Network Dynamics. The percentage error rates of the non-existent and the existent links detection (R_e^0 / R_e^1 , %) are listed for all combinations of the network structure and dynamics. Each entry denotes the error rates R_e^0 / R_e^1 corresponding to a particular type of dynamics (shown in the column) for a real-world network topology (shown in the row). The result for each combination is obtained with $M = 0.4N$ linear measurements used in each implementation. For some combinations, there exists a global absorption state in which the system state configuration freezes, or a vibration state where the system only switches between a small number of configurations. In such a case, not sufficient number of linearly independent measurements can be obtained to implement compressive sensing. The entries associated with these cases are marked by “*/*”. A detailed description of the real world networks can be found in Supplementary Note 2.

Dynamics Type	Social-1	Social-2	Protein-1	Protein-2	Protein-3
Ising Glauber	0.2 / 3.8	0.0 / 0.0	0.0 / 0.0	1.0 / 0.0	0.0 / 0.0
Minority game	0.0 / 0.0	0.0 / 7.0	3.7 / 0.0	* / *	* / *
Voter	0.0 / 0.0	0.0 / 4.2	3.9 / 0.8	* / *	* / *
Majority vote	0.0 / 0.0	0.0 / 32.4	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
Link-update voter	3.0 / 2.5	0.1 / 7.0	0.2 / 0.0	0.0 / 0.0	2.7 / 0.0
Language model	0.2 / 0.0	0.0 / 0.0	0.1 / 0.0	2.1 / 0.0	* / *
Kirman	0.0 / 0.0	0.5 / 34.5	0.0 / 0.0	0.0 / 0.0	0.0 / 0.0
CP	0.2 / 0.0	0.0 / 7.0	0.0 / 0.0	* / *	* / *
SIS	4.7 / 2.5	0.0 / 14.1	0.4 / 0.0	* / *	* / *
SG	0.9 / 5.0	0.5 / 2.8	1.8 / 0.0	1.6 / 0.0	* / *
PDG	0.5 / 0.0	0.9 / 4.2	2.4 / 0.0	3.3 / 0.0	13.9 / 16.0

TABLE 4

Structure Reconstruction Error Rates of 4 Additional Real-World Networks Associated with 11

Types of Complex Network Dynamics. Table legends are the same as for Tab. 3.

Dynamics Type	Food Web-1	Food Web-2	Food Web-3	Circuit
Ising Glauber	* / *	0.0 / 0.0	* / *	0.0 / 0.0
Minority game	0.1 / 0.0	1.0 / 0.0	0.4 / 0.3	0.8 / 0.0
Voter	0.1 / 0.0	0.7 / 0.0	0.6 / 1.7	2.0 / 0.0
Majority vote	0.4 / 0.0	0.0 / 0.0	1.5 / 0.0	0.0 / 0.0
Link-update voter	1.1 / 0.0	15.7 / 0.0	* / *	* / *
Language model	0.2 / 0.0	0.3 / 0.0	0.0 / 4.4	0.0 / 0.1
Kirman	* / *	0.0 / 2.2	* / *	0.0 / 0.0
CP	0.4 / 0.5	0.0 / 0.7	5.3 / 1.5	0.0 / 0.0
SIS	1.3 / 9.4	4.7 / 8.0	17.3 / 17.3	* / *
SG	2.1 / 16.1	8.4 / 25.6	6.2 / 23.0	8.3 / 4.8
PDG	2.8 / 7.6	8.4 / 19.0	8.8 / 15.3	23.0 / 14.3

2.2.3. Parameter Estimation Scheme

From the reconstructed network structure, any node i 's k_i immediate neighbors, $m_1, m_2, \dots, m_{k_i}$, and their state configuration, $\mathbf{X}_i^{\text{MB}} = (x_{m_1}, x_{m_2}, \dots, x_{m_{k_i}})^T$, can be identified. Since the probability that i 's state is 1 at the next time step depends only on i 's immediate neighbors, $\mathbf{X}_i^{\text{R}}(t)$ in Eq. (2.5) can be simplified to $\mathbf{X}_i^{\text{MB}}(t)$, and Eq. (2.5) can be written as

$$P[x_i(t+1) = 1 | \mathbf{X}_i^{\text{R}}(t)] = P[x_i(t+1) = 1 | \mathbf{X}_i^{\text{MB}}(t)] = \frac{1}{1 + \exp[\sum_{m=1}^{k_i} w_{im}x_m(t) + b_i]}. \quad (2.10)$$

Accordingly, Eq. (2.9) can be simplified as

$$\mathbf{Y}_{(k_i+1) \times 1}^{\text{MB}} = \Theta_{(k_i+1) \times (k_i+1)}^{\text{MB}} \cdot \mathbf{V}_{(k_i+1) \times 1}^{\text{MB}},$$

i.e.,

$$\begin{pmatrix} \ln q_i(t_1) \\ \ln q_i(t_2) \\ \vdots \\ \ln q_i(t_{k_i+1}) \end{pmatrix} = \begin{pmatrix} \langle x_{m_1}(t_1) \rangle, & \langle x_{m_2}(t_1) \rangle, \dots, & \langle x_{m_{k_i}}(t_1) \rangle, & 1 \\ \langle x_{m_1}(t_2) \rangle, & \langle x_{m_2}(t_2) \rangle, \dots, & \langle x_{m_{k_i}}(t_2) \rangle, & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \langle x_{m_1}(t_{k_i+1}) \rangle, & \langle x_{m_2}(t_{k_i+1}) \rangle, \dots, & \langle x_{m_{k_i}}(t_{k_i+1}) \rangle, & 1 \end{pmatrix} \begin{pmatrix} w_{im_1} \\ w_{im_2} \\ \vdots \\ w_{im_{k_i}} \\ b_i \end{pmatrix}, \quad (2.11)$$

where the $k_i + 1$ linear equations uniquely solve $w_{im_1}, \dots, w_{im_{k_i}}$ and b_i simply via

$$\mathbf{V}_{(k_i+1) \times 1}^{\text{MB}} = [\Theta_{(k_i+1) \times (k_i+1)}^{\text{MB}}]^{-1} \cdot \mathbf{Y}_{(k_i+1) \times 1}^{\text{MB}}.$$

Our parameter estimation formulation is illustrated schematically in Fig. 4(a).

Fig. 4. A Schematic Illustration of SDBM Parameter Estimation. (a) The parameter estimation framework for the corresponding SDBM of the same system shown in Fig. 1 (a,b). It is formulated by Eq. (2.11). The connections of the network are already known via the structure reconstruction step, and thus the entries are only obtained from a node's Markov blanket [shown in Fig. 1(b)]. This calculation is implemented for each node in the system. (b) The values of the link weights and the node biases (shown as different colors) for the corresponding SDBM are calculated via the parameter estimation scheme in (a) and the degree guided solution substitution operation (see Materials and Methods).

For a particular neighboring state configuration of node i , $\mathbf{X}_i^{\text{MB}}(t)$, its occurring frequency determines the precision in the estimation of $P[x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)] \simeq \langle x_i(t+1) \rangle$, which in turn determines the solution precision of Eq. (2.11). The occurrence of different neighboring state configurations for the same node may differ dramatically. Furthermore, the accuracy in estimating the probability $P[x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)]$ depends on the node degree due to the increasing difficulty in finding exactly the same configurations for larger degree nodes. Overcoming the estimation difficulty is a highly non-trivial problem. Exploiting the available data further, we develop a degree guided solution substitution operation to cope with this difficulty, where sufficient estimation precision is guaranteed for most cases (see **Materials and Methods**). A SDBM with estimated parameters is schematically shown in Fig. 4(b).

2.2.4. Universal Dynamics Approximator

In Table 1, the complex network dynamics are categorized in terms of the specific probabilities $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$. In particular, category I is for the SDBM dynamics and category II contains dynamical processes with $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$ having a mathematical form similar to that of the SDBM. For category III, the forms of $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$ for the three types of dynamics are quite different. Despite the differences among the dynamical processes in the three categories, they share a key property: i.e., $P_i^{0 \rightarrow 1} + P_i^{1 \rightarrow 0} = 1$, which plays a critical role in implementing the parameter estimation. For the processes in categories IV and V, we have $P_i^{0 \rightarrow 1} + P_i^{1 \rightarrow 0} \neq 1$, where $P_i^{1 \rightarrow 0}$ is a time-invariant constant for the nodal dynamics in category V.

We first test our parameter estimation scheme using the state time series generated by the SDBM, which can be validated through a direct comparison of the estimated parameter values with their true values, as shown in Figs. 2(a-c). However, since the parameters only affect the system collectively (not individually) in the form of the product summation [Eq. (2.5)] and our goal is to as-

Fig. 5. Comparison Between the Conditional Probability of the Original System and That Generated Via the Reconstructed SDBM. (a) Conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ versus time t for the original system with SDBM dynamics for a BA-scale free network and (b) the reconstructed approximator using the configuration time series of the original system, where $1 \leq i \leq N$ (the y-axis) and $1 \leq t \leq T$ (the x-axis). (c) Time series of the conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ for the original system with SIS dynamics for a BA-scale free network and (d) the reconstructed approximator.

assess the predictive power, we introduce an alternative validation scheme. For each particular system configuration, the acceptable parameter estimation results would serve as a base to generate another SDBM with identical conditional probabilities for each node at each time step, as compared with those of the original system. Figures 5(a) and 5(b) present a comparison between the conditional probabilities $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ of the original SDBM and the estimated SDBM calculated from Eq. (2.5). We see that the two time series of $P(x_i(t+1) = 1 | \mathbf{X}_i^R(t))$ are visually indistinguishable. Figure 6(a) shows the error distributions of the estimated conditional probability time series, where an overwhelmingly sharp peak occurs at 0, indicating an excellent agreement between the estimated and the true parameter values.

For typical dynamical processes on complex networks, our goal is to find the equivalent SDBMs whose true parameter values are not available. In this case, the performance of the parameter estimation scheme can be assessed through the reconstructed conditional probabilities. Based on the recovered network structure, the time series of the network dynamics are fed into the parameter estimation scheme, and the link weights and the nodal biases are obtained to form a system obeying the SDBM dynamics. Corresponding to the categories in Table 1, the distributions of the estimation errors between the original conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ and the one generated by the newly constructed SDBM are shown in Fig. 6, where panels (a-f) show the error distributions corresponding to the six types of dynamics in categories II and III in Table 1, respectively. In each case, a sharp peak at zero dominates the distribution, indicating the equivalence of the reconstructed SDBM to the original dynamics. Given a particular type of complex network dynamics, the SDBM resulted from our structural and parameter estimation scheme is indeed equivalent to the original dynamical system. The limited amount of data obtained from the original system renders important state prediction of the system, a task that can be accomplished by taking advantage of the equiva-

lence of the SDBM to the original system in the sense that the SDBM produces approximately equal state transition probabilities in the immediate future, given the current system configuration. The SDBM thus possesses a significant predictive power for the original system. Thus, regardless of the type of the dynamical process, insofar it satisfies the relation $P_i^{0 \rightarrow 1} + P_i^{1 \rightarrow 0} = 1$, the reconstructed SDBM can serve as a universal dynamics approximator.

For an SDBM, the relation $P_i^{1 \rightarrow 1} = 1 - P_i^{1 \rightarrow 0} = P_i^{0 \rightarrow 1}$ holds in general. However, for the dynamical processes in categories IV and V, we have $P_i^{1 \rightarrow 1} \neq 1 - P_i^{1 \rightarrow 0} = P_i^{0 \rightarrow 1}$ so that a single SDBM is not sufficient to fully characterize the dynamical evolution. Our solution is to construct two SDBMs, A and B, each associated with one of the two cases: $x_i(t) = 0$ and $x_i(t) = 1$, respectively. The link weights w_{ij}^A (or w_{ij}^B) and the nodal bias b_i^A (or b_i^B) for node i in SDBM A (or B) are computed for all the time steps t satisfying $x_i(t) = 0$ (or $x_i(t) = 1$), leading to $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ for $x_i(t) = 0$ (or $x_i(t) = 1$) from SDBM A (or B). Using this strategy, the dominant peaks at zero persist in the error distributions for the dynamical processes in category IV, as shown in Figs. 6(g-i). For the epidemic spreading dynamics (CP and SIS) in category V, the fixed value of $P_i^{1 \rightarrow 0}$ for each node i can be acquired through $P_i^{1 \rightarrow 0} \simeq \langle x_i(t_1 + 1) \rangle_{t_1}$, where $\langle x_i(t_1 + 1) \rangle_{t_1}$ stands for the average of $x_i(t_1 + 1)$ over all values of t_1 satisfying $x_i(t_1) = 1$. Through this approach, SDBM B is in fact a network without links but with each node's bias satisfying $\mu_i = \exp(b_i) / [1 + \exp(b_i)]$, where μ_i is node i 's recovery rate. Figures 6(j) and 6(k) show the error distributions of the conditional probability recovery for the spreading processes, where we see that the errors are essentially zero. If the method of solving SDBM B in category IV is adopted to the dynamical processes in other categories, i.e., without any prior knowledge about $P_i^{1 \rightarrow 0}$, the resulted SDBM would have nearly identical link weight values with respect to SDBM A (in categories I-III) or have close-to-zero link weights and $\mu_i \simeq \exp(b_i) / [1 + \exp(b_i)]$ for category V,

despite that their conditional probability recovery errors may be slightly larger than those in Fig. 6. The persistent occurrence of a dominant peak at zero in error distribution implies the power of the combined SDBMs as a universal dynamics approximator, regardless of the specific settings of the transition probability. When limited prior knowledge about $P_i^{1 \rightarrow 0}$ is available, SDBM B can be simplified or even removed without compromising the estimation accuracy. In general, the universal approximator has a significant short-term predictive power for arbitrary types of dynamics on complex networks.

In the terminology of machine learning, the conditional probability recovery error is called the “training error,” since it is obtained from the same data set used to build (or “train”) the approximator. In the traditional machine learning literature, the data points generated from the same system that have not been used in the training process can be exploited to validate or test the actual performance of the trained model [39], which in our case is the approximator. As a result, the time series generated from the original complex network system after the approximator is built can be used as the test data set. (Absence of hyper-parameters in the reconstruction process means that cross-validation is unnecessary.) In most cases, the training errors are generally smaller than the test errors, since the training data set is already well-fitted by the model (the approximator) in the training process, while the test data are new and may be out of the fitting range of the current model. Feeding the state configurations of the test data set into the approximator, we calculate the corresponding conditional probabilities using Eq. (2.5) and compare them to the true values. The results reveal a clear advantage of the approximator built from our scheme, i.e., the training and test errors are nearly identical, indicating no over-fitting issues, which are common in various machine learning methodologies [39].

Fig. 6. Distributions of the Conditional Probability Estimation Errors for Various Complex Network Dynamics. The distributions of $\delta = |P[x_i(t+1) = 1|\mathbf{X}_i^R(t)] - P^{\text{est}}[x_i(t+1) = 1|\mathbf{X}_i^R(t)]|$ for the dynamical processes in categories I to V from Table 1 for ER random (green bars) and BA scale-free (blue bars) networks, where $P^{\text{est}}[x_i(t+1) = 1|\mathbf{X}_i^R(t)]$ denotes the conditional probability estimated from the corresponding SDBM approximator through the original state configuration times series, $1 \leq i \leq N$, and $1 \leq t \leq T$.

2.2.5. Model Complexity and Representation Power

As shown in Fig. 6, for dynamical processes in categories I and II, the estimation errors are dominantly distributed at zero. For processes in other categories, besides the dominant peak at zero, there exist multiple small peaks at nonzero error values. These small side peaks are a consequence of the model complexity of the Markov networks.

For dynamical processes in categories I and II, the transition probabilities $P_i^{0 \rightarrow 1}$ all have the form of $1/[1 + \exp(An_i + Bk_i)]$, where A and B are constants, and n_i denotes the number of node i 's neighbors whose states are 1. We have

$$[1 + \exp(An_i + Bk_i)]^{-1} = \{1 + \exp[\sum_{m=1}^{k_i} w_{im}x_m(t) + b_i]\}^{-1},$$

which gives

$$\sum_{m=1}^{k_i} Ax_m(t) + Bk_i = \sum_{m=1}^{k_i} w_{im}x_m(t) + b_i. \quad (2.12)$$

In the ideal case where an absolutely accurate estimate of $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ can be obtained, Eq. (2.12) holds for any possible neighboring state configurations. We thus have $w_{im} = A$ and $b_i = Bk_i$, and the probabilities conditioned on these configurations sharing the same values of n_i in the approximator are all equal to $P_i^{0 \rightarrow 1}$. This means that, in theory, the conditional probabilities can be reconstructed with zero errors. In fact, a one-to-one mapping between the coefficients indicates that the model complexity of the SDBM provides the approximator with sufficient representation power to model the dynamics processes in categories I and II. Practically, since the statistical estimations of $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ may not be absolutely identical under different neighboring configurations with the same values of n_i due to random fluctuations, we have $w_{im} \approx A$ so that the conditional probabilities generated by the approximator may differ from each other slightly and also from the true probability $P_i^{0 \rightarrow 1}$. As a result, random recovery errors can occur.

For dynamical processes in categories other than I and II, the simple coefficient-mapping relation between the corresponding $x_m(t)$ terms on the two sides of Eq. (2.12) become highly nonlinear, due to the fact that the specific forms of $P_i^{0 \rightarrow 1}$ differ dramatically from that of the SDBM. In this case, each particular neighboring state configuration produces a distinct equation. There are 2^{k_i} equations in total, while the number of unknown variables to be solved is only $k_i + 1$ (w_{im} for $m = 1, \dots, k_i$ and b_i). For $k_i \geq 2$, there are thus more equations than the number of unknown variables. As a result, the representation power of the SDBM approximator is limited by its finite model complexity so that, even in theory, the approximator may not be able to fully describe the original dynamical process. In our framework, we calculate the Markov link weights and the node biases according to the $k_i + 1$ most frequently appeared neighboring state configurations. A consequence is that imprecise conditional probability estimations can arise for the configurations with relatively lower occurring frequency, giving rise to the nonzero peaks in the error distribution in Fig. 6. However, interestingly, with respect to the precision of the conditional probabilities produced by the approximator, the SDBM parameters do not show a significantly strong preference towards the most frequently occurring configurations. In practice, the estimated conditional probabilities for the majority of the less frequently occurring configurations also fall within a close range of the true values. This phenomenon suggests the power of the approximator to work beyond the limit set by its theoretical model complexity. This is the main reason for the emergence and persistence of the dominant peak at zero in the error distribution.

2.2.6. Potential for SDBM as a Generative Model?

In real world applications, it is often the case that little information about the details of the dynamical processes is available, i.e., the time series data can be regarded as from a “black box.” Say the data generation mechanism can be uncovered based solely on the observed data, i.e., the

black-box dynamics is simulated to reproduce the “pseudo” time series bearing the same statistical properties of the original data. In machine learning, the model is deemed generative in the sense that it has the same data generating mechanism as that of the original system. Such a generative model captures the essential dynamical and statistical properties of the original system, which can be exploited for long term statistical prediction through simulation. Here we investigate the potential for the SDBM approximator to serve as a generative model of the original dynamics. A basic requirement is that the long-term conditional probability recovery error accumulated at each immediate time step does not drive the model behavior significantly off the trajectory generated by the original dynamical process. Due to the stochastic nature of the original dynamical system and its approximator, starting from a particular state configuration, reproducing similar conditional probabilities in the immediate future does not guarantee a similar state configuration. In fact, the deviation in the configuration at the next time step may lead to further and larger deviations, and so on.

Figure 7 shows three examples: two positive cases where the prediction is reasonable and one negative case where the task fails. For those examples, the approximator is initialized from a random configuration and runs for the same amount of time as the original time series. In the positive cases (SQ-SG and SIS), the conditional probability time series generated by the approximator are visually similar to the original data, indicating that the observed data are qualitatively and quantitatively reproduced by the approximator. In addition, for each node, its conditional probability value $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ can be treated as a random variable, p , whose distributions in the true system should be similar to that produced by the approximator if it can indeed serve as a generative model. To characterize the similarity in the distributions, one could use some standard statistical measures such as the Kullback-Leibler divergence [77], the Jensen-Shannon divergence [78], the

Bhattacharyya distance [79], and the Hellinger distance [80]. However, we find that the performance of these metrics are not stable. It is more effective to directly compare each node's $\langle p \rangle$ and Δp values from the original data and the approximator, e.g., for the three examples shown in Figs. 7(c,f,i). The result is that, except for a few nodes, the values of $\langle p \rangle$ and Δp obtained from the data and SDBM agree with each other for the two positive cases, but there is a substantial difference for the negative case. Our calculation also shows that there are cases where the approximator reproduces similar values of $\langle p \rangle$ and Δp for a substantial fraction of the nodes in the network, but with deviations for the remaining nodes. We conclude that the SDBM approximator can serve as a generative model for some systems. Improvement is necessary to generalize the generative power of SDBM (see Supplementary Note 3 for detailed simulation results for different types of dynamics).

2.3. Discussion

Reconstructing complex dynamical systems from data is a frontier field in network science and engineering with significant applications. Our study was motivated by the question: is it possible to build a universal “machine” to reconstruct, from data only, the underlying complex networked dynamical system and to make predictions? While this paper does not provide a complete solution, significant and (in some cases) striking results are obtained, which give strong credence that such a machine may eventually be possible. In particular, we combine compressive sensing and machine learning to propose two concepts: universal network structural estimator and dynamics approximator. Universality is fundamentally possible due to the fact that many dynamical processes on complex networks are of the Markov type and the interactions among nodes are local. As a result, utilizing basic tools from machine learning and statistical physics, one can build up an energy based Markov network model (e.g., a sparse dynamical Boltzmann machine), to construct a universal estimator and approximator for different types of complex network structures

Fig. 7. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using the Self-Generated State Configuration Time Series. (a,d), (b,e), and (c,f): the conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ time series generated from the original state configuration time series (upper panel) and from the SDBM approximator using the state configuration time series generated by itself (lower panel) for SQ-SG, SIS, and Voter dynamics on ER random networks, respectively. (g,h,i) The mean and standard deviations of $p = P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ for each node i for all time steps for SQ-SG (g), SIS (h), and Voter dynamics (i), respectively, on ER random networks, calculated from the original system (red circles) and the approximator (blue squares), where $1 \leq i \leq N$ (the x-axis).

and dynamics. For a large number of representative dynamical processes studied in this paper, we demonstrate that such a universal SDBM can be reconstructed based on compressive sensing and the scheme of K-means using data only, without requiring any extra information about the network structure or the dynamical process. The working of the SDBM is demonstrated using a large number of combinations of the network structure and dynamics, including many real world networks and classic evolutionary game dynamics. An SDBM with its parameters given by the equations constructed from the times series along with the estimated network structure is able to reproduce the conditional probabilities quantitatively and, accordingly, it is capable of predicting the state configuration at least in a short term. We demonstrate that, for certain types of dynamics, the approximator can reproduce the dynamical process statistically, indicating the potential of serving as a generative model for long term prediction in such cases. Our work adopts binary dynamics, while in principle, the methodology is not confined by such a restriction. Eq. 2.4 can be easily extended to the conditional probability of each possible state value λ_j , that is, $P[x_i(t+1) = \lambda_j | \mathbf{X}_i^R(t)] = P[x_i(t+1) = \lambda_j, \mathbf{X}_i^R(t)] / \sum_s P[x_i(t+1) = \lambda_s, \mathbf{X}_i^R(t)]$. But the configuration space of system states grows exponentially with the choices of λ_j , so that given fine amount of data, the number of time instances corresponding to each particular configuration encounters exponential drop, and thus, the estimation precision of the conditional probability would drop due to the lack of samples. This poses higher requirements for data amount and computational power.

Our effort represents an initial attempt of realizing a universal estimator and approximator, and the performance of our method is quite competitive in comparison with the existing reconstruction schemes designed for specific types of dynamics, such as the compressive sensing technique for evolutionary game and epidemic spreading dynamics [29–31, 37, 38], as well as methods adopt-

ing the Bethe approximation, pseudo-likelihood, mean-field theory, and decimation operation for static and kinetic inverse Ising problems [11–22]. In realistic applications, the data obtained may be discontinuous or incomplete. In such cases, the short-term predictive power possessed by the universal estimator and approximator can be exploited to overcome the difficulty of missing data, as the Markov network nature of the SDBM makes backward inference possible so that the system configurations during the time periods of missing data may be inferred. When long term prediction is possible, the universal approximator has the critical capability of simulating the system behavior and predicting the chance that the system state enters into a global absorption phase, which may find significant applications such as disaster early warning. Another interesting reverse-engineering problem lies in the mapping between the original dynamics and the corresponding parameter value distribution of the reconstructed SDBM. That is, a certain parameter distribution of the SDBM may indicate a specific type of the original dynamics. As such, the correspondence can be used for precisely identifying nonlinear and complex networked dynamical systems. It is also possible to assess the relative importance of the nodes and links in a complex network based on their corresponding biases and weights in the reconstructed SDBM for controlling the network dynamics. These advantages justify our idea of developing a universal machine for data-based reverse engineering of complex networked dynamical systems, calling for future efforts in this emerging direction to further develop and perfect the universal network structural estimator and dynamics approximator.

2.4. Materials and Methods

2.4.1. Structural Estimation Details.

For each type of dynamics, 10 different network realizations are generated for both BA scale-free and ER random topologies. For any node in a particular network realization, the corresponding connection weight vector $\mathbf{V}_{N \times 1}$ is obtained by averaging over 100 compressive sensing implemen-

tations. The elements corresponding to the existent nodes are distinguished from the non-existent links by feeding the averaged $\mathbf{V}_{N \times 1}$ into the K-means algorithm (see below), and the unweighted adjacency matrix is then obtained.

2.4.2. Conflict Resolution.

Since the links are bidirectional, the node on each side provides a weighted solution of the same link. The two solutions may be quite different, giving contradictory indication of the existence of the link and resulting in an asymmetric adjacency matrix. For majority types of dynamical processes on complex networks, compressive sensing almost always gives higher prediction accuracy for lower degree nodes due to the higher connection sparsity, which holds true especially for nodes with their degrees smaller than the network average. Based on this fact, when encountering contradictory solutions, we determine the link existence by examining the lower degree side if the degree value is equal to or smaller than the network average. For cases where the degrees of both sides are larger than the network average, we find that the false negative rates are often high. This is because compressive sensing tends to give over simplified results such as inducing a more than necessary level of sparsity to the solution or causing large fluctuations with the non-existent links so that they are mixed up with the existent ones. As a result, contradictory solutions are treated as positive (existent) solutions. For two types of dynamics, the SDBM dynamics and link-update voter dynamics, or dynamics on real-world networks, treating all contradictory solutions this way, regardless of degree values, can improve the reconstruction precision, albeit only slightly in some cases.

2.4.3. Implementation of the K-Means Algorithm.

K-means is one of the simplest and most popular clustering algorithms and has been widely used for unsupervised clustering tasks [39]. It provides each data example a hard assignment to a cluster within which the data examples are relatively close (or similar) between each other while

Fig. 8. A Schematic Illustration of the K-Means Algorithm for a 2-Dimensional Data Clustering Task. (a) The data points (solid blue circles) in a 2-D feature space to be clustered. It can be visually distinguished that there are three clusters, thus $K = 3$. The K-means algorithm will assign each data point into one of the three clusters. (b) Randomly guess 3 cluster center locations (aqua, green, and red hollow circles), and each data point finds out which center it's closet to. (c) The 2-D space is divided into three regions by three decision boundaries (black dashed lines), each containing the corresponding data points whose closest center is within this region. The data points currently in the regions with the corresponding aqua, green, and red centers are assigned to the aqua, green, and red clusters, respectively. At this stage, data points may be assigned to a cluster that it visually does not belong to. (d) Each center moves to the centroid of the data points currently assigned to it. (e) The updated cluster assignments of the data points are obtained according to the new center locations (movements are shown by the black arrows). The steps in (c) and (d) are repeated until convergence. (f) The final cluster assignments.

Different from this illustration, the compressive sensing solutions of the link weights w_{ij} are 1-dimensional data points, which form two clusters in 1-D space, corresponding to the existent and non-existent links, respectively. (We use the 2-D case to illustrate the K-means algorithm instead of 1-D, due to the representation difficulty in the 1-D

Fig. 9. An Example of Our Proposed Degree Guided Solution Substitution Operation and Final Equation Group Construction. The target node 1 has only two neighbors, nodes 2 and 3. Without loss of generality, we set $k_3 < k_1 = 2 < k_2$ so that the link weight between nodes 1 and 3, w_{13} , is already obtained. The quantity $P_s^{\text{MB}} = \ln(1/\langle x_1(t+1) \rangle_s - 1)$ ($s = 1, \dots, 4$, as $2^{k_1} = 4$ possible (x_2, x_3) configurations exist in total) denotes the estimated probability for $x_1 = 1$ under configuration s at the next time step. The occurring probabilities of the corresponding (x_2, x_3) configurations are listed as percentages in the left column of each panel. **(a)** All the four (x_2, x_3) configurations are shown by the first two columns of the 4×3 matrix in the equation. **(b)** The contribution of the known w_{13} and the configurations of x_3 as marked in **(a)** are moved to the left-hand side of the equation. **(c)** In order to solve for $(w_{12}, b_1)^T$, the two most precise P_s^{MB} values ($s = 2$ and 4) corresponding to the most frequent (x_2, x_3) configurations from **(b)** are chosen to build a linear 2-equation group. However, the 2×2 matrix on the right-hand side does not have full rank and, hence, this equation group is ill defined **(d)** Configurations $s = 1$ and 2 are selected to construct a 2-equation group with a full rank 2×2 matrix on the right-hand side with relatively precise probability estimations, and this equation group is used to solve $(w_{12}, b_1)^T$.

distant from ones in other clusters. The main steps for solving a typical 2-dimensional data clustering problem via K-means are schematically presented in Fig. 8. For each vector $\mathbf{V}_{N \times 1}$ obtained from averaging over multiple applications of compressive sensing, picking out the elements corresponding to the existent links from the non-existent ones with their fluctuating element values is equivalent to a one-dimensional clustering problems with two target clusters, which is one dimension lower than the case shown in Fig. 8. When implementing the K-means algorithm, we choose the initial cluster center positions for the two clusters as the maximum and minimum values of the vector elements with the bias b_i excluded. This is justified because the compressive sensing solution of b_i can have an overwhelmingly high absolute value, which does not provide any structural information but can severely disrupt the clustering process, as demonstrated in Figs. 2(d,f). Due to the sparsity of complex networks, the cluster with smaller number of components is regarded as containing the solutions of the existent links, and the components of the other one correspond to the non-existent links.

2.4.4. Link Weights and Node Biases Estimation.

For each node i , we rank the occurrence of all the existing configurations of its neighbor's states. Among the top ones, $k_i + 1$ are selected to ensure that the coefficient matrix $\Theta_{(k_i+1) \times (k_i+1)}^{\text{MB}}$ on the right hand side of Eq. (2.11) has full rank so that the solutions, $w_{im_1}, w_{im_2}, \dots, w_{im_{k_i}}, b_i$, are unique (the selected configurations are not necessarily on the exact top, since the real top $k_i + 1$ ones may not provide a full rank coefficient matrix).

Due to insufficient number of samples or some particular features of a specific dynamical process, $\langle x_i(t+1) \rangle$, the statistical estimation of $P[x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)]$ (or $P[x_i(t+1)|\mathbf{X}_i^{\text{R}}(t)]$), can be 0 or 1, which respectively makes $1/\langle x_i(t+1) \rangle$ or $\ln[1 - 1/\langle x_i(t+1) \rangle]$ in Eqs. (2.9) and (2.11) diverge. Without loss of generality, we set $\langle x_i(t+1) \rangle = \epsilon$ (or $\langle x_i(t+1) \rangle = 1 - \epsilon$) if $\langle x_i(t+1) \rangle = 0$

(or 1), where ϵ is a small positive constant.

For node i of degree k_i , the total number of the possible neighboring state configurations is 2^{k_i} , so large degree nodes have tremendously more configurations. The process may then regard every particular type of configurations as useful and accordingly lead to inaccurate estimate of $P[x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)]$. Our computations show that properly setting a tolerance in the Hamming distance that allows similar but not exactly identical configurations to be treated as the same $\mathbf{X}_i^{\text{MB}}(t)$ can improve the estimation performance. For a particular neighboring state configuration with too fewer identical matches in the observed data, configurations with difference in one or more digits in the Hamming distance are used instead, until a sufficient number of matches are found. For nodes of degrees larger than, say 15, the Hamming distance tolerances within 2 or 3 usually lead to a reasonable number of matches for estimating $P[x_i(t+1)|\mathbf{X}_i^{\text{MB}}(t)]$. Extensive calculation shows that the small inaccuracy has little effect on the reconstruction accuracy.

2.4.5. Degree Guided Solution Substitution Operation.

For node i 's neighbors, the occurrence frequency of a particular state configuration is generally much higher for smaller values of k_i . Accordingly, the probability estimation conditioned on this configuration can be more accurate for smaller values of k_i . More precise conditional probabilities in turn lead to a higher estimation accuracy of link weights and node biases. Similar to resolving the solution contradiction in the structural reconstruction step, for a pair of linked nodes, i and j of different degrees, w_{ij} solved from the equation group of node i is likely to be more accurate than w_{ji} obtained via node j , if $k_i < k_j$, even they should have the identical values in an ideal case. As a result, using the solution obtained from a lower degree node as the value of the link weight provides a better estimation. We run the calculation for all nodes in the network one by one in a degree-increasing order so that the link weights of the smallest degree nodes are acquired earlier

than other nodes. For the larger degree nodes, the link weights shared with the smaller degree nodes are substituted by the previously obtained values, which are treated as known constants instead of variables waiting to be solved. This operation effectively removes the contribution of the lower degree neighbors from the equation and so reduces the number of unknown variables. A reduced equation group in a form similar to Eq. (2.11) can be built up based on the most frequently occurring state configurations of the remaining neighbors. When the full-rank condition is met, maximum possible precision of the remaining unknown link weights can be achieved. Consequently, with this substitution operation, the k_i link weights of node i , which can be inaccurate if solved from the original $k_i + 1$ -equation group, can be estimated with the maximum possible accuracy. An example of this substitution operation and the final equation group construction process is shown in Fig. 9.

Our calculations show that the application of the degree guided solution substitution operation can significantly increase the accuracy of the estimation of the link weights and nodes biases. This operation is thus key to making the SDBM correspondence a reliable approximator of the original network dynamics.

3 . RECOVERING INTERDEPENDENT STRUCTURE OF MULTI-LAYER COMPLEX NETWORKS

3.1. Introduction

Functional, structural, or information interdependencies are ubiquitous within and between various real-world social, economic, technological, biological, and ecological systems, such as the functional interdependence between the Internet and the power grids, the capital transition between the real economy and the financial institutions, and the preyer-predator relation between different animal species. These interdependencies usually serve as interaction media between systems of different features or natures and may play critical roles in determining the properties in dynamical synchronization [81–84], diffusion and spreading processes [85–89], evolutionary game dynamics [90–93], and the stability or robustness against propagation of malfunction or burst of disasters [94–100]. Since the interaction fashion between the components within a system in most cases differ dramatically with that between multiple systems, the interrelation between different systems is usually subtle and thus hard to extract directly from experimental data. It is hence of crucial importance and urgent need to develop an interaction structure detection scheme that works for various types of independent multiplex infrastructures with high success rate.

In a variety of fields in science and engineering, complex networked infrastructures interdependent with each other are generally modeled via multi-layer complex networks with certain types of interactions between the layers. Previous works on structure recovery mainly focus on complex networked systems whose interior interactions are in an identical form, i.e., the same nodal dynamics is applied to each element of the system [3–10, 23–31, 35–38], and thus, effectively only one single layer exists, while this work is aiming at uncovering not just the inner-layer topologies with the same nodal dynamics but more importantly the interdependent connections that correspond to

a completely different interaction form, given only the state time series data of the system components. We develop a two-stage compressive sensing [42, 43, 101, 102] based approach: (1) the multi-layer system is broken down to separate layers, and the topology is solved layer by layer via the first stage compressive sensing; (2) the components that may have interdependencies with other layers are detected, and the second stage compressive sensing is implemented to determine the inter-connections between these bridge-like components. This breaking-down treatment enables us to deal with systems with large sizes, which is too costly to solve directly via compressive sensing all at once. Without loss of generality, two representative types of inner-layer dynamical processes, the evolutionary game dynamics and the nonlinear oscillator dynamics, respectively in discrete and continuous domains, are adopted in our work to examine the effectiveness and robustness of our approach. Extensive calculation show our method is capable of coping with various modeled and realistic multi-layer complex networks with excellent topology reconstruction and parameter estimation performances, even under noisy environment.

3.2. Results

For the discrete dynamical process, our work consider the classical prisoner's dilemma game (PDG) as an concrete representative of evolutionary game models [20–23, 105]. In the evolutionary PDG model on a single-layer complex network, each individual plays the game with all its nearest neighbors at each time instance, while the payoff, i.e., the characterization of an individual's profit or gain, is calculated at the beginning of the time instance for the individual. Each individual may choose either to cooperate, C, or to defect, D, in any one encounter. If both players choose C, both get a payoff of R ; if one defects while the other cooperates, D gets T , while C gets S ; if both defect, both get P , where $T > R > P > S$ in classic PDG. This setting renders the following payoff

matrix

$$\mathbf{H} = \begin{pmatrix} R & S \\ T & P \end{pmatrix} \quad (3.1)$$

where the above parameters are chosen to be $R = 1$, $T = b > 1$, and $S = P = 0$ in our model [20], and the model is easily changed into another type of game by simply adopting different parameter settings. The total payoff of an individual i at time instance t is the sum of the payoffs gained in its two-player games with all its neighboring players, i.e., $U_i(t) = \sum_{j \in \Omega_i} \mathbf{S}_i^T(t) \cdot \mathbf{H} \cdot \mathbf{S}_j(t)$, where the summation is over the neighborhood of i , Ω_i ; the 2 by 1 vectors $\mathbf{S}_i(t)$ and $\mathbf{S}_j(t)$ stand for the current strategies adopted by individuals i and j , and $\mathbf{S}(t) = (1, 0)^T$ and $(0, 1)^T$ denotes cooperative and defective strategies, respectively. At the strategy updating stage right after payoff calculation, individual i randomly selects a neighbor j and imitates its strategy of the current time instance with probability $P_{i \rightarrow j} = \{1 + \exp[-(U_j(t) - U_i(t))/\kappa]\}^{-1}$, where κ is the “rationality” of the individuals, characterizing the uncertainties in assessing the best strategy.

We define $N - 1$ dimensional vector $\mathbf{A}_i = (a_{i1}, \dots, a_{i,i-1}, a_{i,i+1}, \dots, a_{iN})^T$ as the connection vector of node i in the network, where each element $a_{ij} = 1$ (or 0) indicates an existent (or non-existent) connection between i and j for any j satisfying $1 \leq j \leq N$ and $j \neq i$. Accordingly, the payoff of individual i can be written in the following form: $U_i(t) = \sum_{j=1, j \neq i}^N a_{ij} \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_j(t) = \mathbf{V}_i(t) \cdot \mathbf{A}_i$, where $\mathbf{V}_i(t) = [\mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_1(t), \dots, \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_{i-1}(t), \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_{i+1}(t), \dots, \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_N(t)]$ is a $N - 1$ dimensional vector. While acquiring the information in multiple time instances, we have $\mathbf{U}_i = \mathbf{W}_i \cdot \mathbf{A}_i$, where $\mathbf{U}_i = [U_i(t_1), U_i(t_2), \dots, U_i(t_l)]^T$ is the payoff vector, and $\mathbf{W}_i$ is a l by $N - 1$ matrix with each row equal to vector $\mathbf{V}_i(t_s)$ ($1 \leq s \leq l$). This form well suits the framework of the compressive sensing technique, and the connection vector $\mathbf{A}_i$ and accordingly the network topology can be recovered with high precision [31].

Fig. 10. Schematic Illustrations of a Multi-Layer, Interdependent Network and the Compressing Sensing Approach of Solving the Interdependent Connection Structure. (a) A 3-layer interdependent network with 8 nodes in each layer. Nodes i_{11} and i_{12} are the bridge nodes in layer 1; nodes i_{21} , i_{22} , and i_{23} are the bridge nodes in layer 2; and nodes i_{31} , i_{32} , and i_{33} are the bridge nodes in layer 3. The interdependent node pairs are i_{11} and i_{32} , i_{12} and i_{22} , i_{21} and i_{31} , and i_{23} and i_{33} . (b) A schematic illustration of Eq. 3.2. The numbers of rows and columns only have schematic meaning. The number of time instances selected (the number of rows of $\hat{\mathbf{U}}_i$ and $\mathbf{W}'_i$) is usually much smaller than the length of vector in a typical compressive sensing setting. The colored parts of matrix $\mathbf{W}'_i$ and vector $\mathbf{A}'_i$ corresponds to the inner-layer connections of layer m , while the black-and-white parts corresponds to the interdependent connections between layer m and the bridge nodes in all other layers. Using (a) as an example, if $m = 2$ and $i = i_{21}$, then the black-and-white components in (b) are corresponding to the possible interdependencies between node $i = i_{21}$ and nodes $q = i_{11}$, i_{12} , i_{31} , i_{32} , and i_{33} .

For games taking place in a multi-layer interdependent structure instead of a single layer network, as shown in Fig. 29(a), an individual interacts with not just its neighbors within the layer it is located in but also with an interdependent node in another layer, if exists, and it is the inter-layer interactions between these node pairs that introduces the interdependency between different layers. As an analogy of the widely existent diversified investment and hedge funding behavior in the real-world, a player may simultaneously devote its effort to multiple fields, such as stock market, futures, or real estate, characterized via the interdependent individuals in multiple network layers. Since the funds transfer among the participating fields for balancing the gains or hedging risks is common in reality, we devise a simple but representative mechanism to capture the phenomenon: partial payoff exchange between the interdependent node pairs, that is, node i transfers a part of its original payoff $U_i^{m_1}(t)$ obtained solely within its home layer m_1 , $\eta_{iq} \cdot U_i^{m_1}(t)$ ($0 < \eta_{iq} < 1$), to its interdependent partner q in layer m_2 , while receiving $\eta_{iq} \cdot U_q^{m_2}(t)$ from q , where $U_q^{m_2}(t)$ is q 's original payoff gained within its home layer m_2 , and the final payoffs of nodes i and q are $\hat{U}_i^{m_1}(t) = (1 - \eta_{iq})U_i^{m_1}(t) + \eta_{iq}U_q^{m_2}(t)$ and $\hat{U}_q^{m_2}(t) = (1 - \eta_{iq})U_q^{m_2}(t) + \eta_{iq}U_i^{m_1}(t)$, respectively.

If the compressive sensing based method for single layer network dynamics [31] is adopted to detect the structure of a particular layer among the multiplex, while $\hat{U}_i^{m_1}(t)$ is treated as $U_i^{m_1}(t)$ for each node i , the nodes with interdependent partners can be regarded as being affected by an external hidden source. In previous models with external hidden attributes, the compressive sensing solutions of them may fail in the sense of lacking sparsity and displaying strikingly higher standard deviations compared with the other nodes [35, 36]. Thus, in our model, the bridge nodes, namely, the nodes with interdependencies in other layers, may also be identified via analyzing their standard deviations of their corresponding compressive sensing solutions.

With the identified bridge nodes, we propose a method of extracting their detailed interdepen-

dencies purely from the state and payoff time series data. For a bridge node i in layer m of the multiplex with M layers in total, its final payoff can be expressed in the following form

$$\begin{aligned}\hat{U}_i^m(t) &= (1 - \eta_{iq})U_i^m(t) + \eta_{iq}U_q^*(t) \\ &= (1 - \eta_{iq}) \sum_{j=1, j \neq i}^N a_{ij} \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_j(t) + \sum_{q=1}^{\tilde{N}_m} c_{iq} \eta_{iq} U_q^*(t) = \mathbf{V}'_i(t) \cdot \mathbf{A}'_i,\end{aligned}\quad (3.2)$$

where

$$\begin{aligned}\mathbf{V}'_i(t) = [& \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_1(t), \dots, \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_{i-1}(t), \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_{i+1}(t), \dots, \mathbf{S}_i^T(t) \mathbf{H} \mathbf{S}_N(t), \\ & U_1^*(t), \dots, U_q^*(t), \dots, U_{\tilde{N}_m}^*]\end{aligned}\quad (3.3)$$

and

$$\begin{aligned}\mathbf{A}'_i = [& (1 - \eta_{iq})a_{i1}, \dots, (1 - \eta_{iq})a_{i,i-1}, (1 - \eta_{iq})a_{i,i+1}, \dots, (1 - \eta_{iq})a_{iN}, \\ & \eta_{i1}c_{i1}, \dots, \eta_{iq}c_{iq}, \dots, \eta_{i\tilde{N}_m}c_{i\tilde{N}_m}]^T;\end{aligned}\quad (3.4)$$

$\tilde{N}_m$ denotes the total number of bridge nodes in all layers except for layer m ; U_q^* stands for the payoff for a bridge node q in all layers other than layer m . By this means, for l time instances, the matrix $\mathbf{W}'_i$ formed via vertically stacking the row vectors $\mathbf{V}'_i(t_1), \dots, \mathbf{V}'_i(t_l)$ satisfies

$$\hat{\mathbf{U}}_i = \mathbf{W}'_i \cdot \mathbf{A}'_i, \quad (3.5)$$

where $\hat{\mathbf{U}}_i = [\hat{U}_i(t_1), \dots, \hat{U}_i(t_l)]$. Apparently, this form allows $\mathbf{A}'_i$, accordingly the interdependent connection structure between i and the bridge nodes in all other layers along with the payoff exchange rates η_{iq} , to be simultaneously recovered via compressive sensing, as shown schematically in Fig. 29(b).

Similar approach can also be adopted to systems with continuous state dynamical processes, such as the various types of oscillator systems.

This two-stage compressive sensing based approach enables us to reconstruct inner-layer and inter-layer topology of large size multi-layer infrastructures. The previous compressive sensing based reconstruction approach encounters difficulty in systems with large number of nodes, while for a large size multi-layer system, our approach can firstly break it down into different single layers, whose topology can be solved one by one, and then with the estimated structures of each layer, the interdependencies between the layers are recovered in the second stage. (The detailed procedures of our complete approach are presented in the Method section.)

As a concrete example, our approach is implemented on a three-layer multiplex network system with evolutionary game dynamics. As shown in Fig. 3.2(a1,a2,a3), it is observed that the standard deviations σ of the ordinary nodes are in the order of 10^{-9} , while for the bridge nodes, we generally have $\sigma > 10^{-3}$, which is over 6 orders of magnitude higher than the ordinary ones. This provides a wide range to choose the standard deviation threshold σ_c for bridge node identification, and our extensive calculations show that full identification with zero false positive rate can be achieved as long as enough number of time instances are deployed. The detection of the inter-layer connections between the identified bridge nodes is a challenging task, since simple correlation measurement between a bridge node and its interdependent partner, such as the linear correlation coefficient, does not show any distinguishable behavior against that between the bridge node and all other bridge nodes in different layers, indicating the need for more sophisticated approaches. The compressive sensing solutions $\mathbf{A}'_i$ for every ordinary or bridge node i in each layer is displayed in Fig. 3.2(b1,b2,b3). Each entry position in the augmented part corresponds to a particular bridge node in another layer. Only bridge nodes have nonzero entries in the augmented part of $\mathbf{A}'_i$ and less than one but above zero entries in the non-augmented part. The presented results show that the nodes with less than 1 (but above zero) non-augmented entries and nonzero augmented entries

Fig. 11. Bridge Node Identification and Interdependencies Reconstruction. (a1,a2,a3) The standard deviations of the compressive sensing solution $\mathbf{A}_i$ for each node i within layer 1 (a1), 2 (a2), layer 3 (a3), respectively. The interdependent multiplex structure adopted in this example contains 3 layers, each of which is built in ER-random topology with $N_m = 50$ nodes ($m = 1, 2$, and 3) and average degree $\langle k \rangle = 4$. The x-axis denotes the node index, and the y-axis denotes the standard deviation σ for each corresponding node. The threshold $\sigma_c = 10^{-3}$ is used in our calculation, and the light purple bars marks identified the bridge nodes (BN). (b1,b2,b3) The colormap of the augmented connection vector $\mathbf{A}'_i$ for each node i in layer 1 (a1), 2 (a2), and 3 (a3). The x-axis show the index i , and the entry in the j 's (or the q 's) row and the i 's column stands for the connection between node i and j (or q), where j ($1 \leq j \leq N_m$) denotes an inner-layer node, and q ($q > N_m$) denotes an bridge node in another layer (below the dashed line). The dark blue back ground stands for the non-existent connections, while the all other colored entries implies an existent connection. Only the bridge nodes have nonzero entries $\eta_{iq}c_{iq}$ ($q > N_m$) in the augmented part, while in the non-augmented part, a nonzero entry of an ordinary node has the value $a_{ij} = 1$ (dark red), while that of a bridge node has the value $(1 - \eta_{iq})c_{iq} < 1$ (green, yellow, orange, or light red).

all correspond to sharp σ peaks in the same horizontal positions of the corresponding upper panels, indicating the correct bridge node identification. In our model, partial payoff exchange takes place between each pair of interdependent nodes in two different layers, namely, each bridge node has only one external connection, so that the augmented part of $\mathbf{A}'_i$ for a bridge node always display exactly one nonzero entry, and the position of this entry does stand for the correct independent partner of the bridge node, which in turn demonstrates the effectiveness of our interdependency detection method. It is verified that the values of the payoff exchange rate η_{iq} are also estimated in high precision in this example.

The detailed performance of our reconstruction approach under evolutionary game dynamics obtained via extensive numerical calculations under various scenarios are presented in Fig. 3.2. Since evolutionary game dynamics renders dramatically differed dynamical behavior under different in population interaction structures [105], three 2-layer infrastructures under different topology settings are applied in order to examine the adaptivity of our method: (1) both layer are topologically homogeneous; (2) both layer are topologically heterogeneous; (3) one layer with homogeneous topology is interdependent with another heterogeneous layer. The relative error rate ϵ_C , i.e., the ratio between the correct link existence (or non-existence) detections and the total number of existent (or non-existent) links, and the absolute error $e_W = |\eta^{\text{estimation}} - \eta^{\text{true}}|$ between the estimated link weights (the interdependent payoff exchange rates) and their true values can play the role of performance indicators. Generally, perfect or almost perfect reconstruction of both the interdependent connection structure and the payoff exchange rates can be achieved as the normalized number of measurements $R_M = l/N_m \geq 0.4$ in call cases, implying excellent functionality of our method under diverse situations.

Data obtained in reality is easily contaminated by noise so that the capability of extracting

Fig. 12. The Interdependent Connection Structure Recovery and the Payoff Exchange Rate Estimation Performance Under Evolutionary Game Dynamics. (a,b,c) The relative interdependent connection reconstruction error rate ϵ_C for the nonzero terms (“NZ”, i.e., the existent links, shown in purple) and the zero terms (“Z”, i.e., the non-existent links, shown in green) versus the normalized number of measurements $R_M = l/N_m$ under noise level $\mu_N = 0$ (triangles pointing up), 10^{-4} (circles), 10^{-3} (squares), and 10^{-2} (triangles pointing left). (d,e,f) The absolute estimation error $e_W = |\eta^{\text{estimation}} - \eta^{\text{true}}|$ of the interdependent payoff exchange rate η_{iq} averaged over all the interdependent node pairs versus R_M under the four different noise levels as in (a,b,c). The results are obtained on three 2-layer multiplex networks: (a,d) both layers have ER-random topology; (b,e) both layers have BA scale-free topology; (c,f) one layer has ER-random topology, while another one has BA scale-free topology. All layers in all cases are in size $N_m = 100$ with average degree $\langle k \rangle = 4$. 10% randomly selected nodes in each layer are bridge nodes, each of which owns an interdependent partner in another layer, and there are accordingly 10 interdependent links in each multiplex system. Each data point is averaged over 10 network realizations, each with 10 compressive sensing implementations.

valuable information from unprecise data is of crucial importance. Thus, multiple levels of noise is introduced into the payoff of each node in order to test the effectiveness of our structure extraction approach against noise. Figure 3.2 show the reconstruction errors obtained under the relative noise levels from $\mu_N = 10^{-4}$ to 10^{-2} in the sense that node i 's payoff is changed to $(1 \pm \mu_N) \cdot \hat{U}_i^m$, where $+$ and $-$ appear with even chance, and $\mu_N = 0$ indicates the case without noise. It is observed in Fig. 3.2 that in all cases, the increasing levels of noise do not cause obvious increment in the estimation errors, and the curves corresponding to different noise levels almost over lap with each other, indicating the robustness of our method against noise contamination.

Complex networks existing in reality may differ dramatically with modeled networks in the aspect of structural properties, such as the wide existence of some special motifs that may drive some reconstruction schemes to failure. To test the performance of our method on real-world complex networks, we construct 6 multi-layer structures of 2 to 3 layers with each layer selected from 11 real-world networks in four categories, including social networks, food webs, protein interaction networks, and logic circuits. For 4 out of the 6 multiplex structures, each are built with networks in the same category, while the other two are mixtures consisting layers from multiple categories. As presented in Table 7, highly accurate independency and parameter estimation is achieved in all cases, indicating the effectiveness of our compressive sensing based approach in not only the modeled systems but also the realistic structures.

3.3. Discussion

To sum up, in this work, we developed a two-stage compressive sensing based method, which is able to successfully reconstruct the inner-layer and inter-layer connection structures and estimate the relevant interaction parameters with high accuracy for multiplex systems consisted of both modeled and real-world complex network layers by using clean or noise contaminated time series

TABLE 5

The Interdependency Reconstruction and Parameter Estimation Performance on 6 Multi-Layer Networks Structured Via 11 Real-World Complex Networks. Four categories of real-world networks are adopted: (1) three social networks [4, 5], (2) three food webs [9, 10], (3) three protein interaction networks [5], and (4) two logic circuits [6]. $10\% \times N_{\max}$ randomly selected nodes in each layer are chosen as bridge nodes, where $N_{\max}$ denote the total number of nodes in the layer with the largest size. The relative error rate ϵ_C of detecting interdependent connections for nonzero terms (NZ, the existent links) and zero terms (Z, the non-existent links) and the parameter estimation error e_W for both evolutionary game dynamics and nonlinear oscillator dynamics are presented in the table. $R_M = 60\%$ is adopted for all calculations.

Network type	Social	Food web	Protein interaction
Layer 1	St Martin ($N = 45$)	Seagrass ($N = 49$)	1eawInter ($N = 53$)
Layer 2	Consulting ($N = 46$)	Grassland ($N = 88$)	1a4jInter ($N = 63$)
Layer 3	Prison inmate ($N = 67$)	Ythan ($N = 135$)	1aorInter ($N = 97$)
ϵ_C (NZ)	0	0	0
ϵ_C (Z)	0	0	0
e_W	1.1×10^{-7}	6.4×10^{-8}	2.2×10^{-8}

Protein interaction	Logic circuit	Mixture 1	Mixture 2
Layer 1	S208 ($N = 122$)	St Martin ($N = 45$)	Ythan ($N = 135$)
Layer 2	S420 ($N = 252$)	Seagrass ($N = 49$)	1aorInter ($N = 97$)
Layer 3	*	1eawInter ($N = 53$)	S420 ($N = 252$)
ϵ_C (NZ)	0	0	0
ϵ_C (Z)	0	0	0
e_W	4.9×10^{-9}	1.5×10^{-8}	1.1×10^{-8}

data. The layer-wise treatment enables the method to cope with very large systems, which is hard to directly implement the one-stage compressive sensing for the entire infrastructure as in previous works. This contribution presents a concrete trial that helps to fulfill the urgent requirement of interdependency uncovering in a variety of realistic social, economic, financial, technological, biological, or ecological systems and facilitates further purposes, such as dynamical process control and behavior prediction.

3.4. Methods

Inner-layer and inter-layer connection structure reconstruction procedures for systems with evolutionary game dynamics. Given the time series of each node's state and payoff and the information on which layer each node belongs to, the procedures of recovering the topology within each layer and the interdependent connection structure between different layers in a multi-layer complex networked system are as following: (1) Reconstruct the network topology of each layer via the compressive sensing technique with the state $\mathbf{S}(t)$ and payoff $\hat{U}(t)$ time series data corresponding to the layer; (2) Identify the bridge nodes by comparing the standard deviation σ_i of the compressive sensing solution $\mathbf{A}_i$ for each node i and picking out the ones with substantially higher magnitude of σ ; (3) Calculate the inner-layer payoff $U_i^m(t)$ of all the bridge nodes in each layer by using the state time series and the inner-layer connection structure recovered in (1) (the connections of a bridge node is obtained via its immediate neighbors, which overcomes the difficulty in extracting the connection information due to the high variance of its compressive sensing solution); (4) For the bridge i in layer m , build the augmented base matrix $\mathbf{W}_i'$ with the inner-layer payoff $U_q^*(t)$ of all layers other than m and the payoff vector $\hat{\mathbf{U}}_i$ by randomly selecting l out of the L time instances; (5) Recover the augmented connection vector $\mathbf{A}_i'$ in Eq. 3.2 via compressive sensing for each bridge node i in each layer m and obtain interdependencies c_{iq} between the bridge nodes by identifying

the non-zero terms in the augmented part of each $\mathbf{A}'_i$; (6) The interdependent payoff exchange rate η_{iq} can be obtained not only by recognizing the nonzero terms $\eta_{iq}c_{iq}$ in the augmented part of $\mathbf{A}'_i$ as η_{iq} due to $c_{iq} = 1$ but also by solving it from the nonzero entries $(1 - \eta_{iq})a_{ij}$ in the non-augmented part, and in our results, the average of the two approaches is taken as the estimated value of η_{iq} .

4 . THE PARADOX OF CONTROLLING COMPLEX NETWORKS: CONTROL INPUTS VERSUS ENERGY REQUIREMENT

4.1. Introduction

The past fifteen years have witnessed tremendous advances in our understanding of complex networked structures in various natural, social, and technological systems, as well as the dynamical processes taking place on them [3, 112–126]. The significant issue of control arises naturally, but this remains to be outstanding and extremely challenging, since nonlinear dynamical processes generally take place on complex networks. Control of nonlinear dynamics, especially when chaos is present, can be done but only for low-dimensional systems [127, 128]. Despite the development of nonlinear control methods [129–134] in certain particular situations such as consensus [135], communication [136, 137], traffic [138] and device networks [129, 139], a general framework of controlling complex nonlinear-dynamical networks has yet to be developed. A natural approach is to reduce the problem to controlling complex networks with linear dynamics based on traditional frameworks from control engineering [140–144].

In the past a few years, great progress was made toward understanding the linear controllability of complex networks in terms of the fundamental issue of the minimum number of driver nodes required to steer the whole network system from an arbitrarily initial state to an arbitrarily final state in finite time [2, 2, 145–149]. In particular, Liu et al. successfully adopted the classic structural controllability theory developed by Lin [141] to complex networks of various topologies [2], for which the traditional Kalman’s rank condition [140] is difficult to be applied [145]. The ground breaking results show that, the structural controllability of a directed network can be assessed by using the maximum matching [150–152] algorithm. The effects of the density of in/out degree nodes were incorporated into the structural controllability framework [153], which has also been applied to

protein interaction networks [154]. Recently, based on the classic Popov-Belevitch-Hautus (PBH) rank condition [155] in traditional control engineering, a variant of the structural-controllability theory, the so-called exact controllability framework, was developed [156].

Both the structural- and exact-controllability frameworks consider the standard setting of linear dynamical systems under control input [2, 145, 156]:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \quad (4.1)$$

where $\mathbf{x} = [x_1(t), \dots, x_N(t)]^T$ is the state variable of the whole network system, the vector $\mathbf{u} = [u_1(t), \dots, u_M(t)]^T$ is the control input or the set of control signals, $A = \{a_{ij}\}$ is the $N \times N$ adjacency matrix of the network, and $B = \{b_{ik}\}$ is the $N \times N_D$ control matrix specifying the set of “driver” nodes [2], each receiving a control signal (corresponding to one component of the control vector $\mathbf{u}$). The aim is to determine the minimum number of driver nodes, N_D , for networks of various topologies. However, we have encountered unexpected difficulties in carrying out *actual* control of complex networks by using the minimum set of driver nodes as determined by the controllability frameworks. This concerns effectively the issue of guiding the network system to approach a final state with acceptable proximity error. In particular, given an arbitrary complex network, once N_D is determined, we can calculate the specific control signals by using the standard linear systems theory [157] and apply them at various unmatched nodes. A surprising finding is that, quite often, the actual control of the system is difficult to be achieved computationally in the sense that in any finite time, it is not possible to drive the system from an arbitrary initial state to an arbitrary final state, i.e., the actual state the system finally reaches is unreasonably far from the designated one. This difficulty in realizing actual control, which has not been formerly addressed in any other works, persists for a large number of model and real-world networks, prompting us to study if the developed controllability frameworks can ensure actual control with given finite computational precision

and, more importantly, to consider the issue of control energy.

In this paper, we investigate the issue of control energy in the framework of structural controllability theory. We find that, the energy required to steer a system from a specific initial state to a target state in finite time follows a fat-tail distribution, indicating the existence of extraordinarily high energy requirement. In extreme but not uncommon cases, the energy is practically divergent. This phenomenon signifies the emergence of a *paradox* in controlling complex networks: although a small fraction of driver nodes can guarantee full control of the network system mathematically, the energy required to achieve control is often unbearable. We resolve the paradox by presenting the idea of control chains (not to be confused with control signal paths corresponding to maximum-matching [2]), in which the fat-tail distribution of the energy can be derived as a key structural feature. The theory of control chains enables us to offer simple strategies to significantly reduce the control energy through small augmentation of the number of control signals beyond N_D . In this regard, the quantity N_D , on which the structural controllability theories focus, can effectively be regarded as the lower bound of the actual number of control signals required. To realize actual control of a complex network, it is imperative to find the trade-off between the number of external input signals and feasible energy consumption.

Remark. In Ref. [147], partial theoretical bounds for the control energy were derived. The bounds are partial because, for example, for networks whose lower bounds can be obtained, the upper bounds typically diverge. This property of divergence was puzzling: does it mean that the actual energy required would diverge as well and, if so, can a complex network actually be controlled? The present work was largely motivated by these questions, in which we obtain a detailed understanding of the physically important issue of practical controllability of complex networks through the discovery of a general scaling law for the distribution of the energy required for control. The exis-

tence of control chain is also uncovered, enabling us to articulate practical strategies to significantly reduce the control energy.

4.2. Resolution of Control Paradox

4.2.1. Gramian Matrix and Quantification of Control Energy.

Optimal control of a linear network in the sense that the energy is minimized can be achieved when the input control signals $\mathbf{u}_t$ are chosen as [158]: $\mathbf{u}_t = B^T \cdot e^{A^T(t_f-t)} \cdot W^{-1} \cdot (\mathbf{x}_{t_f} - e^{At_f} \cdot \mathbf{x}_0)$, where

$$W \equiv \int_{t_0}^{t_f} e^{A\tau} B \cdot B^T \cdot e^{A^T\tau} d\tau \quad (4.2)$$

is the Gramian matrix, a positive-definite and symmetric matrix [157], which is the base to determine, quantitatively, if a system is actually controllable. In particular, the system is controllable only when W is nonsingular (invertible) [157, 158]. The energy required through the control input $\mathbf{u}$ is given by [157]

$$E(t_f) = \int_0^{t_f} \mathbf{u}_t^T \mathbf{u}_t dt, \quad (4.3)$$

where control is initiated at $t = 0$. We use the Erdos-Renyi (ER) type of directed random networks [159, 160] and the Barabási-Albert (BA) type of directed scale-free networks [120] with a single parameter P_b . Specifically, for a pair of nodes i and j with a link, the probability that it points from the smaller-degree to the larger-degree nodes is P_b , and $1 - P_b$ is the probability that the link points in the opposite direction (if both nodes have the same degree, the link direction is chosen randomly). (See Supplementary Note 1 for analytical treatment of the in- and out-degree distributions.) To determine the set of driver nodes, we use the maximum-matching algorithm [141], which gives the control matrix B . For each combination of A and B , we first randomly choose the initial and final states. We then calculate the corresponding Gramian matrix W , the input signal $\mathbf{u}_t$, the actual final states $\mathbf{x}_{t_f}^*$ (equation (4.1)), and finally the control energy $E(t_f)$. Repeating this process

for each and every independent network realization in the ensemble entails an extensive statistical analysis of the control process.

4.2.2. Resolution of Control Paradox.

Mathematically, if the Gramian matrix W is singular, the energy diverges. Through extensive and systematic numerical computations, we find that, even when W is non-singular in the mathematical sense, for typical complex networks its condition number can be enormously large [161], making it effectively singular as any physical measurement or actual computation must be associated with a finite precision. Say in a physical experiment the precision of measurement is ε . In a computational implementation of control, ε can be regarded as the computer round-off error. Consider the solution vector $\mathbf{X}$ of the linear equation: $W \cdot \mathbf{X} = \mathbf{Y}$, where $\mathbf{Y}$ is a known vector. Let C_W be the condition number of W . The accuracy of the numerical solution of $\mathbf{X}$, denoted by $e_X = 10^{-k}$ (k is a positive integer), is bounded by the product between C_W and ε [1]. We see that, if C_W is larger than $10^{-k}/\varepsilon \equiv \bar{C}_W$, it is not possible to bring the system to within 10^{-k} of the final state, so control cannot be achieved in finite time.

For a large number of networks drawn from an ensemble of networks with a pre-defined topology, the condition numbers of their Gramian matrices are often orders of magnitude larger than $\bar{C}_W$ (see Supplementary Fig. 1 in Supplementary Note 2.1 for the relation between C_W and e_X). For these networks, not only is the control vector unable to drive the system to the target state, but the associated energy can be extremely large. These observations suggest the following criterion to define controllability in terms of the control energy: a network is controllable with respect to a specific control setting if and only if the condition number of its Gramian matrix is less than $\bar{C}_W$, a critical number determined by both the measurement or computational error and the required precision of control. Quantitatively, for a given set of network parameters (hence a given network ensemble)

and control setting, the probability that the condition number of the Gramian matrix is less than $\bar{C}_W$, $P(\bar{C}_W)$, can effectively serve as a new type of controllability, which we name as *practical controllability*. Increasing the precision of the computation, e.g., by using special simulation packages with round-off error orders of magnitude smaller than that associated with the conventional double-precision computation, would convert a few uncontrollable cases into controllable ones, but vast majority of the uncontrollable cases remain unchanged.

Figures 13a,b show the percentage of driver nodes $n_D \equiv N_D/N$ versus the directional link probability P_b . We see that n_D is minimized for $P_b \approx 0.5$, indicating that, mathematically, only a few control signals are needed to control the whole network, leading to optimal structural controllability. But can practical controllability be achieved in the same parameter regime where the structural controllability is optimized?

Figure 13c show, for the same networks as in Fig. 13a, the measure of control energy, i.e., the probability $P(\bar{C}_W)$, versus the network parameter P_b . We see that, for both regimes of small and large P_b values where the structural controllability is weak (corresponding to relatively high values of n_D in Fig. 13a), the practical controllability is relatively strong. In the regime of small P_b values, most directed links in the network point from small- to large-degree nodes. In this case, the network is more practically controllable, in agreement with intuition. The surprising result is that, in the regime of intermediate P_b values (e.g., P_b around 0.5) where the number of driver nodes to control the whole network is minimized so that the structural controllability is regarded as strong, the practical controllability is in fact quite weak, as the probability of the condition number being small is close to zero. For example, for $\langle k \rangle = 4$, the minimum value of $P(\bar{C}_W)$ is only about 0.1 for $P_b \approx 0.6$; for $\langle k \rangle = 6$ and $\langle k \rangle = 8$, the minimum values are essentially zero. A striking phenomenon is that the minimum value of $P(\bar{C}_W)$ occurs in a wide range of the parameter P_b ,

Fig. 13. Structural and Practical Controllability Measures in Directed Networks. Structural controllability measure n_D versus directional edge probability P_b for (a) ER random networks and (b) BA scale-free networks of size $N = 1000$ and three values of the average degree ($\langle k \rangle = 4, 6$, and 8). The dash-dotted lines represent the results obtained by the cavity method [2, 2], and the squares, triangles, and circles are the simulation results from the maximum matching algorithm [2]. (c, d) Measure of practical controllability $P(\bar{C}_W)$ for ER random and BA scale-free networks of size $N = 100$, respectively, where $P(\bar{C}_W)$ is the probability that the condition number of the Gramian matrix is less than some physically reasonable threshold value versus P_b . Comparing a with c, or b with d, we observe the striking phenomenon that, in the parameter regime where the number of driver nodes is minimized so that the corresponding networks are deemed to be most structurally controllable, they are practically uncontrollable. The phenomenon persists regardless of the network size and type. All nodes are self-loop free. ⁶⁷ The qualitative behavior is robust against the value of $\bar{C}_W$.

e.g., $[0.3, 0.8]$ and $[0.2, 0.9]$ for $\langle k \rangle = 6$ and $\langle k \rangle = 8$, respectively. This indicates that the network is practically uncontrollable for most cases where the structural controllability is deemed to be optimal. The same phenomenon holds for different network sizes (see Supplementary Fig. 2 in Supplementary Note 2.2). Another interesting finding in Fig. 13 is that N_D is symmetric about $P_b = 0.5$. However, the symmetry is broken for $P(\bar{C}_W)$, indicating that there is no simple negative correlation between N_D and $P(\bar{C}_W)$. This prompts us to find more essential structural properties responsible for the smallness of $P(\bar{C}_W)$.

4.3. Distribution of Control Energy and Concept of Control Chains

Suppose the network is practically controllable so that the required control energy is not unrealistically large. For an ensemble of randomly realized network configurations with the same structural properties and for different control settings, the control energy can be regarded as a random variable. What is then its probability distribution? To gain insights, we generate directed networks with different values of $\langle k \rangle$ and P_b . We then implement the maximum matching algorithm [2] to obtain the control matrix B and calculate the minimum energy by using equation (4.3) for final time t_f . For each network, the initial states $\mathbf{x}_0$ and desired final states $\mathbf{x}_{t_f}$ are randomly chosen. The calculation of energy is done only for those networks with condition number smaller than $\bar{C}_W$, and a variety of $\bar{C}_W$ values are adopted. Representative results are shown in Fig. 14, where an algebraic (power-law) scaling behavior with fat tails is observed for all cases with the scaling exponent approximately equal to 1.5. The scaling is robust against various $\bar{C}_W$ values (Figs. 14a,c) and network sizes (see Supplementary Fig. 3 in Supplementary Note 2.3). From Figs. 14a,c, we see that different values of $\bar{C}_W$ result in different groups of practically controllable networks, and the required control energy in general increases with $\bar{C}_W$. In Figs. 14b,d, the value of $\bar{C}_W$ is fixed and the control energy required is larger for larger value of P_b as compared with the case of $P_b = 0$.

Fig. 14. Distributions of Control Energy for Practically Controllable Networks. (a, c) Energy distributions under different values of the threshold condition number $\bar{C}_W$ for $P_b = 0.1$ and $t_f = 1$. $\langle k \rangle = 6$ for random networks (a) and $\langle k \rangle = 8$ for scale-free networks (c). (b, c) Energy distributions for different values of the average degree $\langle k \rangle$ and the directional connection probability P_b under $\bar{C}_W = 10^{12}$ and $t_f = 1$ for ER random and BA scale-free networks, respectively. In all cases, we observe an algebraic (power-law) scaling behavior.

This is intuitively correct as, for $P_b = 0$, all directed links point from small- to large-degree nodes, facilitating control of the whole network.

We develop a physical understanding of the large control energy required and also the algebraic scaling behavior in the energy distribution. To gain insights, we first consider a simple model: an unidirectional, one-dimensional (1D) string network, for which an analytic estimate of the control energy can be obtained (see Supplementary Note 3) as

$$E_l \approx \lambda_{H_l}^{-1}, \quad (4.4)$$

where E_l denotes the energy required to control a 1D string of length l (the number of nodes on the string) and λ_{H_l} is the smallest eigenvalue of the underlying H -matrix, denoted by H_l , which is related to the Gramian matrix by $H \equiv e^{-At_f} W e^{-A^T t_f}$. The condition number of the 1D chain system increases exponentially with its length. For example, the value of C_W of a chain of length larger than 7 has already exceeded $\bar{C}_W = 10^{12}$. This indicates that, even for a simple 1D chain network, the energy required for control tends to increase exponentially with the chain length. Numerical verification of equation (4.4) is presented in Figs. 15a. Although equation (4.4) is obtained for a simple 1D chain network, we find numerically that it holds for random and scale-free network topologies (See Supplementary Fig. 5 in Supplementary Note 4).

In a networked system, control signal and energy originated from the driver nodes travel through 1D-string-like paths towards each of the non-driver nodes. As discussed, identifying maximum matching so that the network is deemed structurally controllable is independent of the control energy. However, when maximum matching is found, we can divide the whole network into N_D control signal paths (CSPs), each being a unidirectional 1D string led by a driver node that passes the control signal onto every node along the path, as illustrated by the vertical paths in Fig. 16a. CSPs thus provide a picture indicating how the signals from the N_D external control inputs reach

Fig. 15. Relationship Among 1D Chain Energy, the Smallest Eigenvalue of H -Matrix, and Network Control Energy. (a) For an 1D chain network of length l , energy E_l and $\lambda_{H_l}^{-1}$ versus l , providing support for the analytic result equation (4.4). (b) Correlation between $\langle E \rangle$, the average of control energy for networks with the same LCC length, and E_L , the energy of a LCC with length $D_C = L$ ($L = 3, 4, 5, 6$, and 7 for ER and $L = 3, 4, 5$, and 6 for BA networks), calculated from ensembles of 10000 networks.

every node in the network to ensure full control (in the sense of structural controllability).

We can distinguish two types of links: one along and another between the CSPs, as shown in Fig. 16a. It may seem that the latter class are less important as the control signal and energy flow along the former set of links. However, due to coupling, a node's dynamical state will affect all its nearest neighbors' states which, in turn, will affect the states of their neighbors, so on, and vice versa. In principle, any driver node connects with nodes both along and outside its CSP. Correspondingly, an arbitrary node in the network is influenced by every driver node, directly through the CSP to which it belongs, or indirectly through the CSPs that it does not sit on. Intuitively, the ability of a driver node to influence a node becomes weaker as the distance between them is increased. In order to control a distant node, exponentially increased energy from the driver is needed. The chain starting from a driver node and ending at a non-driver node along their shortest path is effectively a control chain. In an intuitive picture, control energy flows through control chains, while control signal is transported via CSPs. We can define the length of the longest control chain (LCC), D_C , as the *control diameter* of the network, as shown in Fig. 16b.

There can be multiple LCCs. The node at the end of a LCC is most difficult to be controlled in the sense that the largest amount of control energy is required. The number m of such end nodes dictates the degeneracy (multiplicity) of LCCs. An example is shown in Fig. 16b, where we see that, although there can be multiple LCCs, the ends of them converge to only three nodes, leading to $m = 3$. Since the energy required to control a 1D chain grows exponentially with its length in such a way that even one unit of increase in the length can amplify the energy by several orders of magnitude (Fig. 15a), the energy associated with any chain shorter than the LCC can typically be several orders of magnitude smaller than that with the LCC. Thus, the total energy is dominated by the LCCs. Due to the low value of typical m (see Supplementary Fig. 6 in Supplementary Note

5.1), a single LCC essentially dictates the energy magnitude of the whole system. As shown in Fig. 15b, actual network control energy shares strong positive correlation and similar magnitude to the LCC energy E_L , defined as the energy of a LCC of the corresponding network, especially for networks with long LCCs. Intuitively, the probability to form long LCCs is small. Accordingly, a longer LCC tends to have smaller value of degeneracy m . As a result, the longest LCCs have almost no degeneracy ($m = 1$) so that they effectively rule the control energy of the whole network (see Supplementary Note 5.1).

Under the theory of structural controllability, a network is deemed more structurally controllable if N_D is smaller [2]. However, as the number of driver nodes is reduced, the length of the chain of nodes that each controller drives on average must increase, leading to a growth in LLC length and accordingly an exponential growth in the control energy. In the “optimal” case of structural controllability where $N_D = 1$ is achieved, the LLC length will be maximized, leading to unrealistically large control energy that prevents us from achieving actual control of the system.

The idea of exploiting the length of control chain can be used to explain the algebraic scaling behavior in the energy consumption. The construction in Fig. 16 thus provides a structural profile to estimate the control energy. In particular, a network can be viewed as consisting of a set of structural elements, the control chains, interacting with each other via the links among them, and interactions among these basic structural elements usually play an important role in determining the properties of a physical system. Hence, the total energy E required has two components: E_1 , the sum of energies associated with all control chains, and E_2 , the interaction energies among the chains. Observed from Fig. 15b, E_2 is especially important for networks with short LCCs (higher m values). The energy scaling relation shown in Fig. 14 can be derived by devising appropriate models to analyze the contributions from the two components. We have developed two such models. The first is the

LCC-skeleton model, which only takes E_1 into account and provides an analytic estimate of the control energy distribution function as well as the scaling exponent. The second is the *double-chain interaction* model, in which a system consisting only two interacting control chains captures the key features of the entire network by characterizing the essential effect of interaction energy among the structural elements. Moreover, shown by the the double-chain interaction model, the structural factors other than LCCs could also affect control energy significantly and lead to the energy deviations between BA and ER networks with the same D_C . These two models combined serve as a framework to determine the energy profile associated with controlling a complex networked system, providing a deep understanding of practical controllability (see Supplementary Note 5 for details of the two models).

4.4. Control Energy Optimization of Modeled and Real-World Networks

4.4.1. Practical Controllability of An Electrical Circuit Network.

To further test the concept and framework of practical controllability, we consider a real one-dimensional cascade parallel R-C circuit network, as schematically illustrated in Fig. 17a. (See Supplementary Note 6.1 for how the network representation of a circuit is obtained.) The network can be represented by a bidirectional 1D chain with self-loops for all the nodes, as shown in Fig. 17b. The network size can be enlarged, say by one unit, by attaching an additional branch of resistor and capacitor at the right end of the circuit. The state $u_i(t)$ of node i at time t is the voltage of capacitor i , and the input voltage $u(t)$ represents the control signal. The purpose of control is to drive the voltages of the capacitors from a set of values to another within time t_f through the input voltage $u(t)$. The control energy can then be calculated by equation (4.3). The actual energy dissipated in the circuit during the control process is given by

$$E_{\text{real}} = \int_0^{t_f} U(t) \cdot I(t) dt, \quad (4.5)$$

where $U(t) \equiv u(t)$ and $I(t)$ are the input voltage and current at time t , and E_{real} is in units of Joule. By making the circuit equivalent to a 1D chain network, we have three types of energy: the control energy of the actual circuit calculated from equation (4.3), the dissipated energy of the circuit from equation (4.5), and the control energy of the 1D equivalent network. Figure 18a shows that the control energy and the dissipated energy of the circuit do not differ substantially from the energy calculated from unidirectional 1D chain. Among the three types of energy, the energy cost associated with the control process, as calculated from equation (4.5), is maximal.

4.4.2. Strategies to Balance Control Energy and Extra Inputs.

Our finding of the LCC structure associated with the control and the exponential growth of energy with the length of LCCs suggest a method to reduce the energy significantly. Since the key topological structure that determines the control energy is LCCs, one possible approach is to reduce the length of all the LCCs embedded in a network by making structural perturbations to the network. This, however, will inevitably modify the network structure, which may not always be practically viable. Is it possible to reduce the control energy without having to change the network structure? One intuitive method is to apply additional controllers beyond those calculated from the structural-controllability theory, which we name as *redundant* controllers. A straightforward solution is to add some redundant control signals along the LCCs. (See Supplementary Note 6.2 for how the redundant control input is planted in a circuit system.) To gain insights, we consider a unidirectional 1D chain and add a redundant control input at the i th node. As shown in Fig. 18b, the magnitude of control energy is reduced dramatically. The optimal location to place the extra control should be near the middle of the chain so as to minimize the length of LCCs using a minimal number of redundant control signals. As can be seen from Fig. 18b, this simple strategy of adding one redundant control signal can reduce the required energy by nearly seven orders of magnitude.

More specifically, the redundant control signal to node i breaks a chain of length L into two shorter subchains: one of length $i - 1$ and another of length $L - i + 1$. Roughly, the control energy is the sum of energies required to control the two shorter components, which is dominated by energy associated with the longer component owing to the exponential dependence of the energy on the chain length. By choosing i around $L/2$, the length of the longer part is minimized. For the circuit network in Fig. 17, the redundant control input can be realized by inducing external current input into a capacitor. As shown in Fig. 17b, a reduction in energy of nearly 10 orders of magnitude is achieved. Applying a single redundant control input can thus be an extremely efficient strategy to reduce the required control energy for the one-dimensional chain network.

4.4.3. Control Energy Optimization of Modeled Complex Networks.

For a complex network, there often exist multiple LCCs, requiring multiple redundant control inputs. Say we wish to introduce a small number of extra control signals. Due to the m degeneracy in the end nodes LCCs, it seems that the number of redundant control inputs should exceed m if every LCC receives one such signal. However, since even a unity deduction in the LCC length can significantly lower the control energy, a simpler strategy is to place one redundant control input at each of the m end-nodes to which all possible LCCs converge. In this case, each LCC in the network is broken into a chain of length $L - 1$ and a single node, and consequently, the control energy is now determined by one-dimensional chains of length $L - 1$ instead of length L . Figure 19a shows the effects of two optimization strategies to introduce redundant control signals on the energy distribution: applying one redundant control signal (I) at the middle and (II) at the end of each and every LCC, respectively. For comparison, for each strategy, the same number of redundant control inputs are also applied randomly throughout the network. The ratio between the control energy under optimization strategy, E_x , and the original control energy E characterizes the effectiveness of

the optimization strategies. In particular, if the distribution of E_x/E is concentrated on small values of E_x/E , then the corresponding optimization strategy can be deemed to be effective. As shown in Fig. 19a, for relatively small E_x/E values, $P(E_x/E)$ s corresponding to the optimization strategies are systematically higher than under random control signal augmentation, while the opposite situation is observed for regions with relatively larger E_x/E . Thus, both optimization strategies outperform the random strategies, with strategy (I) performing slightly better than (II). The networks requiring proper optimization to be practically controlled are typically those with long control diameters. Figures 19b-d show that this is indeed the case: for networks with larger D_C , the performance of our optimization strategies deviate more significantly from random selections.

4.4.4. Practical Controllability of Real-World Networks.

Can real-world complex networks be actually controlled? In Ref. [2], the structural controllability of a large number of real-world networks were investigated, with the conclusion that optimal control of most of the networks can be achieved with only a few control signals. We investigate control energies of the same set of real-world networks (see Supplementary Table 1 in Supplementary Note 7 for network details) and find that, when optimal control is applied according to maximum matching, most of the networks require unrealistically high energies. In fact, 15 out of the 18 networks are practically uncontrollable. The main reason lies in the large LCCs of most of these networks. Another factor is that there are subgraphs that are not connected with each other and/or a large number of topological motifs such as loops, self-loops, or bidirectional edges. More strikingly, even with unlimited energy supply, the number of driver nodes as determined by the maximum matching algorithm from the structural controllability theory is generally insufficient to fully control the whole system, where there exists a number M^* of nodes that never converge to their target states. These observations lead to the speculation that, in order to fully control a real-

istic network, more driver nodes are needed than those identified by the structural controllability theory. That is, more independent control signals are needed than those determined by maximum matching to drive all nodes in the network to their target states. The M^* uncontrollable nodes are thus the required augmented set of driver nodes, each with an external control input. In total, $N_D^* = N_D + M^*$ driver nodes need to be deployed to gain full control of the system (see Fig. 20a for n_D and $n_D^* = N_D^*/N$ for the 18 real-world networks). Applying control signals to the nodes as determined by maximum matching and to the augmented driver nodes, we find that 17 out of 18 real-world networks become practically controllable (see Supplementary Table 2 in Supplementary Note 7).

We also test the enhancement strategies using the 18 real-world networks, with the result that their practical controllability can be markedly enhanced (especially for those with large control diameters), as shown Figs. 20b,c. We see that, for each of the real-world networks with unrealistically large energy requirement (see Supplementary Table 2 in Supplementary Note 6), the optimized control energy E_{mid}^* (or E_{end}^*) is several orders of magnitude smaller than the value of the original energy E^* (the control energy with M^* augmented driver nodes but without any redundant control input). This indicates the effectiveness of our optimization strategies. (In fact, strategy (I) works better than (II) in most cases.) The energy reduction due to random control signal augmentation is dramatically less effective than our strategies in most cases (almost equally effective in other cases), shown in Figs. 20b,c, which also indicates the control energy of real-world networks are indeed determined by their LCCs in general. For the networks with small control diameters, even without applying any enhancement strategy the control energies required are already much smaller than those for the other networks. For these networks energy optimization is practically unnecessary (see also Supplementary Table 2 in Supplementary Note 7).

4.5. Discussion

As stated in Ref. [2], the ultimate proof that one understands a complex network completely lies in one's ability to control it. We discover a paradox arising from controlling complex networks with respect to control energy and the number of external input signals. To resolve the paradox, we focus on the situation where the structural-controllability theory yields a minimum number of external input signals required for full control of the network, and determine whether in these situations the control energy is affordable so as to realize actual control. Our systematic computations and analysis reveal a rather unexpected phenomenon: due to the singular nature of the control Gramian matrix, in the parameter regimes where optimal structural controllability is achieved in the sense that the number of driver nodes is minimized, energy consumption can be unbearably large. To obtain a more systematic understanding, we identify the fundamental structures in a network under the action of control signals, the longest control chains (LCCs), and argue that they essentially determine the control energy. We articulate and validate that the required energy increases exponentially with the length of the LCCs. In situations where the required number of controllers is few as determined by the structural controllability theory, the length of LCCs tends to be long, leading to practically divergent control energy. Another finding is that, for minimum input signals, the required energy exhibits a robust algebraic scaling behavior, which can be explained by analyzable models constructed based on interacting LCCs. The discovery of the LCCs associated with controlling complex networks leads naturally to a simple method to resolve the paradox: increasing the number of controllers by placing extra control signals (beyond the number determined by the structural-controllability theory) along the LCCs. Indeed, test of a large number of real-world networks shows that, while they are structurally controllable [2], most of them exhibit enormous energy consumption. They can actually be controlled by placing more drivers than determined by

the structural-controllability theory at proper locations along the LCCs.

Our work indicates that the difficulty of achieving actual control of complex networks associated with even linear dynamics is beyond the current knowledge in the field of network control. Although the controllability theory offers a theoretically justified framework to guide us to apply external inputs on a minimum set of driver nodes, when we implement control to steer a system to a desired state, the energy consumption is likely to be too large to be affordable. This finding suggests that, to achieve control of a complex networked system, the existing controllability framework merely offers a necessary rather than a practically feasible condition to assure actual control. We thus demand a more comprehensive and practically useful theoretical framework for addressing the extremely important issue of controlling complex networks. However, it is difficult to develop such a framework at the present and we do not even know if a mathematically justified theory is available based on the current knowledge. Another issue is that for general networked nonlinear systems, we continue to lack the necessary condition based on the present controllability framework, as well as an understanding of required control energy. So far, we still know too little about controlling complex networked systems, and further effort is needed to address this challenging but greatly important problem shared by a wide range of fields.

Fig. 16. Schematic Illustration of Various Concepts to Characterize and Understand the Practical Controllability of a Network. (a) Control-signal paths (CSPs) of a random network obtained from maximum matching in structural controllability theory, where a control signal enters a CSP via the corresponding driver node (yellow), the starting node of the path, and goes through each matched node (blue) along the path. In this example, the network has $n_D = 10$ CSPs of different lengths. Non-path links, links that are irrelevant to matching, are displayed in green. (b) All possible LCCs in the network. Typically there are multiple LCCs of the same length. In this example, the length of the LCCs is 4, which is defined as the control diameter of the network. Two LCCs sharing no common nodes are marked by red nodes and solid red arrows. Links belonging to other LCCs are marked by red dashed arrows. CSPs are denoted using letters a to j from the left to the right. Each node is specified using its path number and its position along the path sequentially from top to bottom. For example, node $e1$ is the driver node of path e , and node $h2$ is the node right after the driver node on path h . The two LCCs with solid arrows are listed in bold. Eight LCCs in the network converge to only three end-nodes, $e6$, $f5$, and $f6$ (marked by red dashed circles), leading to LCC degeneracy $m = 3$. The control energy is determined by any randomly chosen m LCCs among all existent ones.

Fig. 17. Cascade Parallel R-C Circuit and Its Corresponding Network Presentation. (a) A cascade parallel R-C circuit with $L = 7$ resistors ($R_1, R_2, \dots$, and R_L , each of resistance 1Ω) and 7 capacitors ($C_1, C_2, \dots$, and C_L , each of capacitance $1F$). External voltage input $u(t)$ is applied onto the left side of the circuit, and the voltage of capacitor C_i is $u_i(t)$ ($1 \leq i \leq L$). (b) Network representation of the circuit in a as a bidirectional 1D chain network of seven nodes, where the external voltage input $u(t)$ is injected into node 1 (yellow driver node, the controller). The dynamical state of node i is described by the voltage on its capacitor, $u_i(t)$. Links (blue) between nodes are bidirectional and have uniform weight 1 in either direction. Each node has a self-link (red) of weight -2 , except the ending node (node 7) whose self-link has weight -1 . (c) The circuit network in b with an extra external current input $i_e(t)$ into the capacitor C_3 , where i_3 and i_4 denote the currents through resistors R_3 and R_4 , respectively. In the absence of the extra current input, $i_3(t) - i_4(t)$ is the current through the branch of C_3 . (d) The extra external current input $i_e(t)$ serves as a redundant control input injected into node 3 of the network in b. Now there are two driver nodes (yellow) in the network, nodes 1 and 3.

Fig. 18. Control Energy and Optimization for 1D Chain and Cascade Parallel R-C Circuit. (a) Energy required for controlling a unidirectional chain (red) and the corresponding circuit (blue) as well as the dissipated energy of the circuit calculated from equation (4.5) versus the chain length L . (b) Control and dissipated energies in the presence of a redundant control signal to node i ($i > 1$), which breaks the chain into two subchains of lengths i and $L - i$, respectively.

Fig. 19. Effects of Redundant Control Inputs. (a) For control diameter $D_C = 2$, distribution of the energy ratio E_x/E under optimization strategies (I) (Mid, red circles) and (II) (End, blue squares), where E_x and E are the required energies with and without redundant control, respectively. Results from two randomized optimization strategies are marked by R-Mid (black triangles) and R-End (gray diamonds), corresponding to strategies (I) and (II), respectively. The values of E_x/E are collected from practically controllable networks from an ensemble of 10000 ER random networks ($\langle k \rangle = 6$, $P_b = 0.1$). For each network, if strategy (I) [or (II)] requires r redundant controls, r additional random control inputs are applied to the system 10 times to average out the random fluctuations. Panels (b-d) show the E_x/E distributions for networks with control diameter $D_C = 3$, 4, and 5, respectively.

Fig. 20. Augmented Control Inputs and Energy Optimization. (a) Densities of the original driver nodes n_D (dark purple) and of the augmented controls n_D^* (light purple). (b) Normalized energy reduction $\Delta E_{\text{mid}}^*/E^* = (E^* - E_{\text{mid}}^*)/E^*$ (light blue) when an additional control signal is added to the middle of each LCC [strategy (I)] and normalized energy reduction $\Delta E_{\text{R-mid}}^*/E^* = (E^* - E_{\text{R-mid}}^*)/E^*$ (dark blue) when the same number of control signals are randomly added into each network. (c) Normalized energy reduction $\Delta E_{\text{end}}^*/E^* = (E^* - E_{\text{end}}^*)/E^*$ (light green) when an additional control signal is added to the end node of each LCC [strategy (II)] and normalized energy reduction $\Delta E_{\text{R-end}}^*/E^* = (E^* - E_{\text{R-end}}^*)/E^*$ (dark green) when the same number of control signals are randomly added into each network. All colored bars (light/dark purple, light/dark blue, and light/dark green) start from 0, and the darkly colored bars overlap with the corresponding portions of the lightly colored bars. For the bars with only dark colors, the lightly colored bars are in the same height but covered by the dark ones in front of them. In majority cases, our optimization

strategies work better than random control signal augmentation, and in 7 out of 17 networks, the energy reduction under our strategies is one or several order of magnitude greater than under random

5 . CONTROLLING EXTREME EVENTS ON COMPLEX NETWORKS

5.1. Introduction

Extreme events occur on a large variety of complex networked systems, examples of which include sudden bursts of packet flows in the Internet, jamming in computer or transportation networks, abrupt rise of web requests, and power blackouts in electrical power grid, etc. The intrinsic dynamics responsible for extreme events are network traffic or flow dynamics [163–175] and the corresponding flux-fluctuation behaviors [48, 176–183]. Especially, abnormally large fluctuations in the flow exceeding the nodal or link capacities can lead to extreme events, which either can be triggered by external disturbances or are a type of intrinsically emergent behavior. Extreme events can have catastrophic consequences, demanding the articulation and development of effective control strategies [48, 184, 185]. This problem, despite its uttermost importance to many disciplines, has not been addressed. The purposes of this report are to present an efficient and physically implementable method and to derive an analytic theory to understand its working.

This work aims to develop a physically practical control scheme against extreme events emerged from the intrinsic dynamic factors by using a simple but extensively representative model that captures the essential mechanism of many complex networked systems with information, material, or energy transportation. Within this model, as presented in the **Methods** Section, the random variables essential to the emergence extreme events are sums of random variables distributed in a binomial fashion (thin-tailed) which have well-defined first and second order moments. Thus, these sums follow the Lévy-alpha-stable distribution $g_\alpha(x)$ with $\alpha = 2$ [186, 187], and we do not expect the model to characterize all properties observed in reality, such as the "fat-tailed" distributions corresponding to $\alpha < 2$.

Our principal idea of controlling extreme events is to make the whole network system time-

dependent in that every node is mobile and consequently the associated set of links is time-varying. The mobility of the nodes can be characterized by a velocity-like parameter. The system can thus be modeled as a mobile network in which agents move in random directions but with constant velocity. Between any pair of agents, a link is established to enable exchange of packets only when their physical distance is within a pre-defined communication radius. We find that, the number of extreme events depends on the velocity and, strikingly, it exhibits a bell-shape, resonant-type of functional relation, as exemplified in Fig. 21. That is, there exists an optimal velocity for which the number of extreme events can be significantly reduced as compared with that in a static network. This finding suggests a practical control scheme: the occurrence of extreme events can be suppressed by making the network mobile with proper agent velocity. To understand this striking phenomenon, we develop a detailed analytic theory. To extend the control, we find a generalized mobility capable of dramatically lowering the control cost. We also develop a low-cost “intermittent strategy” that harnesses extreme events not only in mobile networks, but also in the classical ER-random networks [159]. Since extreme events encompass catastrophic situations such as large-scale breakdown induced by random failures of nodes or attacks, the general principle that dynamic networks can be immune to extreme events can have significant applications in the field of security and robustness of complex systems.

We emphasize that our work is focused on controlling extreme events caused by the intrinsic randomness of the system, rather than external attacks. More sophisticated mechanisms such as failures and load redistribution, e.g., as triggered by external attacks, would be an interesting topic of study. Instead of analyzing the extreme value distributions above the threshold as determined by the extreme value theory [188, 189] under the peak-over-threshold setting, our work concentrates on calculating the total number of events that exceed a certain threshold for the purpose of control.

Fig. 21. The Number of Extreme Events Versus Velocity and Mobility. For a network of $N = 1200$ nodes in a square domain of size $L = 10$ (arbitrary unit), **a**, The number of extreme events n_{ex} versus the agent velocity v for $W = 50N$, $80N$, and $100N$; **b**, n_{ex} of extreme events versus velocity v when nodes execute random walk (R) or deterministic motion (D) within square, circular, and stadium-shaped domains. The areas of the circle and the stadium are chosen to be equal to the area of the square $L \times L$. **c**, Robustness of control with respect to heterogeneity in the nodal communication range: n_{ex} versus v for $\beta = -1, 0$, and 1 , where β is a parameter characterizing the distribution of nodal communication range. **d** and **e**, robustness of control with respect to packet-transportation protocol: n_{ex} versus velocity v when packet generation and annihilation are taken into account for two situations where **d** the total number of packets is fixed at W and **e** the number of newly generated packets at each time step is fixed to be W/N . **f**, *Generalization of mobility*: n_{ex} versus mobility, defined as the probability that a node moves with velocity v , for a number of v values. All simulation results are obtained using 100 realizations in $T = 1000$ time steps. The error bars in n_{ex} from different realizations have relatively small and similar magnitude in **(a-e)**, and it monotonously increases with velocity v , as shown in panel **a** [the error bars in **(b-e)** are omitted for clear visualization]. Here, the definition of an extreme event on a node is that its number of packets is at least four standard-deviations above the average.

5.2. Results

We consider a square domain of size $L \times L$ in which N agents move at speed v but in random directions. The communication radius of each agent is $a \ll L$, and the total number of packets on the network at any given time is W . At time t , the number of packets carried by a node is $w(t)$, and each packet is delivered to a randomly selected neighboring node at time $t + 1$. In contrast to previous works on synchronization in mobile-agent networks [190–192] where periodic boundary conditions were often used, we use the “hard-wall” boundary condition for the consideration that a cross-boundary link connecting two nodes close to the same boundary but physically distant from each other is not physical in an actual traffic-flow network, which is discussed in a later section. We will see that boundary effect plays a crucial role in the dynamics of extreme events, a fact that will be fully exploited in our development of the analytic theory.

5.2.1. The Existence of Optimal Mobility and Its Robustness

Our extensive computations indicate that the existence of an optimal moving velocity to suppress extreme events is general: it holds under a variety of boundary shapes, motion schemes, and even heterogeneously distributed communication radii, as shown in Figs. 21a,b, and c. This indicates that the characteristic, non-monotonous relation between n_{ex} and velocity v that underlies the control of extreme events is in fact robust with respect to variants in the system setting such as the geometric shape of the domain, the manner by which the individual nodes move, heterogeneity in the nodal communication range, and packet transportation protocol.

As can be seen from Fig. 21a, different W values do not affect the existence of the optimal moving velocity. Figure 21b shows, for a number of variants in the geometric shape of the domain (square, circle, and stadium), the relation between n_{ex} and v . We observe that the non-monotonous behavior holds for all geometric shapes considered. As shown, the particular manner by which

Fig. 22. Geometrical Setting and the Theory Result. **a**, Illustration of various geometrical quantities used in the analysis: communication radius a , distance b from the middle of the square to the target node, distance b_n from the middle of the square to a neighbor of the target node, the width Δb_n of the bar region, and the cutoff region of the communication circle. **b**, Results of Eq. 5.2 for $db = 0.01$ and $\Delta b_n = 0.002$ (the main features do not depend on the specific values of db and Δb_n).

nodes in the network move in the space has little effect on the non-monotonous behavior. In particular, both random walk and deterministic motion are considered, where for the latter, random moving directions are chosen initially but at time t , a node moves along the direction of the velocity at time $t - 1$. Figure 21b indicates that the existence of an optimal velocity to minimize the number of extreme events is highly robust regardless of the domain geometry, random or deterministic movements.

In realistic systems, mobile agents' ranges are not homogeneous. To model heterogeneity in the nodal communication range, we randomly assign each node a weight μ according to the distribution $P(\mu) \sim \mu^{-3}$. The area of the nodal communication circle is given by $A_i = NA_0 \frac{\mu_i^\beta}{\sum_{l=1}^N \mu_l^\beta}$, where $A_0 = \pi a^2$ is the average communication area over all nodes, μ_i and μ_l are the assigned

weight values corresponding to nodes i and l , and β is an adjustable parameter characterizing the heterogeneity of A_i . The distribution of A_i is thus $P(A_i) \sim A_i^{-3/\beta}$. For $\beta > 0$, majority of nodes have relatively small communication circles except for a few, and the opposite situation occurs for $\beta < 0$. For $\beta = 0$, the communication circles are all identical: namely $A_i = A_0$ and $a_i = a$ for $i \in [1, 2, \dots, N]$. Since the degree of a node is proportional to its communication area, it is equivalent to a power-law degree distribution of the mobile network with the fixed average degree $\langle k \rangle = N/L^2 A_0$. The physical meanings of positive and negative values of β are the following. In a realistic wireless sensor network, for example, there are usually anchor nodes that have much larger communication ranges and play the role of central controllers of the whole network. This situation can be characterized by some positive value of β . There can also be a few nodes that malfunction or run low on power supply. These nodes would have nearly zero communication ranges while most nodes in the network have finite communication ranges. This corresponds to the case of a negative β value. Figure 21c shows that the non-monotonous behavior in the $n_{\text{ex}}-v$ relation holds regardless of whether the value of β is positive or negative, indicating the robustness of mobility-based strategy to harness extreme events in the presence of nodal heterogeneity.

We also consider realistic variants of the packet-transportation model by randomly assigning each packet a destination node where the packet annihilates upon its arrival at the destination, and new packets are generated to balance the annihilation effect. Two packet generation schemes are studied: (1) the total amount of packets W is kept constant by generating the annihilated number of packets at randomly selected nodes; (2) W/N newly generated packets are randomly distributed to nodes to keep W statistically constant but with fluctuations (the average number of annihilated packets is W/N for high velocity). As shown in Figs. 21d and e, nearly identical behavior in the $n_{\text{ex}}-v$ relation is observed, regardless of the specific packet-transportation protocol, providing

further support for the robustness of our control method.

While so far we have focused on dynamic networks where agents move with certain velocity, the idea of making a network mobile to control extreme events can be generalized to different forms of “mobility.” Here we discuss one variant, in which the mobility is defined as the probability that any node moves with velocity v . For relatively high values of v , there exists an optimal mobility value at which the number of extreme events is minimized, as shown in Fig. 21f. Notice that the optimal mobility is near zero, indicating that only a small fraction of the nodes actually move and vast majority of the nodes remain stationary. The implication is that extreme events can be effectively mitigated by moving a few nodes, which is advantageous from the consideration of control cost.

5.2.2. Explanation and Boundary Effect

In the following, we focus on the development of analytical understanding of the phenomenon. To begin, we note that, since each node receives packets at time t from its neighbors that equally distribute all the packets obtained at $t - 1$ to all their neighbors, a node’s flow $w(t)$ at time t is contributed to by its neighbors’ flow at time $t - 1$, denoted by $w_n(t - 1)$. In a static network, the ensemble averaged flow through a node at last time step is proportional to its degree $\langle w(t - 1) \rangle = Wk/(2E)$. However, in a time-varying network, this relation no longer holds. In particular, the flow would be independent of the degree, leading to $\langle w(t - 1) \rangle = W/N$. We thus hypothesize the following general relation between $\langle w(t - 1) \rangle$ and node degree for different regimes of the velocity v :

$$\langle w(t - 1) \rangle = \xi(v) \cdot k(t) + \eta(v), \quad (5.1)$$

where $\xi(v)$ and $\eta(v)$ are parameters that depend on the velocity only, and the first and second terms correspond to the velocity-dependent and velocity-independent flow components, respectively. Generally, the quantity $\xi(v)$ decreases with v , since higher mobility tends to yield a weaker flow-degree

Fig. 23. Boundary Effect. **a1, b1, c1**, Behavior of $\langle w(t) \rangle_b$ versus b from direct simulation (blue squares) and predicted by Eq. (5.8) (solid black lines) for $v = 0, 0.5$ and 3.0 , respectively. Note that $v \approx 0.5$ is the optimal value of the velocity that minimizes n_{ex} in Fig. 21, and $v = 3.0$ represents a relatively high velocity value for the system of size $L = 10$ and communication radius $a = 1$. For **a1**, we have $\eta(v = 0) = 0$. Since Eq. (5.8) indicates a smooth dependence of $\langle w(t) \rangle_b$ on b and since a detailed interdependence between $\eta(v)$ and v is unknown, we find $\eta(v) = 15$ best fits the case in **b1**. For **c1**, we have $\eta(v = 3.0) \rightarrow \eta(v \rightarrow \infty) = 50$. The red dashed line indicates the position of the peak in the curve of $\langle w(t) \rangle_b$ versus b . **a2, b2, c2**, Positions of all extreme events occurred in square domain for $v = 0, 0.5$, and 3.0 , respectively. **a3, b3, c3**, Positions of all extreme events occurred in circular domain for $v = 0, 0.5$, and 3.0 , respectively. **a4, b4, c4**, Positions of all extreme events occurred in stadium domain for $v = 0, 0.5$, and 3.0 , respectively. All results are obtained from 20 independent statistical realizations.

correlation. However, $\eta(v)$ tends to increase with v due to the increasing degree of randomness in the dynamics. Thus, for $v \rightarrow 0$, the steady network exhibits a high flow-degree correlation and little randomness, so we have $\xi(v) = W/(2E)$ and $\eta(v) = 0$. For $v \rightarrow \infty$, the correlation is lost, leading to $\xi(v) = 0$ and $\eta(v) = W/N$. Eq. (5.1) thus represents a universal empirical law, which we have verified computationally through various models (see the Methods section). Conservation of the total number of data packets imposes the following constraint on the parameters $\xi(v)$ and $\eta(v)$: $W = N \sum_{k(t)=0}^{N-1} \langle w(t-1) \rangle P(k) \Rightarrow \xi(v) = [W - N \cdot \eta(v)]/(2E)$. Setting $\eta(v) = 0$ (or $\eta(v) = W/N$) yields $\xi(v) = W/(2E)$ (or $\xi(v) = 0$). We see that Eq. (5.1) contains the results for $v = 0$ and $v \rightarrow \infty$ as two limiting cases.

With the help of Eq. (5.1), the probability for the node at location b (illustrated in Fig. 22a) to carry w packet, $\Psi(w(t))_b$, is obtained. This distribution $\Psi_b(w(t))$ serves as a centro-symmetric probability field for each dimension of the two-dimensional space. For a static node located at b_0 , the PDF of its flow is $\Psi_{b_0}(w(t))$. For any moving node ($v > 0$) initially at b_0 , the PDF of its flow, $\Psi_{\mathfrak{B}}(w(t))$, is contributed by the $\Psi_b(w)$ fields at all the locations it has visited, and thus can be expressed as an integral of $\Psi_b(w)$. In turn, the total number of extreme events occurred within T time steps is obtained as an integral of $\Psi_{\mathfrak{B}}(w(t))$ over all possible starting location,

$$n_{\text{ex}} = NT \int_0^{L/2} \left[\sum_{w(t)=q_{\mathfrak{B}}+1}^W \Psi_{\mathfrak{B}}(w(t)) \right] \cdot \frac{8b}{L^2} db. \quad (5.2)$$

where $q_{\mathfrak{B}}$ is the extreme event threshold. Since $\eta(v)$ monotonously increases with v , Figure 22b shows that Eq. (5.2) successfully explains the numerically observed non-monotonous behavior as exemplified in Fig. 21a, providing the theoretical foundation for mobility-based control of extreme events.

Our theory provides a detailed understanding of the effect of velocity on the occurrence of extreme events. To further demonstrate this, we exploit the effect of domain boundary on the

emergence of extreme events by examining the behavior of $\langle w(t) \rangle_b$ versus b , as exemplified in Figs. 23a1, b1, c1, where a good agreement between our theory and simulation for various velocity values can be seen. Note that $b_f = \frac{L}{2} - a$ specifies the position of one communication radius away from the boundary. For $v = 0$, $\langle w(t) \rangle_b$ is constant for $b \in [0, b_f]$ and decreases with b in the interval $[b_f, \frac{L}{2}]$. As v is increased, a peak in $\langle w(t) \rangle_b$ arises for $b = b_f$, but on the left and right sides of the peak, $\langle w(t) \rangle_b$ is approximately constant and a decreasing function of b , respectively. For convenience, we call $[b_f, \frac{L}{2}]$ the *boundary region*. As a node moves closer to the boundary, more portion of its communication circle is cut off, leading to lower values of $\langle k(t) \rangle$ and $\langle w(t) \rangle_b$. For $v \rightarrow 0$, $\Psi_{\mathfrak{B}}(w(t))$ is highly localized so that $\Psi_{\mathfrak{B}}(w(t)) \rightarrow \Psi_b(w(t))$. For the static network, extreme events are approximately equally probable in the domain, except in the boundary region, as shown in Fig. 23a2. As velocity is increased from zero, $\Psi_{\mathfrak{B}}(w(t))$ depends more strongly on $\Psi_b(w(t))$'s and the peak in $\langle w(t) \rangle_b$ about b_f generates a higher value of $q_{\mathfrak{B}}$ in the neighborhood region of b_f , reducing the probability of extreme events. In the part of the boundary region where low values of $\langle w(t) \rangle_b$ meets with high value of $q_{\mathfrak{B}}$, there are no extreme events, as shown in Fig. 23b2. For high velocity, each node appears in the domain with approximately equal probability and $\Psi_{\mathfrak{B}}(w(t))$ is completely delocalized, so we have $\Psi_{\mathfrak{B}}(w(t)) = (8/L^2) \int_0^{L/2} b \cdot \Psi_b(w(t)) db$. In this case, there is little difference in the dynamical behavior of nodes. When a node is near b_f , the sudden increase in the flow can cause an extreme event, which is the mechanism responsible for high concentration of extreme events about $b = b_f$, as shown in Fig. 23 c2.

Figs. 23a3, b3, c3, and a4, b4, c4 show the patterns of the locations of the extreme events occurring on nodes for $v = 0, 0.5$, and 3 within the circular and stadium-shaped domains, respectively, where the motions of nodes are completely random. For deterministic motions of the nodes, the patterns are similar. Regardless of the geometrical shape of the domain, a frame of extreme events

emerges about one communication radius away from the domain boundary for high velocity, providing visual support for the basic idea underlying our theory: the boundary plays an important role in the emergence of extreme events.

To intuitively understand the non-monotonous phenomenon, we name the nodes within the boundary region (region between the boundary and the flame) as boundary nodes (BNs), the nodes sitting on the frame as frame nodes (FNs), and the nodes within the frame as inside nodes (INs). Each node moves continuously in the space and acts as one of the three types of nodes from time to time. Due to the cutoff in the communication circles in the boundary region, the BNs have lower average degree than the FNs and INs, namely $\langle k_{\text{BN}} \rangle < \langle k_{\text{IN}} \rangle = \langle k_{\text{FN}} \rangle$. At high velocity, most BNs are in fact originally INs from the inside carrying $\langle w_{\text{IN}} \rangle$ packages, where $\langle w_{\text{IN}} \rangle$ is the average flow of the INs. Apparently, we have $\langle w_{\text{IN}} \rangle / \langle k_{\text{IN}} \rangle < \langle w_{\text{IN}} \rangle / \langle k_{\text{BN}} \rangle$. Thus, for BNs, each of their links delivers more packages out than the INs. As recipients, FNs and INs have the same average degree ($\langle k_{\text{IN}} \rangle = \langle k_{\text{FN}} \rangle$) while FNs have more BN neighbors than INs have. We then have $\langle w_{\text{FN}} \rangle > \langle w_{\text{IN}} \rangle$, i.e., FNs receive more packages. In addition, the FNs have more neighbors than the BNs, and thus we have $\langle w_{\text{FN}} \rangle > \langle w_{\text{BN}} \rangle$. This mechanism produces the frame with higher density of extreme events due to a sudden increase in the flow when BNs and INs move into the region of FNs. However, the frame so generated also increases the threshold q_{B} and accordingly decreases the probability of occurrence of extreme events. Consequently, the coexistence of the two competing mechanisms generates the non-monotonous behavior, leading to an optimal velocity. Our theory, albeit non-rigorous, quantifies the relative strength of the two mechanisms and determines whether extreme events are facilitated or suppressed as velocity is increased.

Fig. 24. Effect of Long Range Links. **a**, effect of periodic boundary condition: n_{ex} versus v when 0% (green circles, no PBC), 30% (aqua triangles), 50% (pink triangles), and 100% (purple triangles, complete PBC) randomly selected cross-boundary links are applied. **b**, *Effect of communication wormhole on control*: n_{ex} versus v when a wormhole link is applied to any pair of nodes closer than $d_{\text{HW}} = 13.0, 12.5, 12.3$, and 12.0 for $p_{\text{HW}} = 0.4\%, 2\%, 3\%$, and 5% , respectively.

5.2.3. The Effect of Wormholes

In fact, consider an artificial system in which a fraction of links can be made cross-boundary. We find that, as the number of such “periodic-boundary-condition type” of links is increased, the ability for the mobile network to suppress extreme events is degraded dramatically, as shown in Fig. 24a. This finding inspires significant applications in the emerging field of cybersecurity. Here we present one example by arguing that our control strategy can be used to detect wormhole attacks on mobile networks. In particular, a communication wormhole is a long-range link connecting two nodes beyond each’s communication range, a recently developed concept in the field of network security [193–195]. Suppose any pair of nodes with distance over d_{wh} are connected by a wormhole. A smaller value of d_{wh} will then introduce more wormholes into the system. Let p_{wh} be the ratio between the number of wormholes and the total number of edges E in the network. As shown in Fig. 24b, for moderate or relatively high velocity, a slight decrease in d_{wh} can trigger an explosive increase in n_{ex} but there is only incremental change in p_{wh} . Thus, abnormally large number of extreme events in a mobile network are strong indication that the network is under wormhole attack. Likewise, instilling a small number of wormholes into an adversary mobile network can effectively destroy its functions through the generation of a large number extreme events.

5.2.4. Suppression of Extreme Events in Intermittently Mobile Networks

A network can be made mobile but only intermittently by introducing a “frozen” time interval ΔT during which the network structure is fixed. Specifically, each time after certain nodes in the network have moved, the network structure remains invariant for ΔT time steps before the next movement. For $\Delta T = 0$, the network structure changes at every time step. Regardless of whether nodes are moving, the packet flow on the network is not interrupted. As shown in Figs. 25a-d, the number n_{ex} of extreme events exhibits a non-monotonous behavior with velocity v (for fixed values

Fig. 25. Number of Extreme Events in Intermittently Mobile Networks. **a** and **b**, For a mobile network with hard-wall boundary conditions, n_{ex} versus v and for a number of fixed values of the frozen time ΔT and versus ΔT for three values of v . **c** and **d**, Corresponding results for periodic boundary conditions. **e** and **f**, For a classic ER-network under intermittent random rewiring (a kind of generalized mobility), n_{ex} versus the rewiring probability P_R (for a number of fixed values of ΔT) and versus ΔT (for $P_R = 0.5, 0.8$, and 1.0), respectively.

of ΔT) or with ΔT (for fixed value of v). In particular, Figs. 25a,b show, under hard-wall boundary conditions, n_{ex} versus v for a number of fixed values of ΔT and versus ΔT for three values of the velocity, respectively. We observe that, increasing ΔT slightly above 0 leads to a sharp decrease in n_{ex} , indicating that making the network intermittently mobile can be a cost-effective strategy to suppress extreme events. This holds regardless of the boundary condition. In fact, an intermittently mobile network system under periodic boundary conditions possesses a similar ability to suppress extreme events, as shown in the corresponding plots in Figs. 25c,d. All these indicate mobility as a general and effective strategy to suppress extreme events.

While the concept of “mobility” can readily be generalized to static networks in which the nodes are not allowed to move physically, the existence of a physical boundary plays a critical role in controlling extreme events. For example, a network can be regarded as “mobile” insofar as its connection topology is time-varying, which can be realized, for example, by random rewiring of a subset of links from time to time in a computer network. In this case, an actual physical domain for the network cannot be properly defined. To demonstrate the importance of a physical boundary, we consider a classic ER network [159] with random rewiring. Let P_R be the rewiring probability, the probability that an arbitrary link in the network is removed while a new link is established between a randomly selected pair of nodes that were not previously linked. Since the ER random network has no spatial structure, it is the change of nodal degree that leads to the variation in n_{ex} as P_R is increased. As shown in Fig. 25e, n_{ex} cannot be minimized by P_R , indicating inability for random rewiring to suppress extreme events. Simulations also show that the number of extreme events is not affected by the detailed rewiring mechanism, insofar as the nodal degree is unchanged. However, intermittent rewiring can reduce n_{ex} to certain extent, as shown in Fig. 25f. For high value of P_R in the ER random network, the number of extreme events, n_{ex} , can be solved analytically via the

Fig. 26. Degree Bias and Scaling. For a mobile network in a square domain with hard-wall boundary conditions, **a**, number n_{ex} of extreme events occurring on small (red circles) and large (blue squares) degree nodes versus v , and **b**, n_{ex} versus the total number of packets W for various velocity values. approach of self-consistent equation (see **Methods** Section).

5.2.5. Degree Bias and Scaling

A previous result [48] indicated that, in static networks, extreme events tend to occur more frequently on low-degree nodes. However, in any real-world network, certain degree of mobility is expected. Our computation indicates that, insofar as the network is mobile, extreme events tend to occur on highly connected nodes whose degrees are larger than the average degree of the network, as shown in Fig. 26a for a mobile network with hard-wall boundary conditions. Similar behaviors have been observed for mobile networks with periodic boundary conditions and for ER-random networks as well. We also find a scaling phenomenon between n_{ex} and the total number of packets W , as exemplified in Fig. 26b.

5.3. Discussion

To summarize, we find that extreme events in a complex network can be effectively controlled by making the network mobile, where the concept of mobility is general and can be implemented in various ways. A straightforward manifestation is that every node moves at a certain velocity but

in random directions in a finite physical domain, in which case an analytic theory is developed to understand the existence of an optimal velocity to minimize the number of extreme events. Variants of the mobile schemes can also be considered, proving the generality of our control strategy. Our theory has significant applications in the field of cybersecurity, where we articulate and demonstrate, for example, that wormhole attack on a mobile network can be detected through the monitoring of extreme events. Control of extreme events is an extremely important and challenging problem in science and engineering, and our work may stimulate further efforts.

5.4. Methods

5.4.1. Relation Between $\xi(V)$ and $\eta(V)$

Figure 27 show the relationship between a node's flow $w(t - 1)$ at time $t - 1$ and its degree $k(t)$ at time t for different velocity values, the basic relation employed in our theoretical analysis in the main text, and the function $\eta(v)$ is shown in Fig. 27d. We observe a robust linear relation for mobile networks with periodic boundary conditions and for ER-random networks [159]. For mobile networks with hard-wall boundary conditions, the relation is still linear, especially for low or high velocity values. For moderate velocity values, there is deviation from the linear behavior, due to the *boundary effect* as shown in Fig. 3. (This boundary effect does not exist in mobile networks with periodic boundary conditions and in ER-random networks.) In general, we observe an approximately linear relation between $w(t - 1)$ and $k(t)$, which can be taken as an empirical law for our analysis of the extreme-event dynamics in mobile networks.

We present a detailed derivation of the relation between $\xi(v)$ and $\eta(v)$. The starting point is the

Fig. 27. Linear Relation Between $\langle W(T-1) \rangle$ and $k(T)$. **a-c)**, Linear relation between $\langle w(t-1) \rangle$ and $k(t)$ for mobile networks with hard-wall and periodic boundary conditions, and for ER-random networks, respectively. **d)**, Dependence of $\eta(v)$ on v for mobile networks with hard-wall and periodic boundary conditions, and $\eta(P_R)$ versus P_R for ER-random networks (inlets). All results are averaged over 20 statistical realizations.

following packet conservation law:

$$\begin{aligned}
W &= N \sum_{k(t)=0}^{N-1} \langle w(t-1) \rangle P(k) \\
\Rightarrow \frac{W}{N} &= \sum_{k(t)=0}^{N-1} [\xi(v) \cdot k(t) + \eta(v)] P(k(t)) \\
\Rightarrow \frac{W}{N} &= \xi(v) \cdot \sum_{k(t)=0}^{N-1} k(t) P(k(t)) + \eta(v) \cdot \sum_{k(t)=0}^{N-1} P(k(t)) \\
\Rightarrow \frac{W}{N} &= \xi(v) \cdot \langle k(t) \rangle + \eta(v) \\
\Rightarrow \xi(v) &= \frac{W/N - \eta(v)}{\langle k(t) \rangle}.
\end{aligned} \tag{5.3}$$

The average degree of the whole network is $\langle k(t) \rangle = 2E/N$. We thus have

$$\xi(v) = \frac{W/N - \eta(v)}{2E/N} = \frac{W - N \cdot \eta(v)}{2E} \tag{5.4}$$

5.4.2. Detailed Analytical Explanation to the Non-Monotonous Behavior

As illustrated in Fig. 22a, the probability that z packets are delivered to the focal node located at b from a neighbor of degree $k_n(t)$ located at b_n and carrying $w_n(t-1)$ packets is given by

$$\begin{aligned}
P(z, k_n(t), w_n(t-1)) &= \binom{w_n(t-1)}{z} \left[\frac{1}{k_n(t)} \right]^z \\
&\cdot \left[1 - \frac{1}{k_n(t)} \right]^{w_n(t-1)-z} \cdot P_k(k_n(t)) \cdot P(w_n(t-1)),
\end{aligned} \tag{5.5}$$

where $P_k(k_n(t))$ is the degree distribution of the neighboring nodes:

$$P_k(k_n(t)) = \binom{N-2}{k_n(t)} \left[\frac{S(b_n)}{L^2} \right]^{k_n(t)} \left[1 - \frac{S(b_n)}{L^2} \right]^{N-2-k_n(t)},$$

and $S(b_n)$ is the circular communication area of the neighbor located at b_n . For $b_n \leq L/2 - a$, the area is a complete circle with $S(b_n) = \pi a^2$. However, for $L/2 - a < b_n \leq L/2$, a cutoff by the boundary leads to $S(b_n) = \pi a^2 (1 - \alpha/\pi) + a \sin \alpha \cdot (L/2 - b_n)$, where $\alpha = \arccos((L/2 - b_n)/a)$, as

shown in Fig. 22a. The quantity $P(w_n(t-1))$ is the probability density function (PDF) of $w_n(t-1)$, which depends on the velocity v and further analysis is needed to determine its mathematical form.

Using Eq. (5.1), we can write the probability for a packet to visit a node as $\langle w \rangle / W = (1/W)[\xi(0) \cdot k + \eta(0)]$ for $v = 0$. For $v \neq 0$, the probability for a node at b_n to be visited is $p_n = \langle w_n(t-1) \rangle / W = [\xi(v) \cdot k_n(t) + \eta(v)] / W$. We thus have $P(w_n(t-1)) = \sum_{k_n(t)=1}^{N-2} \binom{W}{w_n(t-1)} p_n^{w_n(t-1)} (1-p_n)^{W-w_n(t-1)} \cdot P(k_n(t))$. Substituting this expression into Eq. (5.5), we obtain the PDF of z as $P(z) = \sum_{w_n(t-1)=z}^W [\sum_{k_n(t)=1}^{N-2} P(z, k_n(t), w_n(t-1))]$. Consider any one of the boundaries of the square domain, as shown in Fig. 22a. For the focal node located at b , $Z = \sum_{i=1}^m z_i$ packets will be delivered to it from its m neighbors located in the bar-shaped region of width $\Delta b_n \rightarrow 0$ at b_n . The PDF of Z associated with b_n is the convolution of the m PDFs, taking into account all possible values of m : $P_{b_n}(Z) = \sum_{m=0}^{N-1} [P(z_1) * P(z_2) * \dots * P(z_m)] \cdot P(m)$, where $P(m)$ is the probability for m nodes to exist in the bar region:

$$P(m) = \binom{N-1}{m} \left(\frac{S_{\text{bar}}}{L^2} \right)^m \left(1 - \frac{S_{\text{bar}}}{L^2} \right)^{(N-1-m)} \quad (5.6)$$

with area given by $S_{\text{bar}} = 2\Delta b_n \sqrt{a^2 - (b_n - b)^2}$ for small Δb_n . The flow $w(t)$ of the node at time t is the sum of the incoming packets from nodes in all the bar-shaped regions within its communication circle: $w(t) = \sum_{j=1}^M Z_j$, where the contribution from each region can be treated independently and $M = [\min(\frac{L}{2}, b+a) - (b-a)] / \Delta b_n$. The PDF of $w(t)$ can then be expressed as the convolution of all $P_{b_{n_j}}(Z_j)$'s for $b_{n_j} \in [b-a, \min(L/2, b+a)]$, as $\Psi_b(w(t)) = P_{b_{n_1}}(Z_1) * P_{b_{n_2}}(Z_2) * \dots * P_{b_{n_M}}(Z_M)$.

For a static node located at b_0 , the PDF of its flow is $\Psi_{b_0}(w(t))$. For any moving node ($v > 0$) initially at b_0 , the PDF of its flow is contributed by the $\Psi_b(w)$ fields at all the locations it has visited,

i.e., the circular region centered at b_0 with estimated radius $R(v)$. The PDF of this node's flow thus contains the contributions from all $\Psi_b(w(t))$'s for $b \in [b_0 - R(v), \min(b_0 + R(v), L/2)]$. The random walk radius $R(v)$ increases with v , and $R(v) \rightarrow 0$ for $v \rightarrow 0$ and reaches $L/2$ (the upper bound of b) for $v \rightarrow \infty$. Since $R(v)$ and $\eta(v)$ both increase monotonically with v , we have, approximately, $R(v) \simeq \eta(v)/(2W/NL)$. Consequently, the flow PDF of a node initially at location b moving with velocity v can be obtained as a function of b :

$$\Psi_{\mathfrak{B}}(w(t)) = \frac{\int_{\max[0, b-R(v)]}^{\min[b+R(v), L/2]} \Psi_b(w(t)) \cdot \frac{8b}{L^2} db}{\int_{\max[0, b-R(v)]}^{\min[b+R(v), L/2]} \frac{8b}{L^2} db}, \quad (5.7)$$

where $8b db/L^2$ is the probability that a node is in the region about the location b via random walk, and the denominator on the right side is a normalization factor. The quantity $\Psi_{\mathfrak{B}}(w(t))$ determines the threshold $q_{\mathfrak{B}} = \langle w(t) \rangle_{\mathfrak{B}} + 4\delta_{w\mathfrak{B}}$ over which extreme events take place, where $\langle w(t) \rangle_{\mathfrak{B}} = \sum_{w(t)=0}^W w(t) \cdot \Psi_{\mathfrak{B}}(w(t))$ and $\delta_{w\mathfrak{B}} = \sqrt{\sum_{w(t)=0}^W [w(t) - \langle w(t) \rangle_{\mathfrak{B}}]^2}$. Finally, we obtain the number of extreme events occurred within T time steps as

$$n_{\text{ex}} = NT \int_0^{L/2} \left[\sum_{w(t)=q_{\mathfrak{B}}+1}^W \Psi_{\mathfrak{B}}(w(t)) \right] \cdot \frac{8b}{L^2} db,$$

which is a function of $\eta(v)$.

In the derivation of Eq. (5.2), the degrees of the neighboring nodes are treated independently in Eq. (5.5) and the expression of $P(w_n(t-1))$, which is an approximation because nodes that are spatially close tend to be degree-correlated. The spatial degree correlation represents the main discrepancy between our moving random-network model and a mobile network in the real physical space (see the following for a detailed discussion about the effect of spatial correlation). Notwithstanding, the degree correlation has no dependence on the velocity and, hence, there is little effect on extreme-event generation in the system for different values of the velocity.

Fig. 28. PDF of Flow $P(W)$ in Simulation and in Different Theory Assumptions. **a**, For the classic ER random networks with rewiring probability $P_R = 1$, PDF of flow $P(w)$ obtained from direct simulation (red squares) and self-consistent equation (solid blue line). **b**, PDF of flow $P(w)$ from direct simulation (red squares), from theory without taking into account spatial degree correlation (black circles), and from theory with degree-degree correlation (solid blue line).

The average flow $\langle w(t) \rangle_b$ at location b can be obtained via $\Psi_b(w(t))$ as well as the mean-field theory:

$$\begin{aligned} \langle w(t) \rangle_b = & \int_{\frac{L}{2}-a}^{\min(\frac{L}{2}, b+a)} \frac{\langle w_n(t-1) \rangle}{\langle k_n(t) \rangle} \cdot \frac{N \sqrt{a^2 - (b - b_n)^2}}{L^2} db_n \\ & + N \frac{\pi a^2 (1 - \frac{\beta}{\pi}) + a(\frac{L}{2} - a - b) \sin \beta}{L^2} \cdot \frac{\langle w_n(t-1) \rangle}{\langle k_n(t) \rangle} \end{aligned} \quad (5.8)$$

where $\beta = \arccos[(L/2 - a - b)/a]$, $\langle w_n(t-1) \rangle = \xi(v) \cdot \langle k_n(t) \rangle + \eta(v)$, and $\langle k_n(t) \rangle = \sum_{k_n(t)=1}^{N-2} k_n(t) \cdot P_k(k_n(t))$. The first term on the right side is the contribution from the neighboring nodes with their communication circles partially cutting off by the boundary, while the second term represents the contribution from neighbors with complete communication circles.

5.4.3. Self-Consistent Equation to Estimate the Number of Extreme Events in ER Random Networks

For high value of P_R in the ER random network, the number of extreme events, n_{ex} , can be solved analytically via the approach of self-consistent equation. Regarding P_R as corresponding to some kind of “velocity” v , we have that the number of extreme events n_{ex} for $v = 0$ is given by

$$n_{\text{ex}} = NT \sum_{k=0}^{N-1} \left[\sum_{w=q+1}^W \binom{W}{w} \left(\frac{k}{2E} \right)^w \left(1 - \frac{k}{2E} \right)^{W-w} \right] P_k(k), \quad (5.9)$$

where q is the threshold by which extreme events are defined, and the degree distribution $P(k)$ of the network can be written in a binomial form:

$$P_k(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}. \quad (5.10)$$

To map the ER random network into a two-dimensional mobile network, we set the link probability to be $p = \pi a^2 / L^2$. Eq. (5.9) yields $n_{\text{ex}} \simeq 125$, which matches the simulation result for $v = 0$ well.

For high velocity, the PDF of flow $P(w)$ becomes independent of k and identical for all nodes, which can be solved analytically by the self-consistent equation method. In particular, the number z of packets delivered to a node of degree k from one of its neighbors of degree k_n has the following PDF

$$P(z) = \sum_{k_n=1}^{N-2} \left[\sum_{w=z}^W P(w) \cdot \binom{w}{z} \left(\frac{1}{k_n} \right)^z \left(1 - \frac{1}{k_n} \right)^{w-z} \right] P_k(k_n), \quad (5.11)$$

where $P_k(k_n)$ is the degree distribution of the neighboring node given by Eq. (5.10), and $P(w)$ can be obtained via the weighted sum of the convolution of $P(z)$:

$$P(w) = \sum_{k=0}^{N-1} [P(z_1) * P(z_2) * \dots * P(z_k)] \cdot P_k(k). \quad (5.12)$$

By substituting any form of $P(w)$ that satisfies $\sum_{w=0}^W w \cdot P(w) = W/N$ into Eq. (5.12), we obtain an updated form of $P(w)$ via Eq. (5.11). The solution of $P(w)$ can be obtained by iterating the

self-consistent equation until the root-mean-square deviation between $P(w)$ and its updated version falls below a small threshold ϵ (e.g., $\epsilon = 10^{-4}$). Figure 28a shows a good agreement between $P(w)$ from direct simulation and from the self-consistent approach. The number of extreme events can then be estimated, yielding $n_{\text{ex}} = NT \sum_{w=q+1}^W P(w) \simeq 238$, which agrees quite well with the value from direct simulation.

5.4.4. Effect of Degree-Degree Correlation

The probability p_{Δ} that a given node i and one of its neighbors j share a neighbor in the ER-random network of link probability p is p^2 . For a mobile network in a two-dimensional domain, this probability is given by

$$p_{\Delta} = \int_0^a \frac{S_{ij}(r)}{\pi a^2} \cdot \frac{2\pi r}{L^2} dr, \quad (5.13)$$

where r denotes the distance between node i and its neighboring node j , and $S_{ij}(r)$ is the overlapping area of i and j 's communication circles, which is given by

$$S_{ij}(r) = a^2 \arccos\left(\frac{r}{2a}\right) - r\sqrt{a^2 - \frac{r^2}{4}}, \quad (5.14)$$

Numerical calculation reveals that $p_{\Delta} \gg p^2$, indicating that a node's degree is much more likely to have a value similar to that of its neighbor, giving rise to a strong correlation between the degrees of neighboring nodes, or the phenomenon of degree-degree correlation. Namely, node i 's degree k is correlated with its neighbor's degree k_n , the so-called k - k_n correlation. In addition, the degrees of the neighboring nodes of i are also correlated, leading to the so-called k_n - k_n correlation. Since the effect of cutoff of the communication circle by a boundary makes the analysis on spatial degree correlation extraordinarily complicated, here we neglect the effect of the boundary or, equivalently, we focus on mobile networks with periodic boundary conditions.

For convenience, we classify the neighbors of node j into two classes with respect to the focal node i : n_{j1} neighbors in the overlapping area $S_{ij}(r)$ (i.e., the regions shared by these nodes and i)

and the remaining $n_{j2} = k_j - n_{j1}$ neighbors in the non-overlapping area. The PDF of n_{j1} is

$$P(n_{j1}) = \binom{k_i(t) - 1}{n_{j1}} p_1^{n_{j1}} (1 - p_1)^{k_i(t) - 1 - n_{j1}}, \quad (5.15)$$

where $p_1 = S_{ij}(r)/(\pi a^2)$. The number n_{j2} of neighbors in the non-overlapping area has the PDF as

$$P(n_{j2}) = \binom{N - k_i(t) - 2}{n_{j2}} p_2^{n_{j2}} (1 - p_2)^{N - k_i(t) - 2 - n_{j2}}, \quad (5.16)$$

where $p_2 = (\pi a^2 - S_{ij}(r))/L^2$. The degree distribution of node j is given by

$$P_k(k_j) = \int_0^a \sum_{n_{j1} + n_{j2} = k_n} P(n_{j1}) \cdot P(n_{j2}) \cdot \frac{2\pi r}{\pi a^2} dr \quad (5.17)$$

where $k_j = n_{j1} + n_{j2}$. In the high-velocity regime, the self-consistent equation method can be adopted to solving $P(w)$ by substituting $P_k(k_n)$ from Eq. (5.11) into Eq. (5.17), while the quantity $P_k(k)$ in Eq. (5.12) can still be determined by Eq. (5.10). Figure 28b shows the flow distribution $P(w)$ obtained from direct simulation and from self-consistent theory with or without the k - k_n correlation. We observe a better agreement between simulation and theory when the k - k_n correlation is taken into account.

It is mathematically difficult to consider both the k - k_n and k_n - k_n correlations. Nonetheless, since the spatial degree correlation depends solely on the spatial distribution of the nodes, which is completely random regardless of the velocity, the combined effect of the two kinds of correlations on n_{ex} has little dependence on the actual value of the velocity. Consequently, the non-monotonous behavior of extreme events caused by the boundary effect should persist even in realistic situations where both types of degree-degree correlations are present, and we expect the theory of mobility-based control developed in the main text to be generally valid.

6 . RESILIENCE OF INTERDEPENDENT COMPLEX NETWORKS - EMERGENCE OF EXTREME EVENTS AND CONTROL

6.1. Introduction

Internal resource competitions are a ubiquitous phenomenon in modern infrastructures, but relatively little attention has been paid to its impact on the dynamical evolution and resilience of the underlying interdependent, multilayer networked systems. Examples are the airport and train-station networks of the public transportation system, the base station network of the cellular communication system, and the virtual networks based on software defined networks (SDNs). For example, in the airport network, different airlines cover different subsets of the airports across the whole country, and airlines operate in the same airport have to share or even compete for the limited resources such as space and time. In the near future, different communication service providers may share base stations, signifying potential competitions for the finite bandwidth and processing capability of the based stations. The next generation of Internet may be built upon the framework of SDN, enabling one same physical server network to be virtually separated into multiple virtually independent sub-networks with scalable sizes, each serving or being operated by a particular user without interfering other subnetwork layers. In this case, the same server may carry the load generated by multiple layers (users), and its finite processing capacity is competed by the layers of virtual servers. All these call for a systematic understanding of the resilience of multilayer, interdependent networks subject to internal resource competitions. The goals of this paper are to develop a model capable of capturing the key topological and dynamical features of the multilayer infrastructures in the real world with inter-layer resource competitions, to study the extreme event dynamics from the standpoint of resilience, and to articulate control strategies to enhance the resilience against global-scale, catastrophic breakdown of the whole network system.

Extreme events associated with transportation dynamics on networks are intimately related to nodal flux fluctuations [48, 177, 178, 180, 182, 183, 196]. In previous models of single-layer transportation networks, extreme events tend to occur independently in space and time, which often have little effect on the system resilience [48, 49, 184, 185]. However, for an interdependent network with multiple interacting layers, fluctuations can induce qualitatively distinct phenomena in the system that is intrinsically nonlinear. Recent works on interdependent networks have uncovered a rich variety of phenomena associated with network fragility and robustness [94–100], diffusion and spreading processes [85–89], game dynamics [90–93], and synchronization [81–84]. Other related works range from redefining the basic structural measures to understanding the impacts of the multilayer nature of the network on dynamical processes. In spite of the tremendous efforts, to our knowledge there were no prior models of interdependent networks with multiple layers defined according to resource competitions, let alone any study of the extreme event dynamics in such networks.

In this paper, utilizing transportation as a prototypical dynamical process, we develop an inter-layer traffic resource competing mechanism to characterize the real-world situation where different network layers, corresponding to different social entities, coexist under limited capacities of public transportation, power, information, or telecommunication service facilities. A striking finding is that, when the number of network layers and/or the overlap among layers are increased, extreme events can emerge in a cascading manner to trigger global-scale catastrophes, even when the capacity is capable of accommodating the same number of independent layers without extreme events. The system typically evolves into one of the two stable absorption states: a state free of extreme events and a state with frequent occurrence of extreme events. We find that the transition between the two states is abrupt in both time and parameter domains. The finding indicates that internal interactions in a multiplex network system can yield qualitatively distinct phenomena. An impli-

cation is that public resource competitions among different service providers can lead to higher resource requirement than anticipated. We derive an analytical theory to understand the emergence of global-scale extreme events based on the concept of effective betweenness that we specifically introduce to describe the extreme event dynamics on interdependent networks. We also articulate an efficient control scheme based on augmenting the capacity of very few hubs, which can dramatically suppress the cascading process of extreme events and protect the entire multi-layer infrastructure against global-scale breakdown.

6.2. Results

We model the entire public service facility in a large infrastructure as a complex network G of N nodes. Among the M service providers, each provides packet transportation service on a subnetwork $G_m (1 \leq m \leq M)$ of $N_m (N_m \leq N)$ nodes connected via the links of G . The entire network G thus has M layers, each being the subnetwork operated by one service provider, as shown schematically in Fig. 29. For convenience, we call G the global network. Packets belonging to layer G_m have their origins and destinations solely within G_m : they can only be transported within G_m through its nodes and links, not to any other layer. Different layers can share common nodes, and the number of layers sharing node i is denoted as M_i .

The load of node i in the global network G is defined as the total number of packets from all the layers containing node i , i.e., $f_i = \sum_{m=1}^M w_{mi}$, where w_{mi} is the flux in layer m , among which there are M_i nonzero values. If f_i reaches its capacity C_i , we say that an extreme event has occurred on node i . The capacity C_i can then be defined as [48]

$$C_i = \langle f_i \rangle + \alpha \cdot \sigma_i, \quad (6.1)$$

where $\langle f_i \rangle$ is the average load, σ_i is the standard-deviation of f_i , and $\alpha > 1$ is the capacity parameter. Following Ref. [48], we set $\alpha = 4$ (somewhat arbitrarily) in our study. In fact, variation

Fig. 29. Schematic Illustration of a Multilayer, Interdependent Network Subject to Internal (Inter-layer) Resource Competitions. Layers 1 (G_1) and 2 (G_2) are two different subsets of Layer 0 (G , or the global network), which is solely used for model generate and does not participate in the transportation dynamics. In this schematic example, $N = 10$ (number of node in Layer 0), $M = 2$ (the number of interdependent layers), and $P_m = 0.5$ ($m = 1$ and 2). Accordingly, $N_m = N \cdot P_m = 5$, and correspondingly, the node degrees in Layers 1 and 2 are 0.5 time of their degrees in Layer 0 (fluctuation exists due to randomness). The combination of Layers 1 and 2 forms a multilayer interdependent network. Nodes 5 and 6 exist in both Layers 1 and 2, and thus, the two layers would compete for resources within each of these two overlapped nodes. Nodes 4 and 9 exist in neither of the two layers. Each of all the other nodes exist in only one of the two layers. (See the Methods section for model details.)

Fig. 30. Extreme Event Occurrence and Packet Arrival Rates. (a) Extreme event occurrence rate R_{EE} and (b) packet arrival rate R_A versus the number of layers, M , for systems with $P_m = 0.25$ (circles), 0.5 (squares), 0.75 (diamonds), and 1 (triangles), respectively. (c) R_{EE} and (d) R_A versus P_m for $M = 3$ (diamonds), 5 (circles), and 10 (squares), respectively. The results are obtained through 20 simulation realizations for each of the 10 network realizations. Each network has the average degree $\langle k \rangle = 4$ and size $N = 1000$. Each realization runs for 500 time steps. The quantities R_{EE} and R_A are averaged over the last 300 steps (typically, the system evolves into equilibrium within 200 time steps).

in α , insofar as it is larger than unity, will not affect the results qualitatively. This setting allows the capacity of node i to be adjusted according to the number of layers M_i sharing it. For simplicity, we assume that all layers share the same nodal coverage, i.e., $P_m = N_m/N$, where $P_m = 1$ corresponds to the special case where every layer is identical to the whole network, i.e., $G_m = G$.

The rules for packet transport/delivery are as follows. Within each layer, at each time step, $\mathcal{G} \cdot N_m$ new packets are generated, where $\mathcal{G}$ is the packet generation rate. The newly generated packets have randomly assigned destinations and are originated from randomly selected nodes that are free of any extreme events at the time. On a node, each previously generated packet is transported to a neighbor of the node along the shortest path towards the packet's destination. If no extreme event occurs on the target neighbor at this time, the movement can be completed. If, however, there is an extreme event on the desired spot that the packet is supposed to move into, the packet is transported to a randomly chosen neighbor with no extreme event, provided such a neighbor exists. If all the neighbors are currently having extreme events, the packet will remain at the original node.

6.2.1. Global Extreme Events and Two Absorption States

To quantify the transportation dynamics on the multilayer network, we use two quantities: (1) the extreme event occurrence rate R_{EE} , defined as the fraction of nodes at which extreme events occur at each time step, and (2) the packet arrival rate R_A defined as the ratio between the numbers of packets arriving and newly generated. Figure 30 shows the behaviors of R_{EE} and R_A for various numbers of layers and nodal coverage values. As shown in Figs. 30(a) and 30(b), R_{EE} for a single layer network ($M = 1$) is close to 0%, and R_A is 100%, implying that the capacity of a single layer network is sufficient to handle all the load in the network. As the number of layers is increased, the capacity is increased accordingly. If the layers are independent of each other, the total capacity can accommodate all the load from all layers, ruling out extreme events. However, when the layers are

interdependent, there is a dramatic increase in R_{EE} , indicating much higher total load generation due to internal resource competitions among the layers. The competitions drive the load of many nodes to their capacities, finally leading to the occurrence of extreme events at the global scale, i.e., $R_{EE} = 100\%$ and $R_A = 0$. Further computations reveal another phenomenon: after the system evolves into an equilibrium, only two types of steady absorption states can arise. They are (1) a *free state* nearly free of extreme events ($R_{EE} \geq 0$), and (2) a *catastrophic state* in which every single node has an extreme event. The surprising feature is that there are no stable intermediate states in between the two cases. In fact, we find numerically that any intermediate state must eventually evolve into one of these two absorption states. The value of R_{EE} thus represents the probability of the occurrence of the catastrophic state.

6.2.2. First-Order Phase Transition

As shown in Figs. 30(a) and Figs. 30(b), the curves corresponding to higher P_m values have systematically higher R_{EE} values, indicating that P_m has an effect on the probability of the catastrophic state. Figures 30(c) and 30(d) are further results of how R_{EE} and R_A vary with P_m when the number of layers (M) is fixed. A higher value of P_m means that it is more likely for two different layers to share common nodes, signifying a higher degree of interdependence with more severe internal competitions. As a result, R_{EE} (or R_A) increases (or decreases) monotonically with P_m . A striking behavior occurs for relatively large values of M , where R_{EE} (R_A) exhibits an abrupt increase (decrease) as P_m passes through a critical point, giving rise to a first-order phase transition. In fact, about the transition point, the probability for the system to evolve into a catastrophic state can show dramatic change, i.e., from 0 and 100% or vice versa, meaning that an arbitrarily small change in P_m can drive the system into catastrophe. The value of P_m thus has a critical impact on the emergence of extreme events on the global scale. The implication is that, when the infrastructure

Fig. 31. System's Time Evolution Towards a Catastrophic State. For a multilayer network of $N = 1000$ nodes, $M = 3$ layers, and $P_m = 1$, (a) the transition between a free (blue) and a catastrophic states (red) in time for all nodes. (b) The ratio of node i 's load f_i to its capacity C_i versus time for all nodes. (c) Time traces of the total number of packets in the layer (denoted as W , thick solid line), of the number of “stuck” packets due to the surrounding extreme events (W_{stop} , triangles), and of the number of movable packets (W_{jump} , squares). Among the W_{jump} movable packets, the number of those that effectively execute random walks (W_{rand} , red dashed line) and the number of the remaining packets moving along their respective shortest paths (W_{short} , blue dot line) are also plotted versus time.

consists of multiple, interdependent layers, its resilience against catastrophe may exhibit a sensitive dependence on the system parameter, such as P_m , whose critical value depends on the number of layers. In the design of infrastructural systems, it is then necessary to construct a detailed multilayer network model to estimate the critical parameter region of the first-order phase transition, and the choices of the parameters should be such that they are far away from the critical region with large margins.

6.2.3. Time Evolution and Packet Lifetime

In order to understand the emergence of extreme events at the global scale, we investigate the time evolution of various dynamical quantities. Figure 31(a) shows the time evolution of the states of

all nodes in the network, where at any time, each node can be in one of the two possible states: free (blue) and catastrophic (red). Figure 31(b) shows the relative load of each node, f_i/C_i , versus time. In both panels, a sudden transition can be seen immediately before $t = 150$, at which the entire population of nodes is taken over by extreme events almost simultaneously. After the transition, transportation dynamics in the entire system is completely stalled, leading to a catastrophe.

If no extreme event occurs throughout the system, each packet would follow the shortest path towards its destination without any delay so that its lifetime τ equals the length L of the path. In the equilibrium, the expected total number of packets accommodated in layer m is given by (see **Methods** for a detailed derivation)

$$W_m = N_m \mathcal{G} \cdot \langle \tau \rangle = N_m \mathcal{G} \cdot \langle L \rangle, \quad (6.2)$$

where $\langle \tau \rangle$ is the average packet lifetime and $\langle L \rangle$ is the average shortest path length of the layer. An extreme event occurring at the packet's next target node along the shortest path is most likely to increase the lifetime τ by causing the packet to take one step onto a randomly selected neighbor of the current node (carrier), because the probability is small for the randomly selected neighbor to be located along another shortest path of the same length. In the case where extreme events occur on all current carrier's neighbors, the packet will stop moving and wait until at least one of the neighboring nodes becomes free of extreme events. This scenario will also cause an increase in τ . Figure 31(c) shows some quantities characterizing packet movements in one layer versus time as the system evolves from a free state into a globally catastrophic state. At each time step, the total number of packets in the layer, W , equals the sum of W_{jump} , the number of moved packets, regardless of whether the packets are along the shortest path, and W_{stop} , the number of stuck packets due to the extreme events in the neighborhood. The quantity W_{jump} is equal to the sum of W_{short} , the number of packets that have moved along their corresponding shortest paths, and W_{rand} , the

number of packets that are stuck through random walks. Based on these considerations, we can write

$$W = W_{\text{stop}} + W_{\text{jump}} = W_{\text{stop}} + W_{\text{short}} + W_{\text{rand}}. \quad (6.3)$$

Until a global catastrophe takes place, W_{stop} is close to zero, W_{short} increases slightly but W_{rand} increases much faster with time. The behaviors of W_{jump} and consequently W are thus dominated by that of W_{rand} , demonstrating the key role played by W_{rand} : a few extreme events lead to random walks that lead to an increase in $\langle \tau \rangle$ and consequently an increase in W . The larger number of packets in turn causes the total load to increase, enhancing the probability of extreme events, which further generate more random walks. This self-stimulating, positive feedback type of mechanism is responsible for driving the whole system into the catastrophic phase.

For a single-layer network, a catastrophic phase cannot occur. In this case, P_{L_d} , the fraction of packets that are L_d steps away from their destinations, are constant in the equilibrium. Assuming that packets corresponding to each L_d value have the same probability η to be driven into random walks, we can solve both P_{L_d} and η analytically by using the principle of detailed balance from statistical physics (see **Methods**).

6.2.4. Positive Feedback Loop

Due to the internal resource competitions, the following process emerges: at a time step, the resource requirement in layer m at node i is high, and a substantial portion of node i 's capacity is devoted to layer m . As a result, other layers sharing node i will have to redistribute their load onto i 's neighbors, which may squeeze the capacity available for layer m of these neighbors, which may have to redistribute their load back onto node i in layer m , and so on. This mechanism can trigger a cascading process and finally drive the system into a global-scale catastrophe. The cascading process is more likely to take place in systems with more layers.

Fig. 32. Numerically Obtained and Theoretical Predicted Probabilities of Extreme Events. For $M = 1$ (circles), 3 (squares), 5 (diamonds), and 10 (triangles), (a) numerically obtained probability $q_{EE}(k)$ for an extreme event to occur on a node of degree k and (b) theoretical prediction based on the concept of betweenness. (c) For $P_m = 1$, the probability of extreme events versus M , where the degree distribution function $P(k)$ used in the calculation of q_{EE} is $P(k) = k^{-3} / \sum_{k=k_{\min}}^{k_{\max}} k^{-3}$ for $k_{\min} = 2$ and $k_{\max} = 75$ (typical values from network realizations). In fact, small changes in the values of $k_{\min}$ and $k_{\max}$ do not affect the main features of the results.

6.2.5. Betweenness and Effective Betweenness

It is difficult to develop a rigorous mathematical theory to fully grasp the mechanism of the cascading process. Our aim is to provide a physical understanding. In fact, the probability for an extreme event to occur on a node can be *quantitatively* analyzed through estimating the nodal flux distribution function using the concept of nodal betweenness [169], denoted as $B(i)$. Previous works [48,49,163,180,183] established that, for a single-layer transportation network, under random routing the probability for a node to be visited is proportional to its degree, but for the shortest-path routing scheme, the probability is proportional to the number of shortest paths through the node. Motivated by these results, we hypothesize that the probability for node i to be visited under shortest path routing is determined by the following normalized betweenness

$$p_B(i) = \frac{B(i)}{\sum_{j=1}^N B(j)}. \quad (6.4)$$

Fig. 33. Probability of Extreme Events Calculated From Our Theory of Effective Betweenness. (a-d) The probability $q_{EE}(k)$ for an extreme event to occur on a node of degree k obtained from simulation (circles), the theory of betweenness (diamonds), and the theory of effective betweenness (squares) for $M = 1, 2, 3$, and 5 , respectively. In (e), the extreme-event probability versus M is shown, obtained from simulation (circles), the theory of betweenness (diamonds), and the theory of effective betweenness (squares), for $M = 10$ and $P_m = 1$. The degree distribution function $P(k)$ used to calculate q_{EE} is obtained using 10 network realizations. The betweenness and effective betweenness for each node of degree value k are calculated from the corresponding network topology and averaged over all such nodes among the 10 network realizations.

Under the shortest path routing scheme, the transportation dynamics can in fact be regarded as a random-walk process with nodal visiting probability $p_B(i)$. If packets are uniformly distributed on various paths of different length, the betweenness centrality scales with the nodal degree in a power-law fashion [169, 197]:

$$p_B(i) \approx \beta k_i^\gamma, \quad (6.5)$$

where β is a normalization constant, γ is the power-law scaling exponent, and both β and γ can be determined numerically. This relationship holds for most degree values (see **Methods**), providing a basis for our physical analysis.

Consider now a multilayer network. For layer m , the probability density function (PDF) of the flux w_{mi} through node i is binomial:

$$P_i(w_{mi}) = \binom{w_{mi}}{W_m} [p_B(i)]^{w_{mi}} [1 - p_B(i)]^{W_m - w_{mi}} = \binom{w_{mi}}{W_m} (\beta k_i^\gamma)^{w_{mi}} (1 - \beta k_i^\gamma)^{W_m - w_{mi}}. \quad (6.6)$$

For simplicity, we set $P_m = 1$ so that all layers are identical. In this case, we can replace W_m with $\langle W \rangle$ so that the number of layers sharing node i is simply $M_i = M$. The total load on node i from the M layers, $f_i = \sum_{m=1}^M w_{mi}$, can then be written as a sum of M binomial random variables with identical distributions as given by Eq. (6.6). During the dynamical evolution, the interdependence among the M random variables is reflected through the inequality $f_i \leq C_i$. It is useful to calculate the probability of $f_i = C_i$, the criterion for an extreme event to occur on node i . If we set the system free by removing the capacity bound of every node and allowing f_i to increase indefinitely, the probability will essentially be given by $P(f_i \geq C_i)$, where f_i is the sum of M independent,

identically distributed binomial random variables. The PDF of f_i is then given by

$$P_i(f_i, M) = \binom{f_i}{M \cdot \langle W \rangle} (\beta k_i^\gamma)^{f_i} (1 - \beta k_i^\gamma)^{M \cdot \langle W \rangle - f_i}. \quad (6.7)$$

The probability for an extreme extent to occur on a node of degree k is

$$q_{\text{EE}}(k, M) = \sum_{f_k=C_k}^{M \cdot \langle W \rangle} \binom{f_k}{M \cdot \langle W \rangle} (\beta k^\gamma)^{f_k} (1 - \beta k^\gamma)^{M \cdot \langle W \rangle - f_k}, \quad (6.8)$$

where f_k and C_k denote the load and the capacity of a node of degree k , respectively. As shown in Figs. 32(a) and 32(b), this analysis captures the qualitative behavior of the system from simulation. We further see that, for nodes of relatively high degrees, $q_{\text{EE}}(k)$ for high values of M (or k) is generally larger than those for lower values of M (or k), revealing that extreme events tend to occur on nodes shared by more layers and of larger degrees. This result should be contrasted to the case of single layer networks, where extreme events tend to occur more on small degree nodes, e.g., under the random routing scheme [48]. Thus, in multilayer networks, the hub nodes play a crucial role in generating extreme events. Based on this, we can write the probability for an extreme event to occur in a M -layer system as

$$P_{\text{EE}}(M) = \sum_{k=1}^{N-1} P(k) \cdot q_{\text{EE}}(k, M), \quad (6.9)$$

where $P(k)$ is the degree distribution of the entire network. Figure 32(c) reveals a monotonically positive correlation between $P_{\text{EE}}(M)$ and M , providing a qualitative explanation for the more frequent occurrence of extreme events in systems with more layers.

If we calculate $p_{\text{B}}(i)$ using the actual betweenness values for all the individual nodes and then averaging the nodes of the same degree (which can be done based on the topological information of the network) instead of using the approximation Eq. (6.5), we find systematical deviation in $q_{\text{EE}}(k)$ as given by theory from that through simulation, as shown in Figs. 33 (a-e). Empirically, we are able

to identify a quantity to replace the betweenness centrality in determining $q_{\text{EE}}(k)$ and $P_{\text{EE}}(M)$. We name it *effective betweenness*, defined for node i as

$$B_{\text{E}}(i) = N_m + \sum_{js} \frac{n_{js}}{L_{js}}, \quad (6.10)$$

where L_{js} is the length of a shortest path through i between the origin j and the destination s ($j \neq s$), and n_{js} is the number of such paths. While a rigorous justification for the effective betweenness is difficult, it gives a better agreement between theory and numerics. An intuitive understanding is the following. For $j \neq s$, i.e., $L_{js} > 0$, since the lengths of these shortest paths are proportional to the lifetime of the packets, the packet assigned to a longer path L_{js} has a longer lifetime. Along the path, the packet occupies each node once, thereby contributing less to the visiting probability of a given node in each time step. The term N_m counts for the contribution from the case of $j = s$, i.e., $L_{js} = 0$.

Using the concept of effective betweenness, we obtain the probability of a node to be visited as

$$p_{\text{E}}(i) = \frac{B_{\text{E}}(i)}{\sum_{j=1}^N B_{\text{E}}(j)}. \quad (6.11)$$

More justifications for the use of the effective betweenness can be obtained through simulations (see **Methods**).

In a similar manner, we can calculate $q_{\text{EE}}(k)$ and $P_{\text{EE}}(M)$ through $p_{\text{E}}(i)$ instead of $p_{\text{B}}(i)$. Figures 33(a-e) show, for a number of different settings, good agreement between the predicted values of $q_{\text{EE}}(k)$ with the numerical values. In fact, the predicted values of $P_{\text{EE}}(M)$ through B_{E} also match the simulation results very well, as compared with those through B , as shown in Fig. 33(f). These results indicate that the effective betweenness is a key quantity characterizing the transportation dynamics on multilayer networks subject to interlayer competitions under the standard shortest path routing protocol. Utilizing this concept, the transportation dynamics can effectively be mapped into a binary stochastic process for further theoretical development.

Fig. 34. Occurrence Rate of Extreme Events and Packet Arrival Rate Under Control. In the control parameter space, n_{top} denotes the number of top-degree hubs whose capacities are augmented, and r_i is the ratio of the enhanced capacity to the original capacity. The extreme event occurrence rate R_{EE} and the packet arrival rate R_A in the parameter space are shown, respectively in (a) and (b). The simulation parameters are the same as in Fig. 30.

6.2.6. Control Strategies to Suppress Extreme Events

From Figs. 30(c) and 30(d), we see that a small increase in the parameter P_m , when it is near the critical point, can lead to a transition between the free and catastrophic states. A straightforward control strategy is then to reduce the value of P_m . Since extreme events tend to occur on nodes shared by many layers, controlled reduction of the overlap between the layers can also suppress extreme events. However, these naive methods will not be effective when P_m has well passed the critical point.

We focus on the case where P_m , by design, exceeds its critical value by a large amount. One can increase the capacity of each node to prevent the occurrence of extreme events at a global scale, but this may be costly. According to Fig. 32(a), despite small fluctuations, extreme events take place on large degree nodes with a higher probability. A practical strategy is then to selectively enhance the capacities of the hub nodes. Figures 34(a) and 34(b) show, respectively, R_{EE} and R_A

when the capacities of the top n_{top} nodes (ranked by degree) in the network are multiplied by the factor $r_i > 1$. There exists a region in the parameter space that is completely free of extreme events, i.e., $R_{\text{EE}} \approx 0$ and $R_A \approx 1$, with a clear boundary separating this region from the catastrophic regions in the parameter space. We see that neither too small values of r_i nor small values of n_{top} can inhibit extreme events, implying that the extreme events occurring at a small set of hub nodes form a positive feedback loop through mutual stimulation and accordingly trigger global-scale cascading processes. We regard these hubs as constituting an *extreme-event core* (EE core), since they serve as the source of the global catastrophe. We see that extreme events can be effectively eliminated through providing reasonably more resources to nodes in the EE core. For example, solely increasing the capacities of the top 5 hubs (5 out of 1000, less than 1%) by 1.7 times can make the entire network system immune to any global cascade. The enhanced capacity is in fact insignificant comparing with the total capacity of the system, but the targeted capacity enhancement can improve the network resilience significantly by drastically reducing the probability of extreme-event cascade. For a multilayer network system of infrastructure, it is thus important to identify the EE core and to assign larger capacities to the nodes in the core.

6.3. Discussion

The ubiquitous phenomenon of resource competitions in real world systems motivates us to articulate and study a class of interdependent complex networks in which a set of nodes representing, e.g., public service facilities in a large infrastructure, are shared by different layers that correspond to, e.g., different service providers. The set of shared nodes have relatively large capacities. We find that internal competitions for common resources have a dramatic and sometimes devastating effect on the network transportation dynamics. In particular, as the number of network layers is increased, extreme events in which the delivering capabilities of certain nodes in the network are

essentially depleted, can occur in a cascading manner, leading to the catastrophic occurrence of such events on a global scale. This should be contrasted to the case of total absence of extreme events in systems with the same nodal capacities but without internal competitions (so that the layers are independent). A striking phenomenon is that, there are two distinct possible asymptotic states for the system: (1) a state free of extreme events and (2) a state completely dominated by extreme events. Varying one of the two key structural parameters, the number of layers and the nodal coverage rate, a first-order phase transition can occur between the two states, meaning that the system can change abruptly from one state to another as a control parameter passes through a critical point. We develop a physical theory to understand the dynamics of the extreme events based on a newly defined topological property, the effective betweenness, and the empirical scaling law for betweenness centrality in general. To suppress extreme events and enhance the resilience of the system against global-scale breakdown, we propose an effective and low-cost control strategy, the articulation of which benefits from the finding of an *extreme events core* formed by a small number of hub nodes, which plays a critical role in “spreading” the extreme events. When the capacities of the nodes in the core are selectively augmented, the network’s ability to resist large-scale extreme events can be enhanced significantly.

Our findings have potential applications in gaining insights into the resilience of large scale infrastructural systems that are typically composed of many layers with shared public service facilities. Our results indicate that competitions for public resources can lead to catastrophic behaviors, and control is necessary to make the system resilient to large scale failures. Generally, control of extreme events in interdependent networked system is an important and challenging problem, and we hope our work to stimulate further efforts.

6.4. Methods

6.4.1. Model Details

For a given network G (Layer 0, shown as Fig. 29), each of the M layers of subnetworks is generated via the following procedure: (1) randomly select a node in G ; (2) randomly select its P_mk neighbors; (3) for each selected neighbor, do (2) and repeat the step for all newly selected nodes until the total number of selected nodes reaches P_mN . This procedure fixes the average degree of each layer to be $P_m\langle k \rangle$. To be concrete, we use the Barabasi-Albert (BA) type of scale-free topology [112] for network G . The power-law exponent of the degree distribution for each layer is approximately the same as that for the entire network G .

Due to randomness in the packet transportation process, the flux of node i in layer m , w_{mi} , is a random variable. Accordingly, f_i is the sum of M random variables (with M_i of those being nonzero). If different layers are independent and packet transportation process can be described as a random walk, we have

$$\langle w_{mi} \rangle = W_m \frac{k_{mi}}{2E_m} \text{ and } \sigma_i^2 = W_m \frac{k_{mi}}{2E_m} \left(1 - \frac{k_{mi}}{2E_m}\right)$$

where W_m is the total number of packets in layer m , k_{mi} is the degree of node i in layer m , and E_m is the total number of links in layer m . Hence, we have

$$\langle f_i \rangle = \sum_{m=1}^M \langle w_{mi} \rangle = \sum_{m=1}^M \frac{W_m k_{mi}}{2E_m} \text{ and } \sigma_i = \sqrt{\sum_{m=1}^M \sigma_{mi}^2} = \sqrt{\sum_{m=1}^M \frac{W_m k_{mi}}{2E_m} \left(1 - \frac{k_{mi}}{2E_m}\right)},$$

where W_m can be determined numerically through a single layer simulation or be calculated analytically from Eq. (6.2).

Numerical simulations of the dynamical process of transportation are carried out, as follows. Initially no packets exist in the system. At each time step, $N_m\mathcal{G}$ newly generated packets with randomly selected destinations are placed on randomly selected nodes that are currently *free of*

extreme events. Any node with load equal to its capacity is not allowed to accept any packet from its neighbors, but the packets on the node can still leave the node, if at least one of its neighbors is not fully loaded. Asynchronous updating scheme is adopted in the simulation to ensure that no node has load higher than its capacity and also for the reason that synchronization in large infrastructure systems is not always realistic and necessary. (In fact, many systems are strongly asynchronous, e.g., the planes or trains usually enter an airport or a train station one after another.) The number of newly generated packets in the network is determined by the packet generation rate $\mathcal{G}$ and the layer population $N_m = P_m N$. Without loss of generality, we treat P_m as a tunable parameter while keeping $\mathcal{G}$ fixed. Variations in $\mathcal{G}$ do not affect the results qualitatively. The packet generation rate is set to be $\mathcal{G} = 0.1$ in our study.

6.4.2. Total Number of Packets in a Layer

In an equilibrium state without extreme events on a global scale, each packet can move freely along the shortest path towards its destination so that its lifetime τ is approximately equal to the length L of the path. Since $W_m(t)$ includes $W_m(t-1)$ and the newly generated $N_m \mathcal{G}$ packets with the number of arrivals subtracted off, we can derive the expression of W_m step by step. In particular,

Fig. 35. Number of Packets L_d Steps Away From Their Destinations. For a single layer network of size $N = 1000$, the number of packets in an equilibrium state that are L_d steps away from their respective destinations from theory (triangles) and simulation (circles). The total number of packets that the system can accommodate in the equilibrium state is about $W = 558$, which includes the newly generated packets within the current time step (diamonds) and all the remaining old packets (squares).

starting from $t = 1$, we have

$$\begin{aligned}
W_m(t=1) &= N_m \cdot \mathcal{G}, \tag{6.12} \\
\Rightarrow W_m(t=2) &= W_m(t=1) + N_m \mathcal{G} - N_m \mathcal{G} \cdot Q^{\text{new}}(L_d=1) = N_m \mathcal{G}[2 - Q^{\text{new}}(L_d=1)], \\
\Rightarrow W_m(t=3) &= W_m(t=2) + N_m \mathcal{G} - N_m \mathcal{G} \cdot Q^{\text{new}}(L_d=1) - N_m \mathcal{G} \cdot Q^{\text{new}}(L_d=2) \\
&= N_m \mathcal{G}[3 - 2Q^{\text{new}}(L_d=1) - Q^{\text{new}}(L_d=2)], \\
\Rightarrow W_m(t=4) &= N_m \mathcal{G}(4 - 3Q^{\text{new}}(L_d=1) - 2Q^{\text{new}}(L_d=2) - Q^{\text{new}}(L_d=3)), \\
\Rightarrow &\dots\dots, \\
\Rightarrow W_m(t=t') &= N_m \mathcal{G}[t' - \sum_{L_d=1}^{t'} (t' - L_d) Q^{\text{new}}(L_d)],
\end{aligned}$$

where $Q^{\text{new}}(L_d)$ denotes the stable fraction of the newly generated packets that are L_d steps away from their destinations. Apparently, if $L_d > L_d^{\text{max}}$, i.e., L_d is larger than the maximum shortest path length L_d^{max} in the focal layer, we have $Q^{\text{new}}(L_d) = 0$ and

$$W_m(t > L_d^{\text{max}}) = N_m \mathcal{G} \sum_{L_d=1}^{L_d^{\text{max}}} L_d Q^{\text{new}}(L_d) = N_m \mathcal{G} \langle L \rangle. \tag{6.13}$$

The relaxation time for the system to evolve into the stable equilibrium state is given by $t = L_d^{\text{max}}$.

We obtain $W_m(t \rightarrow \infty)$, as in Eq. (6.2). However, when extreme events take place in the system, the packet lifetime τ can assume values much larger than the path length L . In this case, we have $W_m = N_m \mathcal{G} \langle \tau \rangle > N_m \mathcal{G} \langle L \rangle$.

6.4.3. Effect of Rerouting Via Random Walk

For a single-layer network, we can develop a theory to explain the qualitative behavior of the transportation dynamics. In an equilibrium state, the probability $Q(L_d)$ for a packet to be L_d steps away from its destination satisfies the condition of detailed balance. For simplicity, we assume that each packet commits one step random walks with probability η , regardless of the distance

Fig. 36. Normalized Effective Betweenness and Free Flux Distribution Function. For a single layer network of size $N = 1000$, (a) the normalized effective betweenness of a node of degree k , p_E , versus k , from simulation (squares) and theory (line). The simulation results are obtained from 20 network realizations. (b) The free flux distribution function $P(w)$ for a node of degree 75 versus w obtained from simulation (squares) and theory (line), where the distribution function is meaningful only when the system is set to be free of extreme events, i.e., each node is set to have infinite capacity so that packets can travel freely via the shortest path towards their destinations. This way the theoretical probability for the node to be visited can be calculated.

from its destination. A walk makes a packet one more step away from its destination (which is numerically observed with high probability). Thus, at time t , packets corresponding to L_d consist of (1) the packets corresponding to $L_d + 1$ at time $t - 1$ and moved along the shortest path towards their destinations at time t , (2) the packets corresponding to $L_d - 1$ at time $t - 1$ and committing random walks at time t , and (3) the newly generated packets corresponding to L_d . Consequently, for $L_d < L_{L_d}^{\max} - 1$, we have

$$Q(L_d) = (1 - \eta)Q(L_d + 1) + \eta Q(L_d - 1) + Q^{\text{new}}(L_d). \quad (6.14)$$

Since, in a typical case, any movement of the packets corresponding to $L_d^{\max}$ makes their distances from the destinations smaller, packets for $L_d = L_d^{\max} - 1$ include all these that are $L_d^{\max}$ steps away from their destinations. We thus have

$$Q(L_d^{\max} - 1) = Q(L_d^{\max}) + \eta Q(L_d^{\max} - 2) + Q^{\text{new}}(L_d^{\max} - 1). \quad (6.15)$$

As boundary conditions, for packets corresponding to $L_d = L_d^{\max}$, we have

$$Q(L_d^{\max}) = \eta Q(L_d^{\max} - 1) + Q^{\text{new}}(L_d^{\max}), \quad (6.16)$$

and for packets corresponding to $L_d = 1$, we have

$$Q(1) = (1 - \eta)Q(2) + Q^{\text{new}}(1). \quad (6.17)$$

Consider the case where $L_d^{\max} = 7$ (a typical case in our study). We obtain a complete equation set for solving $Q(L_d)$, as follows:

$$\left\{ \begin{array}{l} (1 - \eta)Q(2) - Q(1) + Q^{\text{new}}(1) = 0 \\ (1 - \eta)Q(3) + \eta Q(1) - Q(2) + Q^{\text{new}}(2) = 0 \\ (1 - \eta)Q(4) + \eta Q(2) - Q(3) + Q^{\text{new}}(3) = 0 \\ (1 - \eta)Q(5) + \eta Q(3) - Q(4) + Q^{\text{new}}(4) = 0 \\ (1 - \eta)Q(6) + \eta Q(4) - Q(5) + Q^{\text{new}}(5) = 0 \\ Q(6) + \eta Q(5) - Q(6) + Q^{\text{new}}(6) = 0 \\ \eta Q(6) - Q(7) + Q^{\text{new}}(7) = 0 \end{array} \right. \quad (6.18)$$

It can be written in the matrix form

$$\begin{pmatrix} -1 & 1 - \eta & 0 & 0 & 0 & 0 & 0 \\ \eta & -1 & 1 - \eta & 0 & 0 & 0 & 0 \\ 0 & \eta & -1 & 1 - \eta & 0 & 0 & 0 \\ 0 & 0 & \eta & -1 & 1 - \eta & 0 & 0 \\ 0 & 0 & 0 & \eta & -1 & 1 - \eta & 0 \\ 0 & 0 & 0 & 0 & \eta & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & \eta & -1 \end{pmatrix} \begin{pmatrix} Q(1) \\ Q(2) \\ Q(3) \\ Q(4) \\ Q(5) \\ Q(6) \\ Q(7) \end{pmatrix} = - \begin{pmatrix} Q^{\text{new}}(1) \\ Q^{\text{new}}(2) \\ Q^{\text{new}}(3) \\ Q^{\text{new}}(4) \\ Q^{\text{new}}(5) \\ Q^{\text{new}}(6) \\ Q^{\text{new}}(7) \end{pmatrix}, \quad (6.19)$$

where

$$\sum_{L_d=1}^{L_d^{\max}} Q(L_d) = 1 \quad \text{and} \quad \sum_{L_d=1}^{L_d^{\max}} Q^{\text{new}}(L_d) = \frac{N_m \mathcal{G}}{W_m}.$$

This set of equations can be solved numerically. When the network topology is fixed, $Q^{\text{new}}(L_d)$ follows a constant distribution, which can be obtained from simulation. While each value of η generates a set of $Q(L_d)$ values, a proper value makes the sum of $Q(L_d)$'s unity. In our calculation, as η is increased from 0 to 1, we find that the sum of the $Q(L_d)$ increases monotonically and, hence,

only the η values that make $\sum_{L_d=1}^{L_d^{\max}} Q(L_d) = 1$ provides a solution. Figure 35 shows the theoretical solution of the $Q(L_d)$ distribution, which qualitatively agrees with the simulation result.

6.4.4. Effective Betweenness Calculation and Flux Distribution

According to Eq. (6.10), the effective betweenness B_E of each node can be calculated from information about the network topology, and the probability of a node to be visited (the normalized effective betweenness), p_E , can be obtained from Eq. (6.11). Through extensive simulations, we find that the effective betweenness under the condition that the contribution of packets is inversely proportional to the path length exhibits a power-law scaling with the degree. The empirical relationship between p_E and degree k is verified by comparing the averaged $B_E(k)/\sum_{k=1}^N B_E(k)$ values corresponding to each k value obtained via the topology information and the estimated values from $p_E(i) \approx \beta k_i^{-\gamma}$, as shown in Fig. 36(a). We see that, except the nodes with very low connectivity, the p_E value obtained directly from the network topology matches that obtained from the empirical relationship (the error bars are essentially invisible within the scale of the figure).

In our theory, the assumption that packets moving according to the shortest-path routing scheme can be regarded as equivalent to random walks with node-visiting probability proportional to the effective betweenness can be justified through the flux distribution function. As shown in Fig. 36(b), for a typical hub node, the flux distribution function obtained from theory, i.e., the binomial distribution in Eq. (6.6), correctly fits the statistics obtained from simulation. For certain nodes the theory and simulation results may not match well, but the theoretically predicted and numerically obtained peaks typically have a substantial overlap. Further calculations show that the flux distribution functions obtained from the classical betweenness centrality deviate from the simulation systematically with large errors, as there is little overlap between the peaks obtained from theory and numerics. This provides further justification for the necessity to use the effective betweenness.

7 . OPTIMIZING COOPERATION ON COMPLEX NETWORKS IN THE PRESENCE OF FAILURE

7.1. Introduction

Natural selection favors the survival and prevalence of species with competitive edge, yet the phenomenon of cooperation is ubiquitous in many biological, economical, and social systems [198, 199]. Understanding the emergence and evolution of cooperation has thus become a field of significant interdisciplinary interest, where evolutionary-game theory [200] has served as a powerful mathematical paradigm [20, 198]. In a typical setting, a number of agents on a network interact with one another, where the network topology can be regular or complex, and each agent can take on one of the two strategies at any given time: cooperation or defection. The defection strategy is a selfish action that usually generates higher payoff [200] temporally, as in paradigmatic games such as the Prisoner's Dilemma games (PDGs) [20], snowdrift games (SGs) [24], and public goods games (PGGs) [201]. A basic issue is then how cooperation can possibly survive when natural selection favors the defection strategy in order to gain higher individual fitness (at least temporally). In the past two decades, many cooperation-facilitating mechanisms have been uncovered, which include network reciprocity [202–204], reputation and punishment [205], random diffusion [206], success-driven migration [207], memory effect [208], benefit of noise [209–211], social diversity [22, 212], asymmetric cost [213], and teaching ability [214].

In most previous works, no death mechanism was incorporated in the evolutionary-game model on networks, i.e., no agent can be removed from the system, even if it gains no payoff in a substantial amount of time [215]. In real-world situations, an agent can go bankrupt and be eliminated immediately when its payoff falls below a critical threshold for certain period of time. An example is the great economical recession in 2008, where a large number of financial institutions and corporations

collapsed. In an ecological system, death of individuals is a common phenomenon. In this regard, a recent work [216] has incorporated a simple elimination mechanism into the evolutionary-game rules. In particular, a tolerance parameter was assigned to each individual in the network, which is the lowest allowed payoff. An agent dies and is removed from the network when its payoff falls below this threshold. The threshold can be heterogeneously distributed among agents. It was shown that rapid, cascading-like elimination of agents can result from such a death mechanism, and a pure cooperation state can emerge afterwards, where all defectors are eliminated and the survivors are exclusively cooperators [216]. One implication is that defectors, despite their advantages in getting temporarily higher payoffs, may be particularly vulnerable to large-scale, catastrophic failures. These findings thus suggest that, in a complex system where agents are subject to failure or death, cooperation may be beneficial to mitigating large-scale breakdown.

7.2. Results

In this paper, we propose a control scheme to enhance cooperation and eliminate large-scale failures in complex networked systems. Our key idea is that, due to the complex time evolution of the system, although the payoff of any agent can inevitably become arbitrarily low, the probability that the payoff remains low for an extended period of time will be small. We are thus led to introduce a *time tolerance* for each agent, where an agent will not die or be removed unless its payoff remains below a critical threshold for time longer than the tolerance. Since the degree distribution of the network is in general not uniform, it is reasonable that the time tolerance be degree-dependent. A parameter β can then be introduced to characterize the heterogeneity of the distribution of the time tolerance, where $\beta = 0$ signifies completely uniform distribution. Our main result is that properly chosen time delay can optimize cooperation and prevent large-scale death. A surprising finding is that optimal state of cooperation occurs near $\beta = 0$, indicating that making time-tolerance

distribution uniform is an effective strategy to enhance cooperation.

To impose a time tolerance on a complex network, we conceive that a node or an agent's *debt capacity* depends on its relative “importance” in the network. We thus hypothesize the following relationship between agent i 's time tolerance and its degree k_i [22,217]:

$$T_i = NT_0 \frac{k_i^\beta}{\sum_l k_l^\beta}, \quad (7.1)$$

where N is the total number of agents, T_0 is the nominal time tolerance, and β is an externally control parameter. For $\beta < 0$, agents with higher (lower) degree have lower (higher) time tolerance, the situation is the opposite for $\beta > 0$, and $\beta = 0$ corresponds to uniform time tolerance in the network. A large values of T_i means that the node is more resilient to failure or “bankruptcy.” A death mechanism can be introduced for PDG by choosing the following payoff tolerance for agent i [216] $P_i^T \equiv \alpha P_i^N = \alpha k_i$, where agent i dies and is removed from the network if its payoff is lower than P_i for consecutive T_i time steps, P_i^N is the normal payoff of agent i when the system is in a healthy state in which all agents are cooperators, and $0 < \alpha < 1$ is a tolerance parameter. Since an agent's degree may change when their neighbors die, k_i is the “instantaneous” degree of agent i . For $\alpha = 1$, agents have zero payoff tolerance to breakdown, while for $\alpha = 0$, agents are completely tolerant.

In our evolutionary game model, each time step (iteration) thus consists of the following four dynamical processes. (1) *Game playing and payoffs*. Each agent plays the classical prisoner's dilemma game (PDG) with all its nearest neighbors, and the total payoff is the sum of the payoffs gained in its two-player games with all other connected agents. The PDG parameters are chosen to be $R = 1$, $T = b > 1$, and $S = P = 0$ [20]. (2) *Strategy updating*. At each time step, agent i randomly chooses a neighbor j and imitates j 's strategy with the probability $W_{i \rightarrow j} = \{1 + \exp[-(P_j - P_i)/\kappa]\}^{-1}$, where P_i and P_j are the payoffs agents i and j , and κ is the level of

agents' "rationality" representing the uncertainties in assessing the best strategy. We set $\kappa = 0.1$.

(3) *Failure and agent removal*. At each iteration, for agent i , the time in debt t_i increases by 1, if P_i falls below the payoff tolerance P_i^T during the prior t_i time steps. Otherwise, we set $t_i = 0$. Since k_i varies with time, P_i^T and T_i also change with time. If $t_i > T_i$ or if $k_i = 0$, agent i and all its links will be removed from the network. (4) *Random rewiring*. For agent i whose neighbor j has been removed in step (3), a new connection is added between agent i and an randomly selected agent in the remaining agents outside i 's current neighborhood, provided that such an agent exists. This is motivated by the consideration that an agent in general will try to seek and engage new partners when the payoffs of some of its current partners become insignificant, and lack of global information leads to random selection. Note that, dynamical processes (1) and (2) are conventional for typical evolutionary-game dynamics, but processes (3) and (4) are unique features of our model [104].

In our simulations, we use the standard scale-free network model [105, 112] with parameters $m_0 = 5$, $m = 5$, and average degree $\bar{k} = 10$. The total number of agents is $N = 1000$. The initial condition is that both cooperative and defective strategies populate the scale-free network with equal probability. A synchronous updating scheme is adopted. For the results that will be presented below, the transient time is chosen to be 10000 iterations, and each data point is obtained by averaging over 25 independent network realizations with 20 runs for each realization.

To characterize the behavior of the network dynamics in the presence of death, we use three quantities: death rate S_d (the number of the dead agents in the final state normalized by the network size), and the asymptotic frequencies of cooperators ρ_C and defectors ρ_D , where $S_d + \rho_C + \rho_D = 1$. Figures 37(a-f) show, for six combinations of three values of the control parameter β and two values of the nominal time tolerance T_0 , the asymptotic values of the three characterizing quantities versus the profit-tolerance parameter α . We observe that, while in all cases, S_d increases and ρ_C

Fig. 37. (Color online.) Death rate S_d (red circles) and frequencies of cooperation ρ_C (blue squares) and defection ρ_D (green triangles) versus the tolerance parameter α for six combinations of three values of the control parameter β (-1 , 0 , and 1) and two values of T_0 (20 and 100) for $b = 1.5$.

Fig. 38. (Color online.) For four combinations of the parameter values of α and T_0 , S_d (red circles), ρ_C (blue squares), and ρ_D (green triangles) versus the control parameter β for $b = 1.5$.

decreases nearly monotonically with α , a large value of T_0 can delay the occurrence of total death. In fact, for $T_0 = 100$, there exists a region of small α values in which the death rate is nearly zero and vast majority of the agents are cooperators. For $T_0 \rightarrow \infty$, cooperators rule the whole region of $0 \leq \alpha \leq 1$. Comparing the three cases of β values, we see that this interval of sustainable cooperation appears to be the largest for $\beta = 0$, implying that adjusting β can have the effect of significantly promoting cooperation. Further investigation shows that during the cascading failure and random rewiring process, the degree distribution of the network can evolve from power-law to being Poisson eventually.

To demonstrate the ability of control strategy to optimize cooperation, we explore how the three characterizing quantities depend on the control parameter β . Figure 38 shows their behaviors for four combinations of two values of α and T_0 . We observe that ρ_C exhibits a non-monotonic behavior with β , indicating that ρ_C can be optimized by a proper choice of β and this optimization effect is stronger for larger value of T_0 [panels (b) and (d)]. For example, for $\alpha = 0.3$ and $T_0 = 100$ [panel (b)], for β near zero, the networked system becomes “super-cooperative” in the sense that almost no agent dies ($S_d \approx 0$) and nearly all agents survive as cooperators ($\rho_C \approx 1$). The results do not depend on the choices of b and κ . The revelation that the optimal value of β occurs near zero is striking: it indicates that distributing time-tolerance uniformly can be tremendously advantageous to promote and sustain cooperation on the networked system, despite the highly heterogeneous character of the network topology. This finding has certain implication for policy making. For example, an economical network constituting financial institutions and banks of various sizes as its nodes may be subject to catastrophic breakdown [216]. External intervention, e.g., government bailout, can prevent large-scale breakdown. Our result indicates that the intervention should be so as to make the time tolerance as uniform as possible to keep any agent “alive” after it becomes broke,

Fig. 39. (Color online.) Time tolerance $T(k)$ (left) and its derivative $dT(k)/d\beta$ (right) versus control parameter β for $T_0 = 20$.

regardless of the size of the agent. Particularly, given a large and a small bank, both facing possible bankruptcy, the criterion to determine the amount of governmental bailout is that they can survive for approximately the same amount of time before turning themselves into profitable agents.

We now present a heuristic argument to understand the mechanism of cooperation optimization. It is useful to rank the nodes into three classes according to their degree values. From Eq. (7.1) and the degree distribution of BA network $P(k) = 2m^2k^{-3}$, we obtain a power-law distribution of the time tolerance T as

$$P(T) = 2m^2 \left[\left(\frac{\sum_l k_l^\beta}{NT_0} \right)^{\frac{-3}{\beta}} \right] T^{\frac{-3}{\beta}}. \quad (7.2)$$

From the expression of the degree distribution $P(k)$, we obtain

$$\bar{k}^\beta = \int_{k_S}^{k_L} 2m^2 k^{-3} k^\beta dk = \frac{2m^2}{(\beta - 2)} (k_L^{\beta-2} - m^{\beta-2}), \quad (7.3)$$

where k_L and $k_S = m$ denote the largest and smallest degree value in the network, respectively.

From Eqs. (7.1) and (7.3), we get the dependence of an agent's time tolerance $T(k)$ on its degree

value k as

$$T(k) = NT_0 \frac{k^\beta}{\sum_l k_l^\beta} = T_0 \frac{k^\beta}{k^\beta} = \frac{T_0 k^\beta}{\frac{2m^2}{(\beta-2)}(k_L^{\beta-2} - m^{\beta-2})},$$

and, hence,

$$\frac{dT(k)}{d\beta} = \frac{2m^2 T_0 k^\beta}{[\frac{2m^2}{(\beta-2)}(k_L^{\beta-2} - m^{\beta-2})]^2} [f(k_L) - f(m)], \quad (7.4)$$

where

$$f(x) \equiv \left(\frac{x^{\beta-2}}{\beta-2}\right) \left(\ln \frac{k}{x} + \frac{1}{\beta-2}\right), (m \leq x \leq k_L). \quad (7.5)$$

We have $df/dx = x^{\beta-3} \ln(k/x)$. For agent with the largest (or the smallest) degree value, i.e., $k = k_L$ (or m), the sign of df/dx is fixed in the open interval (m, k_L) of x , and accordingly $f(x)$ is a monotonous function. For $k = k_L$, we have $k \geq x$, thus $df/dx \geq 0$ holds. The situation $df/dx = 0$ occurs if and only if $x = k_L$, and subsequently $f(x)$ monotonously increases with x . Since $k_L > m$, we have $f(k_L) - f(m) > 0$ and correspondingly $dT(k)/d\beta > 0$. This means that the time tolerance of the networks's largest degree node monotonically increases with β . By the same argument, the time tolerance of the smallest-degree agents decreases monotonically with β . This means that the agents with the largest-degree value, which play a critical role in maintaining cooperation [105], and the agents with the smallest degree values, which account for the majority of the system population, do not play a significant role in optimizing cooperation. Note that, the effects are in fact insignificant, as can be seen from Fig. 38, where ρ_C , ρ_D , and s_d vary non-monotonously with the control parameter β . Since $T(m)$ and $T(k_L)$ exhibit a monotonous behavior (which can be argued analytically), the non-monotonous behavior in ρ_C , ρ_D , and s_d is unlikely to be caused by the behavior of $T(m)$ and $T(k_L)$. In fact, it is caused by the non-monotonous behavior of $T(k)$ for $m < k < k_L$.

The key to cooperation optimization thus lies in the nodes of medium degrees. In particular, for $m < k < k_L$, the sign of $df(x)/dx$ becomes indeterminate and, accordingly, the sign of $dT(k)/d\beta$

cannot be determined. The dependence of T on β for agents with different degrees can be studied numerically. We find that, except for the smallest- and highest-degree agents, T is typically a convex function of β . Since the time tolerance values for the agents with medium degrees peak near $\beta = 0$, as shown in Fig. 39, the corresponding agents live longer than other agents. Since the peak values of T are not much larger than the nominal value T_0 , a slight increase in the medium-degree agents' lifetimes can help stabilize cooperation. The lifetimes of the medium-degree agents are thus critically important for promoting cooperation [22].

7.3. Discussion

In summary, we find that in complex networked systems where agent interactions are governed by evolutionary-game dynamics but agents are subject to death, an effective control scheme can be introduced to optimize cooperation, which has the advantage to stabilize the system by preventing large-scale cascading failures. The key to our control scheme is a time tolerance that prevents any agent from being removed from the system immediately when its payoff falls below a threshold. Our computation and heuristic analysis indicate that, despite the network's being highly heterogeneous, making the time tolerance as uniformly as possible across the network can lead to the emergence of a stable cooperation cluster that has recruited majority of the agents in the network. Simultaneously, substantial death of agents can be avoided. This finding may have implications to policy making to prevent, for example, large-scale breakdown of social and economical systems. The emergence and evolution of cooperation in complex systems have been recognized as a fundamental issue in natural, social, and economical sciences, and our work provides insights into the control of complex dynamical systems in terms of critical issues such as stability, performance, and sustainability.

8 . SPATIOTEMPORAL PATTERNS AND PREDICTABILITY OF CYBERATTACKS

8.1. Introduction

Highly networked communication and information infrastructures built via various state-of-the-art technologies play crucial roles in modern economic, social, military, and political activities. However, such sophisticated infrastructures are facing more and more severe security challenges on the global scale [218–222]. Earlier theoretical works focused on understanding the complex topologies of the Internet [223] and on the likelihood of large scale failures caused by node removal in complex networks [224–228]. Recent years have witnessed tremendous efforts devoted to mitigating and coping with increasing cybersecurity threats. For example, attack graphs were invented to analyze the overall network vulnerability and to generate a global view of network security against attacks [229–232]. By deploying network sensors at particular points in the Internet, monitoring systems were built to detect cyberthreats and statistically analyze the time, sources, and the types of attacks [233], and various visualization methods were developed to better understand the result of the detection and analysis [234–237]. Quite recently, a genetic epidemiology approach to cybersecurity was proposed to understand the factors that determine the likelihood that individual computers are compromised [238], and the general concept of cybersecurity dynamics was introduced [239].

Attack traffic analysis were mainly done in the field of Intrusion Detection System (IDS), the cyberspace's equivalent to the burglar alarm. IDS has become one of the fundamental technologies for network security [240]. There are three approaches to building an IDS: (1) signature or misuse detection, (2) anomaly detection, and (3) hybrid or compound detection. In particular, signature detection technique is based on a predefined set of known attack signatures obtained from security experts. The system observes the activities of subjects and alarms if their behaviors match the malicious ones in the attack signature set. Both host-based [241, 242] and network-

based [243,244] detection systems were developed. Anomaly detection technique is based on machine learning methodologies, such as system call based sequence analysis [245–247], Bayesian networks [248–250], principal component analysis [251–253], and Markov models [254,255]. The IDS monitoring capability can be improved by taking a hybrid approach that combines both signature and anomaly detection strategies [256,257]. All these methods are often based on data packet payload inspection and thus are difficult to perform for high speed networks. Another limitation of these approaches is the assumption that either the attacks are well defined (i.e., signatures) or the normal behaviors are well defined (so are the abnormal behaviors). Recently, there has been a growing interest in flow-based intrusion detection technologies, by which communication patterns within the network are analyzed, instead of the contents of individual packets [258,259]. Interestingly, a quite recent study analyzing the data obtained from the host IDSs reveals strong associations between the network services running on the host and the specific types of threats to which it is susceptible [238]. Making use of the plan recognition method in artificial intelligence, one can predict the attack plan from the IDS alert information [260]. Utilizing virtual or physical networks to test these IDS techniques can be costly and time consuming, hence, as an alternative, simulation modeling approaches were developed to represent computer networks and IDS to efficiently simulate cyberattack scenarios [261–263]. As botnets have become a major threat in cyberspace, cyberattack traffic patterns have also been used to understand botnet’s Command-and-Control strategies [264–266].

In this paper, we uncover the existence of intrinsic spatiotemporal patterns underlying cyberattacks and address the important question of whether certain such attacks may be predicted or anticipated in advance. The overwhelming complexity of the modern cyberspace would suggest complete randomness in the distribution of cyberattacks and, as a result, the intuitive expectation is that attackers’ behaviors are random and attacks are unpredictable. However, our discovery of the

spatiotemporal patterns and quantitative characterization of the predictability of these patterns suggest the otherwise. In particular, distinct from previous works on cyberattack analysis, our efforts concentrate on analyzing the *macroscopic* properties of the attack traffic flows using a data set of cyberattacks available to us. Especially, the data set recorded attacks on 491 consecutive victim IP addresses (sensors) in 18 days. The IP addresses can thus be regarded, approximately, as a variable in space. An attack is regarded as an event occurring in both space and time, and we speak of events in spatiotemporal dimensions. This is much more comprehensive than the analysis of the individual time series obtained from sampled IP addresses or the time series obtained by treating the IP addresses as a whole [267–275]. Our results reveal, for the first time, that robust macroscopic patterns exist in the seemingly random cyberspace: majority of the attacks are governed by a few very limited number of patterns, indicating that cyberattacks are mainly committed by a few types of major attackers, each with unique spatiotemporal characteristics. More specifically, the patterns can be divided into two types: deterministic and stochastic. The emergence of deterministic patterns implies predictability, which can potentially be exploited to anticipate certain types of attacks to achieve greater cybersecurity. We characterize the predictability of attack frequency time series based on information entropy [276]. Our results suggest a surprisingly high degree of predictability, especially for the IPs under deterministic attack. Effective algorithms can then be developed to predict the future attack frequencies. We also develop methods to evaluate the inference probability between the attack frequency time series based on series similarity, which may allow us to plant much fewer attack probes into the Internet while still achieving effective monitoring. The stochastic patterns can be quantified using the flux-fluctuation law in statistical and nonlinear physics [48, 49, 176–180, 182, 183, 196]. Our findings outline a global picture of how cyberattacks are initiated and distributed into the Internet. This will be of potential value to the development of

defense strategies against cyberattacks on a global scale.

8.2. Results

Existence of IP address blocks sharing similar attack patterns and common backgrounds. We define attack frequency $w(t)$ as the number of attacks received by a victim IP per time unit Δt . Figure 40 shows the time series $w(t)$ from all the victim IP addresses for $\Delta t = 1000$ seconds. Surprisingly, instead of overwhelming randomness, we observe substantial regularity: the IP-space can be unequivocally divided into distinct colored blocks, where the amplitudes of the time series within each block are approximately of the same order of magnitude, but the amplitudes from different blocks vary considerably. Note that, within each block, the time series $w(t)$ are approximately synchronized, which correspond to a particular attack pattern. For different attack patterns, there exists a common background in the different IP sectors. For example, about half of the IP addresses (from 1 to 246) belong to the background of aqua block [$w(t) \sim 10^{1.5}$], in which yellow, orange, and red patterns all exist, while all other IP addresses possess a dark-blue background [$w(t) < 10^{0.5}$] with a number of vertical light-blue lines going through them.

To further investigate the properties of each pattern, we concentrate on a small time period, as shown in Fig. 41. Firstly, under much higher time resolution ($\Delta t = 10\text{sec}$), the large number of vertical light-blue lines in Fig. 40 (a) become curves evenly distributed along the t axis and occupying about half of the IP-space (from 1 to 246). These curves are approximately parallel lines within the same IP region but with different slopes in different IP regions (the sloped lines become visually vertical under low time resolution or large observational time period). It is these curves that form the aqua background throughout the upper half of the IP space in Fig. 40. Secondly, all the light-blue curves exist on a dark-blue background with three vertical light-blue lines appearing throughout the entire IP-space, indicating a common background covering every single IP address

Fig. 40. Time Series of Attack Frequency $w(T)$ for All IP Addresses. (a) Spatiotemporal representation of attack frequencies $w(t)$ of all IP addresses on a logarithmic scale, where the x -axis and y -axis are, respectively, time t and IP address index from 1 to 491 (top to bottom). The IP region 1-246 is denoted as the aqua background, where each of the four IP regions (19-31, 35-47, 50-130, and 131-191) exhibits a particular attack pattern that is overlaid on the background. The IP region 247-363 possesses an attack pattern that is overlaid on the dark-blue background lying under the entire IP-space. (b) Three-dimensional presentation of the attack frequency $w(t)$ on a logarithmic scale. The “walls” of unity height are not visible on the logarithmic scale, but higher “walls” are visible in the IP region 364-491 [corresponding to the vertical light-blue lines on the bottom dark-blue background in (a)], which actually occur in the entire IP-space but are mixed with other patterns. The time resolution is $\Delta t = 1000$ seconds.

Fig. 41. Time Series of Attack Frequency with Higher Time Resolution. Time range: $t = 86200$ to 86600 . Time resolution: $\Delta t = 10$ seconds. (a,b) Two- and three-dimensional representations of the attack frequency $w(t)$ for the entire IP-space on a linear scale, respectively.

on which other attack patterns are superimposed. We call the vertical lines “walls,” which represent the level of simultaneous attacks on each victim IP and are visually similar to walls in the three-dimensional representation, as shown in Figs. 40(b) and 41(b).

These observations have the following implications. Firstly, IP addresses in the same colored block are attacked through quite similar patterns, and the block as a whole shows unique spatiotemporal features, which can be effectively associated with the “fingerprint” of a particular set of attackers. For an IP block corresponding to a particular attack pattern, since the relatively strong spatiotemporal regularity of the pattern can only be achieved via attacks highly organized in time and space, it is reasonable to speculate that the corresponding attacks are solely committed by a set of intimately correlated attackers, or even one major attacker, rather than by a set of multiple independent attackers. A single major attacker is more likely to play the role, due to the requirement of close cooperation between the attackers that may be unrealistic. Thus, a particular type of attack patterns, or fingerprints, corresponds to a particular major attacker. Globally, according to

Fig. 42. Deterministic Attack Pattern on Different Time Scales. For time resolution $\Delta t = 10$ seconds and IP region 1-250, (a) deterministic attack pattern in a relatively large time scale: the sloped light-green lines in the IP region 19-31, where the time range is $t = 38000$ to 42000 , (b) enlarged section between the two vertical pink lines in (a) in the time range from $t = 39500$ to 40000 , which provides a time-scale comparison between two types of deterministic attack patterns.

the limited types of attack patterns, a small number of major attackers are responsible for almost all the attacks. Secondly, the consecutiveness of IP addresses attacked in similar patterns reveals the way how the corresponding attackers select their targets, i.e., targeting each IP address within a consecutive IP sector rather than distantly separated IP addresses. Thirdly, the coexistence of multiple patterns overlaying on some identical background indicates that some IP addresses may be under the cross fire from multiple major attackers.

8.2.1. Deterministic Attacks

Counterintuitively, the parallel light-blue lines in Fig. 41(a) constituting the aqua background in the upper half IP-space show a substantial component of deterministic attack behaviors. We speculate that each light-blue line is generated by one of the attacker's devices, such as a zombie

computer in a botnet that launches attacks on a certain IP region with a constant sweeping speed and a certain order with similar time intervals. Such an “organized” attack pattern makes the light-blue lines nearly evenly distributed along the time axis, which can be regarded as *deterministic attacks*. Because of the deterministic rules that the attacks follow, it is possible to predict when and where the next attack is going to take place by identifying the ordered time interval and extrapolating the sweeping speed. We observe that the attacker associated with a light-blue line typically attacks each IP address once approximately within every 3 to 8 seconds. Specifically, the three different sweeping rates in the three sub-regions of IP addresses are about 8 seconds per IP from 51 to 130, 3 seconds per IP from 131 to 191, and 6 seconds per IP from 192 to 246. Similar deterministic attack patterns can also be observed in a relatively narrow IP region close to the top of the IP region (IP 19-31), but in a much larger time scale, indicated as the thicker light-green lines right below the top of Fig. 42(a). Fig. 42(b) is the enlarged graph with the contrast slopes of the two deterministic attack patterns shown more clearly. The deterministic attacks in the IP range 19-31 typically take about 200 to 1100 seconds per IP before switching to the next, with the attack rate of about once per second.

The “wall” attacks in the dark-blue background show another type of deterministic attack patterns [Figs. 40 and 41], which are instantaneous attacks to each IP that can be observed with time resolution of $\Delta t = 1$ second. However, when a higher time resolution is used, e.g., $\Delta t = 10^{-3}$ second, the “wall” attacks are found to occur one after another in the order of the IP index. Fig. 43 shows two typical types of “wall” attacks. For the first type, attacks are performed exactly once on each IP in the order of IP index but with time delays of the order of 10^{-2} second. This is the most frequently observed walls with unity height in the data set, while close occurrence of several “wall” attacks induces a higher wall when larger values of Δt are used. The second kind of “wall” attacks

Fig. 43. Deterministic Attack Pattern of “Walls”. For time resolution $\Delta t = 10^{-3}$ second, (a) one type of “wall” attacks over the entire IP space, with each IP receiving one attack. The time delays for the attacks to reach the targeted IP addresses exhibit three distinct values, separating the dots in this panel into three lines. (b) A different type of “wall” attack with different delay, and (c) enlarged section in the small red square in (a). Five consecutive attacks on each IP occurred before the next IP is attacked.

shown in Figs. 43(b) and 43(c) consists of 5 consecutive attacks on one IP before skipping to the next, strictly in terms of the IP order.

The three different deterministic attack patterns observed in Fig. 42, i.e., the thick green lines in IP 19-31, the background in IP 1-246, the “wall” attacks over the whole IP space, all sweep the IP space strictly in order but with time scales differing by orders of magnitude. The multiscale behaviors associated with the deterministic attack patterns are crucial for understanding and predicting attacks in the cyberspace.

8.2.2. Flux-Fluctuation Relation and Stochastic Attack

Recently there has been a great deal of attention to the flux-fluctuation relations in complex networked systems with stochastic transportation dynamics [48, 49, 176–180, 182, 183, 196], where the flux $w(t)$ of a node is the amount of goods or the number of data packets that it transmits per time

Fig. 44. Flux-Fluctuation Relation. For time resolution $\Delta t = 10$ seconds, $\langle w \rangle$ - σ relations (a) for each IP in the IP space, (b) for IP regions 247–257 and 364–491, and for each IP in the dark-blue background with walls (red circles) and without walls (blue squares), (c) for IP region 258–363 with (red circles) and without (blue squares) walls. The blue dashed lines in (b) and (c) have the slope $1/2$.

unit at time t , and the standard deviation σ of $w(t)$ corresponds to the fluctuation. The following flux-fluctuation relation holds when the traffic system evolves under a given external drive [196]:

$$\sigma = \sqrt{\langle w \rangle + \left(\frac{\sigma_{ext}^2}{\langle W \rangle^2} - \frac{1}{\langle W \rangle} \right) \cdot \langle w \rangle^2}, \quad (8.1)$$

where $\langle w \rangle$ is the time average of flux $w(t)$, $\langle W \rangle$ is the time average of the external drive $W(t)$, and σ_{ext} denotes the standard deviation of $W(t)$. If the external drive W is approximately a constant, i.e., $\sigma_{ext} \approx 0$, or if W follows the Poisson distribution ($\sigma_{ext} = \langle W \rangle$), then $\sigma \sim \sqrt{\langle w \rangle}$. However, if the external drive has large fluctuations, the relation becomes $\sigma \sim \langle w \rangle$.

Our key idea is that the stochastic component of cyberattacks can be characterized as a flux distribution processes among all the IP addresses. Using the flux-fluctuation relation, we can identify and distinguish the patterns of the external drives. Fig. 44(a) shows the flux-fluctuation relation on a double logarithmic scale, where the attack frequency $w(t)$ of a victim IP corresponds to its flux, and the total number of attacks on a certain IP region is regarded as the external drive $M(t)$. We observe that a substantial part of the flux-fluctuation relation follows the scaling $\sigma \sim \sqrt{\langle w \rangle}$ (with slope 1/2 on the logarithmic scale), while a small portion follows the scaling $\sigma \sim \langle w \rangle$ (with slope 1). These results suggest that cyberattacks share some intrinsic common features with stochastic transportation dynamics. The flux-fluctuation theory can then be used to analyze the stochastic components of cyberattack patterns.

From Fig. 44(a), we see that the flux-fluctuation relation for the most heavily bombarded IP region (35-47, see Fig. 40) displays the unity slope, indicating a non-Poisson type of external drive with strong fluctuations. The IP region under mostly deterministic attacks (IP 192-246, the aqua background with no other overlayed attack patterns) corresponds to only one dot in the flux-fluctuation diagram, establishing the deterministic nature of the attack without any randomness. Figure 44(b) shows the σ - $\langle w \rangle$ relation for the dark-blue background with no overlayed attack pat-

Fig. 45. Attack Frequency Deviation Within An IP Region Versus the Average At Each Time Unit for (a) the the dark-blue background and (b) IP region 258-363. The time resolution is $\Delta t = 10$ seconds.

terns, where we obtain the relation $\sigma = \sqrt{\langle w \rangle}$ by removing the deterministic attacks. This is mainly due to the sparsity of attacks associated with the dark-blue background. That is, without walls, mostly only one attack was received in the corresponding IP region within each time unit. We thus have $W = 1$ or 0 . The average flux per unit time for one given IP is $\langle w \rangle \leq 1/N$, with N being the size of the IP region. For observation with T time units, the number of time units with $W = 1$ is denoted by t . We have

$$\sigma_{ext}^2 = \langle W^2 \rangle - \langle W \rangle^2 = \frac{t}{T} - \left(\frac{t}{T} \right)^2 = \langle W \rangle - \langle W \rangle^2. \quad (8.2)$$

Substituting this relation into Eq. (8.1), we obtain

$$\sigma = \sqrt{\langle w \rangle + \left(\frac{\langle W \rangle - \langle W \rangle^2}{\langle W \rangle^2} - \frac{1}{\langle W \rangle} \right) \langle w \rangle^2} = \sqrt{\langle w \rangle - \langle w \rangle^2} \approx \sqrt{\langle w \rangle}. \quad (8.3)$$

When deterministic attacks such as those represented by the walls are included, the fluctuation becomes

$$\sigma' = \sqrt{\frac{(t + t_{\text{wall}})}{T} - \left(\frac{t + t_{\text{wall}}}{T} \right)^2} = \sqrt{\frac{t + t_{\text{wall}}}{T} - \left(\frac{t + t_{\text{wall}}}{T} \right)^2} = \sqrt{\langle w' \rangle - \langle w' \rangle^2}, \quad (8.4)$$

where $\langle w' \rangle = (t + t_{\text{wall}})/T$, and t_{wall} is the number of walls with one attack to each IP. Since $t_{\text{wall}} \gg t_a$, $\langle w' \rangle^2$ cannot be ignored, and so the slope of the σ' - $\langle w' \rangle$ relation on a double logarithmic scale is given by

$$k_s = \frac{\log_{10} \sigma'}{\log_{10} \langle w' \rangle} = \frac{1}{2} \frac{\ln (\langle w' \rangle - \langle w' \rangle^2)}{\log_{10} \langle w' \rangle} \leq \frac{1}{2} \frac{\log_{10} \langle w' \rangle}{\log_{10} \langle w' \rangle} = \frac{1}{2}. \quad (8.5)$$

As shown in Fig. 44(b), walls tend to reduce the slope slightly. This example illustrates the effect of deterministic attacks on the fluctuation of the attack flux. Similar phenomenon is observed in another IP region (IP 258-363) where a less sparse but larger $\langle w \rangle$ scale attack pattern takes place, as shown in Fig. 44(c).

8.2.3. Spatial Concentration of Attacks

Our analysis so far has focused on the *temporal* flux and fluctuation behavior of each IP. It is useful to study the distribution of the attacks in the IP space to distinguish the attack patterns in different IP blocks. The relation between the average attack frequency over the IP addresses in a given region, denoted by $\overline{w}_{\text{IP}}$, and the standard deviation of the attacks distributed among these IP addresses, denoted by σ_{IP} , is shown in Fig. 45, where the left and right panels correspond to the attacks associated with the dark-blue background and with IP address block 258-363, respectively. In the dark-blue background, attacks are sparse. The extreme case with n attacks homogeneously distributed among N IP addresses leads to $\overline{w}_{\text{IP}} = n/N$, where each of n IP addresses receives $w = 1$ attack and the remaining IP addresses have $w = 0$. The standard deviation of the attacks among these IP addresses can simply be written as

$$\sigma_{\text{IP}} = \sqrt{\frac{n}{N} - (\overline{w}_{\text{IP}})^2} = \sqrt{\frac{1}{4} - \left(\overline{w}_{\text{IP}} - \frac{1}{2}\right)^2}, \quad (8.6)$$

which is the equation of an ellipse. This equation matches the lower bound of the real data very well, while any degree of inhomogeneity in the attack distribution will lead to a larger value of σ_{IP} .

Fig. 46. Correlation Coefficients. (a) Correlation coefficient matrix associated with the IP space, (b) total number of attacks on an IP address, M (blue), the average time interval between consecutive attacks on an IP, $\langle \tau \rangle$ (red), and the cluster size to which each IP belongs (green). The clustering threshold is set to be 0.7 (somewhat arbitrary). (c) Time series of attack frequency for IP 155 (red), 160 (blue), and 165 (green). All three IP addresses belong to the region of the aqua background without any other overlaid attack patterns. (d) Time series of attack frequency for IP 405 (red), 409 (blue), and 420 (green), which belong to the region of the dark-blue background without any other overlaid attack patterns. One abrupt peak occurs at IP 409, and another at IP 420. The time resolution is $\Delta t = 10000$ seconds for all four panels.

In addition, for the time resolution of 10 seconds, the effect of “wall” attacks with unity height is to increase each $\overline{w}_{\text{IP}}$ exactly by 1, since a wall introduces one attack to each single IP. However, the change in σ_{IP} due to the “wall” attacks is small. Thus we observe two ellipses from real data in both Figs. 45(a) and 45(b).

The major difference between Figs. 45(a) and 45(b) is the straight line that appears only in 45(b). This is due to the concentration of multiple attacks on a few IP addresses, leaving the remaining IP addresses free of attacks. The extreme case is where all n attacks focus merely on one IP. Then, the average value of $\overline{w}_{\text{IP}}$ is still n/N , but the standard deviation becomes

$$\sigma_{\text{IP}} = \sqrt{\frac{n^2}{N} - (\overline{w}_{\text{IP}})^2} = \sqrt{N-1} \cdot \overline{w}_{\text{IP}}, \quad (8.7)$$

which gives the upper bound fit (the straight line) with the real data, as shown in Fig. 45(b). Consequently, it is the concentration of attacks that distinguishes the attack pattern in the IP region (258-363) from the dark-blue background. We thus see that, for observational time resolution $\Delta t < 10$ seconds in which the attacks are sparse, the plots of $\sigma_{\text{IP}}-\overline{w}_{\text{IP}}$ relation are confined in the region bounded by Eqs. (8.6) and (8.7), and larger values of σ_{IP} imply that the attacks are inhomogeneous in the IP space with uneven spatial concentration.

We conclude that the target selection scheme in the IP region 259-491 is highly stochastic due to the lack of regular spatiotemporal pattern. Within each time unit, there are mainly two target selection schemes: (1) all attacks are concentrated only on a single IP; (2) attacks are evenly spread over the whole region, i.e., each attack packet is received by a different IP.

8.2.4. Inference Probability of Attack Patterns

Due to the surprisingly stable similarity of the attack patterns within an IP block, attack time sequences of frequencies on all IP addresses may be inferred from any single sequence. High inference probability of a consecutive IP region indicates that information obtained from one sensor

Fig. 47. Markov State Transition Probability Matrix (MSTPMs). For time resolution $\Delta t = 10$ seconds, MSTPMs obtained from two randomly selected IP addresses within the corresponding IP region. See the legend of Fig. 45(a) for the eight IP regions under different attack patterns: IP addresses (a1,a2) 10 and 33, (b1,b2) 20 and 30, (c1,c2) 40 and 45, (d1,d2) 60 and 120, (e1,e2) 140 and 190, (f1,f2) 200 and 240, (g1,g2) 260 and 360, and (h1,h2) 420 and 490.

(one IP) may be sufficient to capture the key features of the attack patterns in the whole region. Thus, our discovery of the attack pattern similarity may help to reduce dramatically the number of sensors needed to monitor the attack behavior in the whole cyberspace of interest.

A straightforward measure of inference probability is the correlation coefficient, as the time series belonging to a similar attack pattern are likely to be highly correlated. The correlation coefficient between two time series i and j is given by

$$\rho_{i,j} = \frac{\langle (w_i - \langle w_i \rangle)(w_j - \langle w_j \rangle) \rangle}{\sigma_i \sigma_j}, \quad (8.8)$$

where w_i and w_j are the attack frequencies of IP addresses i and j , and σ_i and σ_j are the corresponding standard deviations. Fig. 46(a) shows the correlation coefficient matrix containing $\rho_{i,j}$ for each i - j pair with relatively low time resolution ($\Delta t = 10^4$ seconds). We see that relatively large correlation coefficients appear within the IP regions corresponding to the readily distinguishable colored blocks in Fig. 40. To further exploit the use of the correlation coefficients, we consider the IP group generated from one given IP i by calculating the correlation coefficients between the time series from this IP and those from all other IP addresses, ρ_{ij} , over a certain threshold ρ_c . The sizes of the groups generated from the IP addresses under a similar attack pattern would be close to each other. From the group size of each IP shown in Fig. 46(b), we see that time series associated with IP addresses in different blocks behave differently, despite the large fluctuations. The correlation coefficient, however, may not be a reliable measure to characterize similarity and inference probability of the main attack pattern. Figure 46(c) shows a case where the correlation coefficient works well, but an unsuccessful case is shown in Fig. 46(d) where the two abrupt peaks produce very low correlation coefficients among the three time series, in spite of their similarities in regions excluding the peaks. We note that the peaks correspond to extreme events with highly concentrated attacks. When the extreme events on different IP addresses are out of phase, the correlation coefficient fails

to reflect the similarity of the main attack pattern and can lead to large fluctuations. Other statistical properties, such as M , the total number of attacks on a particular IP, and $\langle \tau \rangle$, the average time interval between consecutive attacks, can also be used to characterize the attack pattern similarity, as shown in Fig. 46(b). We see that the fluctuations of these quantities are much smaller than ρ_{ij} and they are thus able to better distinguish the IP addresses under different attack patterns. Another disadvantage of the correlation coefficients is that they are sensitive to time resolution Δt . For high resolution (small Δt), the reduction in the attack frequency values may make a time series so sparse that it is dominated by random fluctuations, and this can result in a sharp decrease in the correlation coefficients between such sparse series. For example, for $\Delta t = 10$ seconds, $\rho_{i,j}$ is close to 0 for almost any i - j pairs.

To better characterize the attack patterns and overcome the sensitivity on time resolution and frequency value fluctuations, we coarse-grain the time series according to the order of magnitude of their amplitude, i.e., we replace each $w(t)$ by a state number, $X(t) = \text{int}[\log_{10} w(t)]$, where int denotes the integer closest to $\log_{10} w(t)$ [for $w(t) = 0$, we set $w(t) = 0.1$ so that $X(t) = -1$]. Based on $X(t)$, we can construct a Markov state transition probability matrix (MSTPM) to capture the key information about the attack pattern on any particular IP. The entry in the matrix at the m th row and n th column denotes the probability that the IP is in state m at $t - \Delta t$ and it transitions to state n at time t . Since the length of the time series is relatively large, the transition probabilities are robust against random fluctuations. In addition, for IP addresses under similar attack patterns, the similarity in their state transition patterns would hold irrespective of the time resolution. As shown in Fig. 47, IP addresses within the same attack pattern region have similar MSTPMs, even for high resolution (e.g., $\Delta t = 10$ second). This suggests that MSTPMs, which can be measured via the correlation coefficients where the matrix entries are organized into a vector according to a

Fig. 48. Correlation Coefficient Matrix Associated with the MSTPMs for (a) $\Delta t = 10$ seconds and (b) $\Delta t = 100$ seconds.

certain order, can be used to quantify the inference probability, as shown in Fig. 48. We observe that the IP addresses attacked under similar patterns have relatively high correlation coefficients, while low correlation coefficients can distinguish the deterministic from stochastic attacks. In general, the attack frequencies of a group of IP addresses under deterministic (or stochastic) attack can be inferred from one of them under attack of the same type.

Generally, attack frequency time series have high inference probability, which can be measured by the correlation coefficients or the MSTPMs. The latter, however, are more reliable especially for high time resolutions. This finding would enable one to monitor the cyberattack patterns throughout the entire IP space of interest using fewer sensors.

8.2.5. Predictability of Cyberattacks

Predicting future cyberattacks is an ultimate goal in investigating cyberattack patterns through data analysis. The degree of predictability can be characterized by the uncertainty associated with the state transitions in the coarse-grained time series of attack frequencies, which can be further quantified by the information entropy. Taking into account the temporal correlations in the state

transition process, we define the information entropy of IP i as

$$E_i = - \sum_{S'_i \subset S_i} P(S'_i) \log_2[P(S'_i)], \quad (8.9)$$

where $S_i = \{X_1, X_2, \dots, X_T\}$ denotes the sequence of states that the time series reached for IP i , and $P(S'_i)$ is the probability that the state sequence S'_i appears to IP i . The information entropy E_i takes into account both the heterogeneous probability distribution in different states and the temporal correlations among the states. The entropy is thus able to provide a realistic characterization of the attack patterns.

The predictability Π of a state sequence is defined as the success rate that an algorithm can achieve in predicting the sequence's future states [276]. For a sequence with N_S possible states, the predictability measure is subject to the Fano's inequality: $\Pi \leq \Pi^{\max}(E, N_S)$, where the predictability upper bound $\Pi^{\max}(E, N_S)$ is obtained by solving the following equation

$$E = -\Pi^{\max} \log_2(\Pi^{\max}) - (1 - \Pi^{\max}) \log_2(1 - \Pi^{\max}) + (1 - \Pi^{\max}) \log_2(N_S - 1). \quad (8.10)$$

In the predictability calculation, high time resolutions, e.g., $\Delta t = 10$ seconds, can make the series so sparse that the overwhelming majority of the states are reduced to the “ground state” $X(t) = -1$ [corresponding to $w(t) = 0$]. In this case, the predictability $\Pi^{\max}$ assumes artificially high values, as the future states can trivially be predicted to be $X(t) = -1$, leading to a high success rate. To avoid this artifact, we need to use moderate time resolutions, e.g., $\Delta t = 100$ seconds or 1000 seconds. The length of the time series can affect the value of the predictability. Predictable patterns may not be fully contained within short time series and, hence, we evenly divide the whole time series into multiple sections, each of length of h hours. Figures 49(a) and 49(b) show the predictability upper bound $\Pi^{\max}$ for different section lengths and time resolutions. While longer time series (higher h) tend to be more predictable as expected, all time series have surprisingly high

Fig. 49. Predictability Calculated From Coarse-Grained Time Series. (a,b) Predictability upper bound $\Pi^{\max}$ for all 491 IP addresses for various section lengths under time resolutions $\Delta t = 100$ seconds and 1000 seconds, respectively, where $\Pi^{\max}$ values are averaged over all sections. (c) Average $\Pi^{\max}$ versus section length h for $\Delta t = 100$ seconds (red circles) and 1000 seconds (blue squares). (d) Average $\Pi^{\max}$ of the deterministic (open symbols) and stochastic (closed) attacks versus the section length h for $\Delta t = 100$ seconds (red circles) and 1000 seconds (blue squares). In the figure, “D” denotes deterministic and “R” stands for random or stochastic. See text for explanation on why, in the case of $\Delta t = 100$, stochastic attacks can have a higher predictability upper bound than deterministic attacks.

predictability. There are cases where the predictability probability is over 90%, indicating that it is possible to make correct predictions for over 90% of the cases. We observe that the IP regions of different attack patterns in Fig. 40 have different $\Pi^{\max}$ values, and the IP addresses under each attack pattern are approximately equally predictable. Deterministic attacks have higher predictability than stochastic attacks for $\Delta t = 1000$ seconds, but the opposite occurs for $\Delta t = 100$ seconds, due to the fact that stochastic attacks are much more sparse and, as such, a higher time resolution can introduce overwhelmingly more ground states into the state sequence. Despite the different attack patterns, the average predictability over all IP addresses converges to about 93% with relatively small error bars as the section length is increased, as shown in Fig. 49(c). Figure 49(d) shows the average predictability of the deterministic and stochastic attacks. We see that deterministic attacks systematically have higher predictability than stochastic attacks for $\Delta = 1000$ seconds. Again, sparsity contributes to the relatively high predictability (over 85% for long sections) of stochastic attacks.

8.3. Discussion

8.3.1. How Do Attackers Choose Their Targets?

From Fig. 46(b), we see that the IP addresses under a similar attack pattern have similar M values, which means that different IP addresses in the IP region dominated by one particular attack pattern are approximately equally visited by the attackers. Due to this, we speculate that in most cases, an attacker does not target some specific IP addresses, but a consecutive IP region. In the case of deterministic attacks, the attacker scans a whole block of adjacent IP addresses by using its botnet, and each IP in the targeted block has equal probability to be hit. In the case of stochastic attacks, at each time an attack is launched, IP addresses in the targeted block have similar chances to receive it. It may indicate that the attackers are most likely to view the targeted IP block as a

black-box, the internal structure of which may be irrelevant.

8.3.2. Interplay Between Different Attack Patterns

Except for the parts in Fig. 44 that show slope 1 or $1/2$, there are some parts of the data that do not exhibit a plausible scaling, which correspond to the high attack density patterns that are overlaid on the deterministic aqua background. Figs. 41(a) and 42(b) show that the light-blue lines change their slopes when other attack pattern are superimposed on them. This reveals an interesting phenomenon: different attack patterns may interfere with each other. If so, it would be of great importance to investigate how a certain type of attack pattern promotes or suppresses the efficiency of another type, and how the attack packets differ from normal data packets when transporting through the Internet. Making use of the suppression effect can lead to a dramatic reduction in the transportation efficiency of the attack packets. Our work indicates that it is potentially possible to mitigate or eliminate cybersecurity threats substantially through a macroscopic approach, a field that requires much further efforts.

8.3.3. Attack Pattern Inference and Prediction

The surprising finding that cyberattack patterns are highly predictable encourages us to develop inference and prediction algorithms. The former can provide us with global insights into cybersecurity based on limited information resources, and the latter would enable us not just to get a better understanding of current cyberattack data but also help us to forecast future cyberthreats.

8.3.4. Possible Improvement for Future IDS

The analysis on the deterministic attack reveals the possibility for an IDS to be prepared in advance to the coming attack based on the estimation of the relatively constant attack frequency. If we could make the IDSes in a consecutive IP region communicate on attack information, then the sweeping speed can be estimated, which more accurately tells each IDS when shall the next attack

take place, since the adjacent victim IPs are swept by the attacker one after another in order within a constant time interval.

8.3.5. About the IP Information of the Attackers

With the IP information of the attackers, if possible, we would be able to further verify our speculation that each sweeping is generated by a simple IP address. The spatial distribution of the attackers' IPs can also be retrieved. For deterministic attack, the sweepings are targeted on consecutive IPs, and it is very likely that a substantial part of the attackers' IPs are also consecutive or nearly consecutive, since those attacker could be ex-victims of the same type of attack, which have been successfully compromised by a major attacker and are now serving as a part of its botnet. More properties of cyberattack which can not be observed on the victims' side could also be revealed via such information.

8.4. Materials and Methods

The data set we analyzed was collected between 2/9/2011 and 2/25/2011 by a cyber instrument known as honeypot (see <http://www.honeynet.org>). The data set contains 491 honeypot IP addresses. The honeypot simulates vulnerable computer services corresponding to distinct TCP ports. Since there are no legitimate services associated with these IP addresses, the traffic arriving at these ports is widely deemed as attacks. The raw network traffic arriving at these ports was initially recorded as `pcap` files. By using some standard pre-processing procedure, we reformulated the raw traffic into *flows*, which are the commonly accepted representation of cyber attacks. The pre-processing procedure uses two widely used parameters: the *flow timeout time* of 60 seconds, meaning that a flow expires after 60 seconds of no more packets arriving activities; the *flow lifetime* of 300 seconds, meaning that a flow expires after 300 seconds. The supplementary material contains the data set, which has two columns. Each row is a tuple (a, b), which represents that IP

address a was attacked at time b at some TCP port. Since our analysis treats an IP address as a whole, we do not distinguish the specific ports. For the purpose of protecting privacy, the honeypot IP addresses (i.e., the first column) were anonymized via a one-to-one mapping that also maintains their consecutiveness.

REFERENCES

- [1] Y.-Y. Liu, J.-J. Slotine, A.-L. Barabási, Controllability of complex networks. *Nature (London)* **473**, 167-173 (2011).
- [2] W.-X. Wang, X. Ni, Y.-C. Lai, C. Grebogi, Optimizing controllability of complex networks by minimum structural perturbations. *Phys. Rev. E* **85**, 026115 (2012).
- [3] S. Grün, M. Diesmann, A. Aertsen, Unitary events in multiple single-neuron activity: I. detection and significance. *Neural Comput.* **14**, 43 (2002).
- [4] S. Grün, A. Aertsen, S. Rotter, Statistical significance of coincident spikes: Count-based versus rate-based statistics. *Neural Comput.* **14**, 121 (2002).
- [5] G. Pipa, S. Grün, Nonparametric significance estimation of joint-spike events by shuffling and resampling. *Neurocomputing* **52-54**, 31 (2003).
- [6] T. S. Gardner, D. di Bernardo, D. Lorenz, J. J. Collins, Inferring genetic networks and identifying compound mode of action via expression profiling. *Science* **301**, 102 (2003).
- [7] M. Bansal, V. Belcastro, A. Ambesi-Impiombato, D. di Bernardo, How to infer gene networks from expression profiles. *Mol. Syst. Biol.* **3**, 78 (2007).
- [8] F. Geier, J. Timmer, C. Fleck, Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge. *BMC Syst. Biol.* **1**, 11 (2007).
- [9] K. Supekar, V. Menon, D. Rubin, M. Musen, M. D. Greicius, Network analysis of intrinsic functional brain connectivity in alzheimers disease. *PLoS Comput. Biol.* **4**, e1000100 (2008).
- [10] M. Hecker, S. Lambeck, S. Toepferb, E. van Someren, R. Guthke, Gene regulatory network inference: Data integration in dynamic modelsa review. *BioSystems* **96**, 86 (2009).
- [11] T. Tanaka, Mean-field theory of boltzmann machine learning. *Phys. Rev. E* **58**, 2302 (1998).
- [12] S. Cocco, R. Monasson, Adaptive cluster expansion for inferring boltzmann machines with noisy data. *Phys. Rev. Lett.* **106**, 090601 (2011).
- [13] E. Aurell, M. Ekeberg, Inverse ising inference using all the data. *Phys. Rev. Lett.* **108**, 090201 (2012).
- [14] H. C. Nguyen, J. Berg, Mean-field theory for the inverse ising problem at low temperatures. *Phys. Rev. Lett.* **109**, 050602 (2012).

- [15] F. Ricci-Tersenghi, The bethe approximation for solving the inverse ising problem: a comparison with other inference methods. *J. Stat. Mech.* **2012**, 08015 (2012).
- [16] Y. Roudi, J. Hertz, Mean field theory for nonequilibrium network reconstruction. *Phys. Rev. Lett.* **106**, 048702 (2011).
- [17] M. Méard, J. Sakellariou, Exact mean-field inference in asymmetric kinetic ising systems. *J. Stat. Mech.* **2011**, 08015 (2011).
- [18] J. Sohl-Dickstein, P. B. Battaglino, M. R. DeWeese, New method for parameter estimation in probabilistic models: Minimum probability flow. *Phys. Rev. Lett.* **107**, 220601 (2011).
- [19] P. Zhang, Inference of kinetic ising model on sparse graphs. *J. Stat. Phys.* **148**, 502-512 (2012).
- [20] H.-L. Zeng, M. Alava, E. Aurell, J. Hertz, Y. Roudi, Maximum likelihood reconstruction for ising models with asynchronous updates. *Phys. Rev. Lett.* **110**, 210601 (2013).
- [21] L. Barnett, J. T. Lizier, M. Harré, A. K. Seth, T. Bossomaier, Information flow in a kinetic ising model peaks in the disordered phase. *Phys. Rev. Lett.* **111**, 177203 (2013).
- [22] A. Decelle, F. Ricci-Tersenghi, Pseudolikelihood decimation algorithm improving the inference of the interaction network in a general class of ising models. *Phys. Rev. Lett.* **112**, 070603 (2014).
- [23] M. Timme, Revealing network connectivity from response dynamics. *Phys. Rev. Lett.* **98**, 224101 (2007).
- [24] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. U.S.A.* **104**, 9943 (2007).
- [25] S. G. Shandilya, M. Timme, Inferring network topology from complex dynamics. *New J. Phys.* **13**, 013004 (2011).
- [26] D. Napoletani, T. D. Sauer, Reconstructing the topology of sparsely connected dynamical networks. *Phys. Rev. E* **77**, 026103 (2008).
- [27] A. Clauset, C. Moore, M. E. J. Newman, Hierarchical structure and the prediction of missing links in networks. *Nature (London)* **453**, 98 (2008).
- [28] Z. Levnajić, A. Pikovsky, Network reconstruction from random phase resetting. *Phys. Rev. Lett.* **107**, 034101 (2011).

- [29] W.-X. Wang, Q.-F. Chen, L. Huang, Y.-C. Lai, M. A. F. Harrison, Scaling of noisy fluctuations in complex networks and applications to network prediction. *Phys. Rev. E* **80**, 016116 (2009).
- [30] J. Ren, W.-X. Wang, B. Li, Y.-C. Lai, Noise bridges dynamical correlation and topology in coupled oscillator networks. *Phys. Rev. Lett.* **104**, 058701 (2010).
- [31] W.-X. Wang, Y.-C. Lai, C. Grebogi, J. Ye, Network reconstruction based on evolutionary-game data via compressive sensing. *Phys. Rev. X* **1**, 021021 (2011).
- [32] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, M. A. F. Harrison, Time-series based prediction of complex oscillator networks via compressive sensing. *Europhys. Lett.* **94**, 48006 (2011).
- [33] W.-X. Wang, Y.-C. Lai, C. Grebogi, J.-P. Ye, Network reconstruction based on evolutionary-game data via compressive sensing. *Phys. Rev. X* **1**, 021021 (2011).
- [34] R.-Q. Su, X. Ni, W.-X. Wang, Y.-C. Lai, Forecasting synchronizability of complex networks from data. *Phys. Rev. E* **85**, 056220 (2012).
- [35] R.-Q. Su, W.-X. Wang, Y.-C. Lai, Detecting hidden nodes in complex networks from time series. *Phys. Rev. E* **85**, 065201R (2012).
- [36] R.-Q. Su, Y.-C. Lai, X. Wang, Y.-H. Do, Uncovering hidden nodes in complex network in the presence of noise. *Sci. Rep.* **4**, 3944 (2014).
- [37] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, Y.-C. Lai, Reconstructing propagation networks with natural diversity and identifying hidden sources. *Nat. Commun.* **5**, 4323 (2014).
- [38] X. Han, Z. Shen, W.-X. Wang, Z. Di, Robust reconstruction of complex networks from sparse data. *Phys. Rev. Lett.* **114**, 028701 (2015).
- [39] C. M. Bishop, *Pattern recognition and machine learning* (springer, 2006).
- [40] S. Russell, P. Norvig, *Artificial Intelligence: A Modern Approach (3rd Edition)* (Prentice Hall, 2009), third edn.
- [41] D. H. Ackley, G. E. Hinton, T. J. Sejnowski, A learning algorithm for boltzmann machines. *Cognitive Science* **9**, 147-169 (2014).
- [42] E. J. Candès, J. K. Romberg, T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Info. Theo.* **52**, 489-509 (2006).

- [43] E. J. Candes, J. K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math.* **59**, 1207-1223 (2006).
- [44] D. L. Donoho, Compressed sensing. *IEEE Trans. Info. Theo.* **52**, 1289-1306 (2006).
- [45] R. G. Baraniuk, Compressive sensing. *IEEE Sig. Proc. Mag.* **24**, 118-121 (2007).
- [46] E. J. Candès, M. B. Wakin, An introduction to compressive sampling. *IEEE Sig. Proc. Mag.* **25**, 21-30 (2008).
- [47] J. Romberg, Imaging via compressive sampling. *IEEE Sig. Proc. Mag.* **25**, 14-20 (2008).
- [48] V. Kishore, M. S. Santhanam, R. E. Amritkar, Extreme events on complex networks. *Phys. Rev. Lett.* **106**, 188701 (2011).
- [49] Y.-Z. Chen, Z.-G. Huang, Y.-C. Lai, Controlling extreme events on complex networks. *Sci. Rep.* **4**, 06121 (2014).
- [50] Y.-Z. Chen, *et al.*, Extreme events in multilayer, interdependent complex networks and control. *Sci. Rep.* **5**, 17277 (2015).
- [51] R. J. Glauber, Time-dependent statistics of the ising model. *J. Math. Phys. (N.Y.)* **4**, 294 (1963).
- [52] K. Gao, W.-X. Wang, B.-H. Wang, Self-questioning games and ping-pong effect in the ba network. *Physica A* **380**, 528-538 (2007).
- [53] D. Challet, Y.-C. Zhang, Emergence of cooperation and organization in an evolutionary game. *Physica A* **246**, 407-418 (1997).
- [54] D. Challet, M. Marsili, Y.-C. Zhang, Minority games: interacting agents in financial markets. *OUP Catalogue* (2013).
- [55] M. Paczuski, K. E. Bassler, A. Corral, Self-organized networks of competing boolean agents. *Phys. Rev. Lett.* **84**, 3185-3188 (2000).
- [56] T. Zhou, B.-H. Wang, P.-L. Zhou, C.-X. Yang, J. Liu, Self-organized boolean game on networks. *Phys. Rev. E* **72**, 046139 (2005).
- [57] Z.-G. Huang, J.-Q. Zhang, J.-Q. Dong, L. Huang, Y.-C. Lai, Emergence of grouping in multi-resource minority game dynamics. *Sci. Rep.* **2**, 703 (2012).

- [58] T. M. Liggett, *Interacting Particle Systems* (Springer, New York, 1985).
- [59] M. J. de Oliveira, Isotropic majority-vote model on a square lattice. *J. Stat. Phys.* **66**, 273 (1992).
- [60] A. Barrat, M. Barthélemy, A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, Cambridge, England, 2008).
- [61] C. Castellano, S. Fortunato, V. Loreto, Statistical physics of social dynamics. *Rev. Mod. Phys.* **81**, 591 (2009).
- [62] F. Vazquez, X. Castelló, M. S. Miguel, Agent based models of language competition: Macroscopic descriptions and order-disorder transitions. *J. Stat. Mech.* **04**, 04007 (2010).
- [63] D. M. Abrams, S. H. Strogatz, Modelling the dynamics of language death. *Nature (London)* **424**, 900 (2003).
- [64] A. Kirman, Ants, rationality, and recruitment. *The Quarterly Journal of Economics* **108**, 137 (1993).
- [65] S. Alfarano, T. Lux, F. Wagner, Estimation of agent-based models: The case of an asymmetric herding model. *Computational Economics* **26**, 19 (2005).
- [66] C. Castellano, R. Pastor-Satorras, Non-mean-field behavior of the contact process on scale-free networks. *Phys. Rev. Lett.* **96**, 038701 (2006).
- [67] E. Volz, L. A. Meyers, Epidemic thresholds in dynamic contact networks. *J. R. Soc. Interface* **6**, 233-241 (2009).
- [68] N. T. J. Bailey, *The Mathematical Theory of Infectious Diseases* (Griffin, London, 1975).
- [69] R. M. Anderson, R. M. May, *Infectious Diseases in Humans* (Oxford University Press, Oxford, England, 1992).
- [70] R. Pastor-Satorras, A. Vespignani, Epidemic spreading in scale-free networks. *Phys. Rev. Lett.* **86**, 3200-3203 (2001).
- [71] R. Pastor-Satorras, A. Vespignani, Epidemic dynamics and endemic states in complex networks. *Phys. Rev. E* **63**, 066117 (2001).
- [72] R. Sugden, *The Economics of Rights Co-operation and Welfare* (Blackwell, Oxford, UK, 1986).

- [73] R. Axelrod, W. D. Hamilton, The evolution of cooperation. *Science* **211**, 4489 (1981).
- [74] G. Szabó, C. Tóke, Evolutionary prisoners dilemma game on a square lattice. *Phys. Rev. E* **58**, 69 (1998).
- [75] Y.-Z. Chen, Z.-G. Huang, S.-J. Wang, Y. Zhang, Y.-H. Wang, Diversity of rationality affects the evolution of cooperation. *Phys. Rev. E* **79**, 055101(R) (2009).
- [76] Y.-Z. Chen, Y.-C. Lai, Optimizing cooperation on complex networks in the presence of failure. *Phys. Rev. E* **86**, 045101(R) (2012).
- [77] S. Kullback, R. A. Leibler, On information and sufficiency. *The Annals of Mathematical Statistics* **22(1)**, 79 (1951).
- [78] J. Burbea, C. R. Rao, On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory* **28(3)**, 489 (1982).
- [79] A. Bhattacharyya, On a measure of divergence between two multinomial populations. *Sankhyā: The Indian Journal of Statistics (1933-1960)* **7(4)**, 401 (1946).
- [80] E. Hellinger, Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik (Crelle's Journal)* **136**, 210 (1909).
- [81] D. Irving, F. Sorrentino, Synchronization of dynamical hypernetworks: Dimensionality reduction through simultaneous block-diagonalization of matrices. *Phys. Rev. E* **86**, 056102 (2012).
- [82] F. Sorrentino, Synchronization of hypernetworks of coupled dynamical systems. *New J. Phys.* **14**, 033035 (2012).
- [83] J. Aguirre, R. Sevilla-Escoboza, R. Gutiérrez, D. Papo, J. Buldú, Synchronization of interconnected networks: the role of connector nodes. *arXiv:1406.0081* (2014).
- [84] V. Louzada, N. Araújo, J. Andrade Jr, H. Herrmann, Breathing synchronization in interconnected networks. *Sci. Rep.* **3**, 3289 (2013).
- [85] R. G. Morris, M. Barthelemy, Transport on coupled spatial networks. *Phys. Rev. Lett.* **109**, 128703 (2012).
- [86] S. Gomez, *et al.*, Diffusion dynamics on multiplex networks. *Phys. Rev. Lett.* **110**, 028701 (2013).

- [87] F. Radicchi, A. Arenas, Abrupt transition in the structural formation of interconnected networks. *Nat. Phys.* **9**, 717–720 (2013).
- [88] A. Sole-Ribalta, *et al.*, Spectral properties of the laplacian of multiplex networks. *Phys. Rev. E* **88**, 032807 (2013).
- [89] W. Wang, *et al.*, Asymmetrically interacting spreading dynamics on complex layered networks. *Sci. Rep.* **4**, 5097 (2014).
- [90] J. Gómez-Gardeñes, C. Gracia-Lázaro, L. M. Floría, Y. Moreno, Evolutionary dynamics on interdependent populations. *Phys. Rev. E* **86**, 056113 (2012).
- [91] L.-L. Jiang, M. Perc, Spreading of cooperative behaviour across interdependent groups. *Sci. Rep.* **3**, 2483 (2013).
- [92] M. Santos, S. Dorogovtsev, J. Mendes, Biased imitation in coupled evolutionary games in interdependent networks. *Sci. Rep.* **4**, 4436 (2014).
- [93] J. Gómez-Gardeñes, I. Reinares, A. Arenas, L. M. Floría, Evolution of cooperation in multiplex networks. *Sci. Rep.* **2**, 620 (2012).
- [94] A. Vespignani, Complex networks: The fragility of interdependency. *Nature* **464**, 984–985 (2010).
- [95] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, S. Havlin, Catastrophic cascade of failures in interdependent networks. *Nature* **464**, 1025–1028 (2010).
- [96] R. Parshani, C. Rozenblat, D. Ietri, C. Ducruet, S. Havlin, Inter-similarity between coupled networks. *Europhys. Lett.* **92**, 68002 (2010).
- [97] J. Gao, S. Buldyrev, S. Havlin, H. Stanley, Robustness of a network formed by n interdependent networks with a one-to-one correspondence of dependent nodes. *Phys. Rev. E* **85**, 066134 (2012).
- [98] G. Baxter, S. Dorogovtsev, A. Goltsev, J. Mendes, Avalanche collapse of interdependent networks. *Phys. Rev. Lett.* **109**, 248701 (2012).
- [99] O. Yağan, V. Gligor, Analysis of complex contagions in random multiplex networks. *Phys. Rev. E* **86**, 036103 (2012).
- [100] J. Y. Kim, K.-T. Goh, Coevolution and correlated multiplexity in multiplex networks. *Phys. Rev. Lett.* **111**, 058702 (2013).

- [101] E. J. Candès, *Proceedings of the international congress of mathematicians* (Madrid, Spain, 2006), vol. 3, pp. 1433–1452.
- [102] E. J. Candès, M. B. Wakin, An introduction to compressive sampling. *Signal Processing Magazine, IEEE* **25**, 21–30 (2008).
- [103] In real-life situations the punishment can be different from Sucker’s payoff, and this sometimes can have a dramatic effect on evolutionary-game dynamics. However, our computations indicated that our problem of optimizing cooperation through the use of time delays does not depend on the choices of parameters such as the temptation, temperature, and payoff-matrix parameters.
- [104] The random rewiring mechanism can affect the network dynamics, especially the occurrence of cascading dynamics. Numerically, we find that random rewiring can make the system more susceptible to large-scale cascading failures. This is so because, without random rewiring, agent death tends to reduce the network connectivity, which can inhibit the spreading of the defective strategy and suppress any possible cascading process. In contrast, random rewiring serves to retain the network connectivity so that the defective strategy can “invade” the cluster of cooperative agents effectively. In a defective environment, an agent’s partners can make it more vulnerable to bankruptcy by maintaining some high degree value without improving agent’s payoff. As a result, cascading failures are more likely. In the presence of random rewiring, death of agents can transform the originally power-law distribution into a Poisson distribution. During the cascading process, the center of the Poisson distribution moves toward some moderate degree value for $\beta < 0$ or toward some high degree value for $\beta \geq 0$.
- [105] F. C. Santos, J. M. Pacheco, Scale-free networks provide a unifying framework for the emergence of cooperation. *Phys. Rev. Lett.* **95**, 098104 (2005).
- [106] M. A. J. V. Duijn, M. Huisman, F. N. Stokman, F. W. Wasseur, E. P. H. Zeggelink, Evolution of sociology freshmen into a friendship network. *J. Math. Sociol.* **27**, 153-191 (2003).
- [107] R. Milo, *et al.*, Superfamilies of evolved and designed networks. *Science* **303**, 1538-1542 (2004).
- [108] R. R. Christian, J. J. Luczkovich, Organizing and understanding a winters seagrass foodweb network through effective trophic levels. *Ecol. Modelling* **117**, 99-124 (1999).
- [109] J. A. Dunne, R. J. Williams, N. D. Martinez, Food-web structure and network theory: the role of connectance and size. *Proc. Natl. Acad. Sci.* **99**, 12917-12922 (2002).
- [110] R. Milo, *et al.*, Network motifs: Simple building blocks of complex networks. *Science* **298**, 824-827 (2002).

- [111] D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks. *Nature* **393**, 440-442 (1998).
- [112] A.-L. Barabási, R. Albert, Emergence of scaling in random networks. *Science* **286**, 509-512 (1999).
- [113] R. Albert, H. Jeong, A.-L. Barabási, Internet: Diameter of the world-wide web. *Nature* **401**, 130-131 (1999).
- [114] A. L. A. N., A. Scala, M. Barthelemy, H. E. Stanley, Classes of small-world networks. *Proc. Natl. Acad. Sci. USA* **97**, 11149-11152 (2000).
- [115] R. Albert, H. Jeong, A.-L. Barabási, Error and attack tolerance of complex networks. *Nature (London)* **406**, 378-382 (2000).
- [116] R. Cohen, K. Erez, D. Ben-Avraham, S. Havlin, Resilience of the internet to random breakdowns. *Phys. Rev. Lett.* **85**, 4626-4628 (2000).
- [117] H. Jeong, S. P. Mason, A.-L. Barabási, Z. N. Oltvai, Lethality and centrality in protein networks. *Nature* **411**, 41-42 (2001).
- [118] R. Pastor-Satorras, A. Vespignani, Epidemic dynamics and endemic states in complex networks. *Phys. Rev. E* **63**, 066117 (2001).
- [119] M. E. J. Newman, D. J. Watts, S. H. Strogatz, Random graph models of social networks. *Proc. Natl. Acad. Sci. USA* **99**, 2566-2572 (2002).
- [120] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks. *Rev. Mod. Phys* **74**, 47-97 (2002).
- [121] M. E. J. Newman, The structure and function of complex networks. *SIAM Rev.* **45**, 167-256 (2003).
- [122] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society. *Nature* **435**, 814-818 (2005).
- [123] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: Structure and dynamics. *Phys. Rep.* **424**, 175-308 (2006).
- [124] G. Caldarelli, Scale-free networks: complex webs in nature and technology. *OUP Catalogue* (2007).

- [125] M. Nagy, Z. Ákos, D. Biro, T. Vicsek, Hierarchical group dynamics in pigeon flocks. *Nature* **464**, 890-893 (2010).
- [126] S. Fortunato, Community detection in graphs. *Phys. Rep.* **486**, 75-174 (2010).
- [127] E. Ott, C. Grebogi, J. A. Yorke, Controlling chaos. *Phys. Rev. Lett.* **64**, 1196-1199 (1990).
- [128] S. Boccaletti, C. Grebogi, Y.-C. Lai, H. Mancini, D. Maza, Control of chaos: theory and applications. *Phys. Rep.* **329**, 103-197 (2000).
- [129] J.-J. E. Slotine, W. Li, *Applied Nonlinear Control*, vol. 199 (Prentice-Hall Englewood Cliffs, NJ, 1991).
- [130] X. F. Wang, G. Chen, Pinning control of scale-free dynamical networks. *Physica A* **310**, 521-531 (2002).
- [131] W. Wang, J.-J. E. Slotine, On partial contraction analysis for coupled nonlinear oscillators. *Biol. Cyber.* **92**, 38-53 (2005).
- [132] F. Sorrentino, M. di Bernardo, F. Garofalo, G. Chen, Controllability of complex networks via pinning. *Phys. Rev. E* **75**, 046103 (2007).
- [133] W. Yu, G. Chen, J. Lü, On pinning synchronization of complex dynamical networks. *Automatica* **45**, 429-435 (2009).
- [134] A. Rahmani, M. Ji, M. Mesbahi, M. Egerstedt, Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM J. Cont. Opt.* **48**, 162-186 (2009).
- [135] M. Egerstedt, S. Martini, M. Cao, K. Camlibel, A. Bicchi, Interacting with networks: How does structure relate to controllability in single-leader, consensus networks? *IEEE control. Sys.* **32**, 66-73 (2012).
- [136] F. P. Kelly, A. K. Maulloo, D. K. H. Tan, Rate control for communication networks: shadow prices, proportional fairness and stability. *J. Oper. Res. Soc.* pp. 237-252 (1998).
- [137] M. Chiang, S. H. Low, A. R. Calderbank, J. C. Doyle, Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE* **95**, 255-312 (2007).
- [138] R. Srikant, *The Mathematics of Internet Congestion Control* (Springer, 2004).
- [139] D. Luenberger, *Introduction to dynamic systems: theory, models, and applications* (Wiley, 1979).

- [140] R. E. Kalman, Mathematical description of linear dynamical systems. *J. Soc. Indus. Appl. Math. Ser. A* **1**, 152-192 (1963).
- [141] C.-T. Lin, Structural controllability. *IEEE Trans. Automat. Contr.* **19**, 201-208 (1974).
- [142] R. W. Shields, J. B. Pearson, Structural controllability of multiinput linear systems. *Rice Univ. ECE Tech. Rep.* (1975).
- [143] K. J. Reinschke, G. Wiedemann, Digraph characterization of structural controllability for linear descriptor systems. *Linear Alg. Its Appl.* **266**, 199-217 (1997).
- [144] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*, vol. 6 (Springer, 1998).
- [145] A. Lombardi, M. Hörnquist, Controllability analysis of networks. *Phys. Rev. E* **75**, 056110 (2007).
- [146] T. Nepusz, T. Vicsek, Controlling edge dynamics in complex networks. *Nat. Phys.* **8**, 568-573 (2012).
- [147] G. Yan, J. Ren, Y.-C. Lai, C.-H. Lai, B. Li, Controlling complex networks: How much energy is needed? *Phys. Rev. Lett.* **108**, 218703 (2012).
- [148] Y.-Y. Liu, J.-J. Slotine, A. L. Barabási, Control centrality and hierarchical structure in complex networks. *PLoS ONE* **7**, e44459 (2012).
- [149] J. Nacher, T. Akustu, Dominating scale-free networks with variable scaling exponent: Heterogeneous networks are not difficult to control. *New. J. Phys.* **14**, 073005-073028 (2012).
- [150] J. E. Hopcroft, R. M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.* **2**, 225-231 (1973).
- [151] H. Zhou, Z.-C. Ou-Yang, Maximum matching on random graphs. *arXiv preprint cond-mat/0309348* (2003).
- [152] L. Zdeborová, M. Mézard, The number of matchings in random graphs. *J. Stat. Mech.* **2006**, P05003 (2006).
- [153] G. Menichetti, L. Dall’sta, G. Bianconi, Network controllability is determined by the density of low in-degree and out-degree nodes. *IEEE Trans. Autom. Contr.* **58**, 1719-1731 (2014).

- [154] S. Wuchty, Controllability in protein interaction networks. *ACM Transactions on Mathematical Software (TOMS)* **111**, 7156-7160 (2014).
- [155] M. L. J. Hautus, *Ned. Akad. Wetenschappen, Proc. Ser. A* (Elsevier, 1969), vol. 72, pp. 443–448.
- [156] Z.-Z. Yuan, C. Zhao, Z.-R. Di, W.-X. Wang, Y.-C. Lai, Exact controllability of complex networks. *Nat. Commun.* **4**, 2447 (2013).
- [157] W. J. Rugh, *Linear Systems Theory* (Prentice-Hall, Inc., 1996).
- [158] C. T. Chen, *Linear System Theory and Design* (Oxford University Press, Inc., 1984), first edn.
- [159] P. Erdős, A. Rényi, On random graphs, i. *Publ. Math.* **6**, 290-297 (1959).
- [160] P. Erdős, A. Rényi, On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **5**, 17-61 (1960).
- [161] J. Sun, A. E. Motter, Controllability transition and nonlocality in network control. *Phys. Rev. Lett.* **110**, 208701 (2013).
- [162] G. Strang, *Linear Algebra and Its Applications* (Academic Press, 1976).
- [163] J. D. Noh, H. Rieger, Random walks on complex networks. *Phys. Rev. Lett.* **92**, 118701 (2004).
- [164] V. Tejedor, O. Bénichou, R. Voituriez, Global mean first-passage times of random walks on complex networks. *Phys. Rev. E* **80**, 065104(R) (2009).
- [165] C. Nicolaides, L. Cueto-Felgueroso, R. Juanes, Anomalous physical transport in complex networks. *Phys. Rev. E* **82**, 055101(R) (2010).
- [166] L. Zhao, Y.-C. Lai, K. Park, , N. Ye, Onset of traffic congestion in complex networks. *Phys. Rev. E* **71**, 026125 (2005).
- [167] W.-X. Wang, B.-H. Wang, B. Hu, G. Yan, Q. Ou, General dynamics of topology and traffic on weighted technological networks. *Phys. Rev. Lett.* **94**, 188702 (2005).
- [168] H.-X. Yang, W.-X. Wang, Y.-B. Xie, Y.-C. Lai, B.-W. Wang, Transportation dynamics on networks of mobile agents. *Phys. Rev. E* **83**, 016102 (2011).

- [169] K.-I. Goh, B. Kahng, D. Kim, Universal behavior of load distribution in scale-free networks. *Phys. Rev. Lett.* **87**, 278701 (2001).
- [170] P. Holme, B. J. Kim, Vertex overload breakdown in evolving networks. *Phys. Rev. E* **65**, 066109 (2002).
- [171] R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, A. Arenas, Optimal network topologies for local search with congestion. *Phys. Rev. Lett.* **89**, 248701 (2002).
- [172] J.-T. Sun, S.-J. Wang, Z.-G. Huang, Y.-H. Wang, Effect of degree correlations on networked traffic dynamics. *Physica A* **388**, 3244-3248 (2009).
- [173] J.-Q. Dong, *et al.*, Enhancing transport efficiency by hybrid routing strategy. *EPL* **99**, 20007 (2012).
- [174] P. Echenique, J. G. omez Gardeñes, Y. Moreno, Improved routing strategies for internet traffic delivery. *Phys. Rev. E* **70**, 056105 (2004).
- [175] P. Echenique, J. G. omez Gardeñes, Y. Moreno, Dynamics of jamming transitions in complex networks. *EPL* **71**, 325-331 (2005).
- [176] M. A. deMenezes, A. L. Barabási, Fluctuations in network dynamics. *Phys. Rev. Lett.* **92**, 028701 (2004).
- [177] J. Duch, A. Arenas, Scaling of fluctuations in traffic on complex networks. *Phys. Rev. Lett.* **96**, 218702 (2006).
- [178] S. Meloni, J. G. omez Gardeñes, V. Latora, Y. Moreno, Scaling breakdown in flow fluctuations on complex networks. *Phys. Rev. Lett.* **100**, 208701 (2008).
- [179] Z. Eisler, J. Kertesz, Random walks on complex networks with inhomogeneous impact. *Phys. Rev. E* **71**, 057104 (2005).
- [180] S. Yoon, S.-H. Yook, Y. Kim, Scaling property of flux fluctuations from random walks. *Phys. Rev. E* **76**, 056104 (2007).
- [181] S. He, S. Li, H. R. Ma, Integrating fluctuations into distribution of resources in transportation networks. *Eur. Phys. J. B* **76**, 31-36 (2010).
- [182] B. Kujawski, B. Tadić, G. J. Rodgers, Preferential behaviour and scaling in diffusive dynamics on networks. *New. J. Phys.* **9**, 154 (2007).

- [183] Z. Zhou, *et al.*, Universality of flux-fluctuation law in complex dynamical systems. *Phys. Rev. E* **87**, 012808 (2013).
- [184] V. Kishore, M. S. Santhanam, R. E. Amritkar, Extreme events and event size fluctuations in biased random walks on networks. *Phys. Rev. E* **85**, 056120 (2012).
- [185] V. Kishore, A. R. Sonawane, M. S. Santhanam, Manipulation of extreme events on scale-free networks. *Phys. Rev. E* **88**, 014801 (2013).
- [186] J.-P. Kahane, *Lévy Flights and Related Topics in Physics*, M. F. Shlesinger, G. M. Zaslavsky, U. Frisch, eds. (1995), vol. 450 of *Lecture Notes in Physics*, Berlin Springer Verlag.
- [187] K. A. Penson, K. Górska, Exact and explicit probability densities for one-sided lévy stable distributions. *Phys. Rev. Lett.* **105**, 210604 (2010).
- [188] E. J. Gumbel, *Statistics of Extremes* (Cambridge University, New York, 1958), first edn.
- [189] S. Sabhapandit, S. N. Majumdar, Density of near-extreme events. *Phys. Rev. Lett.* **98**, 140201 (2007).
- [190] J. Skufca, E. Boltt, Communication and synchronization in disconnected networks with dynamic topology: moving neighborhood networks. *Math. Biosci. Eng.* **1**, 347-359 (2004).
- [191] D. J. Stilwell, E. M. Boltt, D. G. Roberson, Sufficient conditions for fast switching synchronization in time varying network topologies. *SIAM J. Dyn. Sys.* **5**, 140-156 (2006).
- [192] B. S. Kim, Y. Do, Y.-C. Lai, Emergence and scaling of synchronization in moving-agent networks with restrictive interactions. *Phys. Rev. E* **88**, 042818 (2013).
- [193] P. Papadimitratos, Z. J. Haas, *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)* (KTH, 2002), vol. 625, pp. 193–204.
- [194] Y. C. Hu, A. Perrig, D. Johnson, *INFOCOM 2003 - Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*. *IEEE Societies* (IEEE, 2003), vol. 3, pp. 1977–1986.
- [195] R. Maheshwari, J. Gao, S. R. Das, *INFOCOM 2007 - 26th IEEE International Conference on Computer Communications* (IEEE, 2007), pp. 107–115.
- [196] Z.-G. Huang, J.-Q. Dong, L. Huang, Y.-C. Lai, Universal flux-fluctuation law in small systems. *Sci. Rep.* **4**, 6787 (2014).

- [197] K. Park, Y.-C. Lai, N. Ye, Characterization of weighted complex networks. *Phys. Rev. E* **70**, 026109 (2004).
- [198] .
- [199] A. M. Colman, *Game theory and its applications: in the social and biological sciences* (Psychology Press, 2013).
- [200] J. M. Smith, *Evolution and the Theory of Games* (Cambridge university press, 1982).
- [201] G. Hardin, The tragedy of the commons. *science* **162**, 1243–1248 (1968).
- [202] M. A. Nowak, R. M. May, Evolutionary games and spatial chaos. *Nature* **359**, 826–829 (1992).
- [203] G. Szabó, C. Hauert, Phase transitions and volunteering in spatial public goods games. *Physical Review Letters* **89**, 118101 (2002).
- [204] C. Hauert, M. Doebeli, Spatial structure often inhibits the evolution of cooperation in the snowdrift game. *Nature* **428**, 643–646 (2004).
- [205] E. Fehr, S. Gächter, Altruistic punishment in humans. *Nature* **415**, 137–140 (2002).
- [206] E. A. Sicardi, H. Fort, M. Vainstein, J. J. Arenzon, Random mobility and spatial structure often enhance cooperation. *Journal of theoretical biology* **256**, 240–246 (2009).
- [207] D. Helbing, W.-J. Yu, The outbreak of cooperation among success-driven individuals under noisy conditions. *Proceedings of the National Academy of Sciences* **106**, 3680–3685 (2009).
- [208] W.-X. Wang, J. Ren, G.-R. Chen, B.-H. Wang, Memory-based snowdrift game on networks. *Physical Review E* **74**, 056113 (2006).
- [209] G. Szabó, J. Vukov, A. Szolnoki, Phase diagrams for an evolutionary prisoner’s dilemma game on two-dimensional lattices. *Physical Review E* **72**, 047107 (2005).
- [210] J. Vukov, G. Szabó, A. Szolnoki, Cooperation in the noisy case: Prisoners dilemma game on two types of regular random graphs. *Physical Review E* **73**, 067103 (2006).
- [211] J. Vukov, G. Szabó, A. Szolnoki, Evolutionary prisoner’s dilemma game on newman-watts networks. *Physical Review E* **77**, 026109 (2008).

- [212] F. C. Santos, M. D. Santos, J. M. Pacheco, Social diversity promotes the emergence of cooperation in public goods games. *Nature* **454**, 213–216 (2008).
- [213] W.-B. Du, X.-B. Cao, M.-B. Hu, W.-X. Wang, Asymmetric cost in snowdrift game on scale-free networks. *Europhysics Letters* **87**, 60004 (2009).
- [214] A. Szolnoki, G. Szabó, Cooperation enhanced by inhomogeneous activity of teaching for evolutionary prisoner’s dilemma games. *Europhysics Letters* **77**, 30004 (2007).
- [215] C. Hauert, M. Holmes, M. Doebeli, Evolutionary games and population dynamics: maintenance of cooperation in public goods games. *Proceedings of the Royal Society of London B: Biological Sciences* **273**, 2565–2571 (2006).
- [216] W.-X. Wang, R. Yang, Y.-C. Lai, Cascade of elimination and emergence of pure cooperation in coevolutionary games on networks. *Physical Review E* **81**, 035102 (2010).
- [217] Z.-G. Huang, Z.-X. Wu, A.-C. Wu, L. Yang, Y.-H. Wang, Role of collective influence in promoting cooperation. *EPL (Europhysics Letters)* **84**, 50008 (2008).
- [218] R. Pastor-Satorras, A. Vespignani, *Evolution and Structure of the Internet: A Statistical Physics Approach* (Cambridge University Press, 2007).
- [219] K.-K. R. Choo, The cyber threat landscape: Challenges and future research directions. *Computers and Security* **30**, 719-731 (2011).
- [220] J. Brenner, *America the Vulnerable: Inside the New Threat Matrix of Digital Espionage, Crime, and Warfare* (Penguin Group US, 2011).
- [221] H. DVLabs, 2011 top cyber security risks report .
- [222] Verizon, 2012 data breach investigations report .
- [223] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the internet topology. *ACM SIGCOMM Comput. Commun. Rev.* **29**, 251-262 (1999).
- [224] R. Pastor-Satorras, A. Vázquez, A. Vespignani, Dynamical and correlation properties of the internet. *Phys. Rev. Lett.* **87**, 258701 (2001).
- [225] R. Cohen, K. Erez, D. ben Avraham, S. Havlin, Resilience of the internet to random breakdowns. *Phys. Rev. Lett.* **85**, 46264629 (2000).

- [226] R. Cohen, K. Erez, D. b Avraham, S. Havlin, Breakdown of the internet under intentional attack. *Phys. Rev. Lett.* **86**, 3682-3685 (2001).
- [227] A. E. Motter, Y.-C. Lai, Cascade-based attacks on complex networks. *Phys. Rev. E* **66**, 065102(R) (2002).
- [228] L. Zhao, K. Park, Y.-C. Lai, Attack vulnerability of scale-free networks due to cascading breakdown. *Phys. Rev. E* **70**, 035101(R) (2004).
- [229] O. Sheyner, J. Haines, S. Jha, R. Lippmann, J. Wing, *Proceedings. 2002 IEEE Symposium on Security and Privacy* (IEEE, 2002), pp. 273–284.
- [230] L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia, An attack graph-based probabilistic security metric. *Data and Applications Security XXII vol. 5094 of Lecture Notes in Computer Science* pp. 283–296 (Springer Berlin Heidelberg, 2008).
- [231] K. Ingols, M. Chu, R. Lippmann, S. Webster, S. Boyer, *Computer Security Applications Conference, 2009. ACSAC 09. Annual* (2009), pp. 117–126.
- [232] S. Jajodia, S. Noel, B. OBerry, Topological analysis of network attack vulnerability. *Managing Cyber Threats of Massive Computing* **5**, 247-266 (2005).
- [233] D. I. D. System.
- [234] H. Koike, K. Ohno, *VizSEC/DMSEC '04 Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004), pp. 143–147.
- [235] K. Lakkaraju, W. Yurcik, A. J. Lee, *VizSEC/DMSEC '04 Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004), pp. 65–72.
- [236] R. Ball, G. A. Fink, C. North, *VizSEC/DMSEC '04 Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004), pp. 55–64.
- [237] H. Koike, K. Ohno, K. Koizumi, *VIZSEC '05 Proceedings of the IEEE Workshops on Visualization for Computer Security* (2005), p. 11.
- [238] S. Gil, A. Kott, A.-L. Barabási, A genetic epidemiology approach to cyber-security. *Scientific Reports* **4**, 5659-5665 (2014).
- [239] S. Xu, *2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)*.

- [240] A. Patcha, J. M. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks* **51**, 34483470 (2007).
- [241] S. E. Smaha, *IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL* (IEEE, 1988), pp. 37–44.
- [242] N. Ye, S. M. Emran, Q. Chen, S. Vilbert, *IEEE Transactions on Computers* (IEEE, 2002 July), vol. 51, pp. 810–820.
- [243] D. Anderson, T. Frivold, A. Tamaru, A. Valdes, *Computer Science Laboratory, SRI International, Menlo Park, CA, USA, Technical Report SRI-CSL-95-0*, (1994 May).
- [244] S. Staniford, J. A. Hoagland, J. M. McAlerney, Practical automated detection of stealthy portscans. *Journal of Computer Security* **10**, 105-136 (2002).
- [245] S. Forrest, S. A. Hofmeyr, A. Somayaji, T. A. Longstaff, *IEEE Symposium on Research in Security and Privacy, Oakland, CA, USA* (IEEE, 1996), pp. 120–128.
- [246] C. Warrender, S. Forrest, B. Pearlmutter, *IEEE Symposium on Security and Privacy, Oakland, CA, USA* (1999), pp. 133–145.
- [247] E. Eskin, S. J. Stolfo, W. Lee, *DARPA Information Survivability Conference and Exposition II, Anaheim, CA* (2001), pp. 165–175.
- [248] A. Valdes, K. Skinner, *Recent Advances in Intrusion Detection Toulouse, France* (2000), pp. 80–92.
- [249] N. Ye, M. Xu, S. M. Emran, *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop, West Point, New York* (2000, Jun), pp. 175–179.
- [250] C. Kruegel, D. Mutz, W. Robertson, F. Valeur, *19th Annual Computer Security Applications Conference, Las Vegas, NV* (IEEE, 2003, Dec 8-12), pp. 14–23.
- [251] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, *IEEE Foundations and New Directions of Data Mining Workshop, Melbourne, Florida, USA* (2003), pp. 172–179.
- [252] W. Wang, X. Guan, X. Zhang, *International Symposium on Neural Networks, Dalian, China* (2004), pp. 657–662.
- [253] W. Wang, R. Battiti, *The First International Conference on Availability, Reliability and Security, Vienna, Austria* (2006), pp. 270–279.

- [254] N. Ye, Y. Zhang, C. M. Borror, Robustness of the markov-chain model for cyber-attack detection. *IEEE Transactions on Reliability* **53**, 116-123 (2004).
- [255] D. Y. Yeung, Y. Ding, Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition* **36**, 229-243 (2003).
- [256] E. Tombini, H. Debar, L. Mé, M. Ducassé, *20th Annual Computer Security Applications Conference, Tucson, AZ, USA* (IEEE, 2004, Dec. 6-10), pp. 428–437.
- [257] J. Zhang, M. Zulkernine, *The First International Conference on Availability, Reliability and Security, Vienna University of Technology* (IEEE, 2006, Apr. 20-22), pp. 262–269.
- [258] A. S. Kim, H. Kang, S. C. Hang, S. H. Chung, J. W. Hang, *Network Operations and Management Symposium* (2004 , Apr. 23-23), pp. 599–612.
- [259] A. Sperotto, *et al.*, An overview of ip flow-based intrusion detection. *Communications Surveys and Tutorials, IEEE* **12**, 343-356 (2010).
- [260] X. Qin, W. Lee, *Proceedings of the 20th Annual Computer Security Applications Conference* (2004, Dec 6-10), pp. 370–379.
- [261] S. Cheung, U. Lindqvist, M. W. Fong, *Proceedings of the DARPA Information Survivability Conference and Exposition* (IEEE, 2003, Apr. 22-24), pp. 284–292.
- [262] D. M. Nicol, G. H. Yan, J. Liu, M. Jijensam, *Proceedings of the 2003 Winter Simulation Conference* (IEEE, 2003, Dec. 7-10), pp. 650–657.
- [263] M. Kuhl, J. Kistner, K. Costantini, M. Sudit, *Proceedings of the 2007 Winter Simulation Conference* (IEEE, 2007, Dec. 9-12), pp. 1180–1188.
- [264] G. Gu, R. Perdisci, J. Zhang, W. Lee, *SS'08 Proceedings of the 17th conference on Security symposium* (2008), pp. 139–154.
- [265] G. Gu, J. Zhang, W. Lee, *Proceedings of 16th Annual Network and Distributed System Security Symposium* (2008, Feb).
- [266] E. J. Kartaltepe, J. A. Morales, S. H. Xu, R. Sandhu, *Proceedings of the 8th international conference on Applied cryptography and network security* (2010, Jun. 22-25), pp. 511–528.
- [267] Z. Zhan, M. Xu, S. Xu, Characterizing honeypot-captured cyber attacks: Statistical framework and case study. *IEEE Transactions on Information Forensics and Security* **8**, 1775-1789 (2013).

- [268] Z. Li, A. Goyal, Y. Chen, V. Paxson, Towards situational awareness of large-scale botnet probing events. *Information Forensics and Security, IEEE Transactions on* **6**, 175-188 (2011).
- [269] S. Almotairi, A. Clark, G. Mohay, J. Zimmermann, *Proc. International Conference on Internet Monitoring and Protection* (2009), pp. 7–13.
- [270] S. I. Almotairi, A. J. Clark, G. M. Mohay, J. Zimmermann, *Proc. IFIP International Conference on Network and Parallel Computing* (2008), pp. 147–154.
- [271] A. Clark, M. Dacier, G. Mohay, F. Pouget, J. Zimmermann, Internet attack knowledge discovery via clusters and cliques of attack traces. *Journal of Information Assurance and Security* **1**, 21–32 (2006).
- [272] S. I. Almotairi, *et al.*, *5th Australian Digital Forensics Conference* (2007), pp. 79–87.
- [273] G. Conti, K. Abdullah, *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (2004), pp. 45–54.
- [274] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, L. Peterson, *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC'04)*, IMC '04 (ACM, New York, NY, USA, 2004), pp. 27–40.
- [275] E. Wustrow, M. Karir, M. Bailey, F. Jahanian, G. Huston, *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC'10)* (2010), pp. 62–74.
- [276] C.-M. Song, Z.-H. Qu, N. Blumm, A. Barabási, Limits of predictability in human mobility. *Science* **327**, 1018 (2010).
- [277] V. Sood, S. Redner, Voter model on heterogeneous graphs. *Computational Economics* **94**, 178701 (2005).
- [278] P. J. Antsaklis, A. N. Michel, *Linear Systems* (McGraw-Hill, 1997).
- [279] T. A. Davis, Y. Hu, The university of florida sparse matrix collection. *ACM Trans. Math. Software (TOMS)* **38**, 1 (2011).
- [280] D. Baird, J. Luczkovich, R. R. Christian, Assessment of spatial and temporal variability in ecosystem attributes of the st marks national wildlife refuge, apalachee bay, florida. *Estuarine Coast. Shelf Sci.* **47**, 329-349 (1998).
- [281] J. Memmott, N. D. Martinez, J. E. Cohen, Predators, parasitoids and pathogens: species richness, trophic generality and body sizes in a natural food web. *J. Animal Ecol.* **69**, 1-15 (2000).

- [282] N. D. Martinez, Artifacts or attributes? effects of resolution on the little rock lake food web. *Ecol. Mono.* **61**, 367-392 (1991).
- [283] R. E. Ulanowicz, D. L. DeAngelis, Network analysis of trophic dynamics in south florida ecosystems. *US Geo. Sur. Prog. South Florida Ecosystem* **114** (2005).

A

UNIVERSAL STRUCTURAL ESTIMATOR AND DYNAMICS APPROXIMATOR FOR COMPLEX NETWORKS

SUPPLEMENTARY MATERIALS FOR UNIVERSAL STRUCTURAL ESTIMATOR AND DYNAMICS APPROXIMATOR FOR COMPLEX NETWORKS

Supplementary Note 1: Implementation details of 14 dynamical processes on complex networks

All types of network dynamical processes studied in the main text, except for PDG and SG, are implemented in the following way. At each time step t , the probabilities for node i to have a state $x_i(t+1) = 1$ and 0 at the next time step conditioned on the state configuration of i 's neighbors, i.e., $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$, are calculated according to Table 1 in the main text. If $x_i(t) = 0$ (or 1), then it is switched to state $x_i(t+1) = 1$ (or 0) with the probability $P_i^{0 \rightarrow 1}$ (or $P_i^{1 \rightarrow 0}$) or remains at 0 (or 1) with the probability $1 - P_i^{0 \rightarrow 1}$ (or $1 - P_i^{1 \rightarrow 0}$).

Listed below is a detailed description of the dynamical processes and the parameter settings. (Reasonable changes in parameter values do not affect the reconstruction performance).

1. *SDBM*. The dynamics is described in detail in the main text. The network parameters (link weights and node biases) are uniformly chosen between 0.3 and 0.7 in our calculation.
2. *Ising Glauber*. Opinion dynamics models often adopt the classic Ising model of ferromagnetic spins [1]. The temperature parameter κ characterizes the level of “rationality” of the individuals, which represents the uncertainties in accepting the opinion. The coupling strength parameter J (or the ferromagnetic-interaction parameter) characterizes the intensity of the interaction between the connected nodes. The simulation parameters are $\kappa = 1$ and $J = 0.1$.
3. *Majority vote*. The majority-vote model [2, 3] is a non-equilibrium spin model. Spins tend to align with the neighborhood majority under the influence of noise with parameter Q quantifying the probability of misalignment. In our simulations, we set $Q = 0.3$.

4. *Minority game.* In the minority game model [4–8], each individual chooses the strategy adopted by the minority of its neighbors with a higher probability than that for the majority. The probability for node i to be 1 at the next time step is proportional to the number of its 0-state neighbors, $k_i - n_i$. We then have $P_i^{0 \rightarrow 1} = (k_i - n_i)/k_i$ and $P_i^{1 \rightarrow 0} = n_i/k_i$.
5. *Voter and link-update voter.* In the voter model [2,9], each node i adopts the state of one of its randomly selected neighbors. If i is currently inactive (state 0), while there are n_i neighbors in the active state (state 1), it becomes active with the probability $P_i^{0 \rightarrow 1} = n_i/k_i$. An active node i becomes inactive with the probability $P_i^{1 \rightarrow 0} = (k_i - n_i)/k_i$. The voter model is similar to the minority game model but with the definition of $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$ swapped.
The link-update voter model is a variant of the voter model [10, 11].
6. *Language model.* In the language model [12, 13], the 0 and 1 states stand for an individual's two primary choices of the language. The probability of switching language is proportional to the fraction of the speakers in its neighborhood, raised to the power α and multiplied by the status parameter s or $1 - s$ of the respective language. In our simulations, we use $\alpha = 0.5$ and $s = 0.6$.
7. *Kirman ant colony model.* In the model [14, 15], to choose between stock market trading strategies, nodes move from the non-adopted state (state 0) to the adopted state (state 1) with the probability $P_i^{0 \rightarrow 1} = c_1 + dn_i$, where c_1 represents the individual strategy, which is independent of the influence of its neighbors. The parameter d quantifies the *herding behavior*, whereby a non-adopted-state individual copies the strategy of their n_i adopted-state neighbors. A node i with n_i of its k_i neighbors in the adopted state has $k_i - n_i$ neighbors in the non-adopted state. Due to the herding effect, the probability for the transition from an

adopted state to a non-adopted state is $P_i^{1 \rightarrow 0} = c_2 + d(k_i - n_i)$. In our simulations, we set $c_1 = c_2 = 0.1$ and $d = 10/N$, where N is the network size.

8. *SIS and CP*. In the susceptible-infected-susceptible (SIS) disease-spread model [?, 16, 17], an infected individual transmits disease to node i in its neighborhood with the probability λ_i . If a susceptible node i has n_i infected neighbors, the probability that it will be infected is $P_i^{0 \rightarrow 1}$. The recovery rate $P_i^{1 \rightarrow 0} = \mu_i$ is kept constant for node i . When simulating the SIS model, all infected neighbors of node i affect i . In particular, for each infected neighbor, a random number is drawn and checked if it is smaller than λ_i , and i gets infected if any of the random number is smaller than λ_i . We thus have $P_i^{0 \rightarrow 1} = 1 - (1 - \lambda)^{n_i}$, where the second term gives the chance for i to remain susceptible.

For the contact process (CP) model [18, 19], at a time step, each node i randomly selects one of its k_i neighbors and check whether it is an infected node. If yes, it will transmit the disease to i with the probability λ_i . Since the chance of getting an infected node selected is n_i/k_i , the probability of infecting node i at the current time step is $(n_i/k_i)\lambda_i$.

9. *Evolutionary game models: SG, SQ-SG, PDG, and SQ-PDG*. The implementation of evolutionary game dynamics, PDG [20–23] and SG [24], is slightly more complicated. In the PDG model, the following two processes occur at each time step. (1) *Game playing and payoffs*. Each agent plays the classical prisoner’s dilemma game (PDG) with all its nearest neighbors, and the total payoff is the sum of the payoffs gained in its two-player games with all the connected agents. Each player may choose either to cooperate, C, or to defect, D, in a single encounter. If both players choose C, both will get payoff R . If one defects while the other cooperates, D gets T , while C gets S . If both defect, both get P , where $T > R > P > S$. The PDG parameters in our work are chosen to be $R = 1$, $T = b > 1$, and $S = P = 0$ [20].

(2) *Strategy updating.* At each time step, agent i randomly chooses a neighbor j and imitates j 's strategy with the probability $P_{i \rightarrow j} = \{1 + \exp [-(U_j - U_i)/\kappa]\}^{-1}$, where U_i and U_j are the payoffs for agents i and j , and κ is the level of agents' "rationality" representing the uncertainties in assessing the best strategy. The only difference between PDG and SG is the choice of the game parameter values. In SG, the parameters are chosen to be $R = 1$, $T = 1 + r$, $S = 1 - r$, and $P = 0$, where $0 < r < 1$. The probability that node i chooses to be cooperative or defective at the next time step requires information about its neighbors' states and payoffs at the current time step. This violates the interdependency assumption underlying a Markov network. It is thus difficult to analytically calculate the probabilities $P_i^{0 \rightarrow 1}$ and $P_i^{1 \rightarrow 0}$. We find, however, that the structure of this non-Markov network can be successfully reconstructed by SDBM, a Markov network.

A self-questioning mechanism [25] can be introduced to ensure the interdependency properties of the Markov networks for PDG and SG. At the strategy updating stage, we define the probability that a cooperative node i chooses a defective strategy at the next time step to be $P_i^{0 \rightarrow 1} = \{1 + \exp [-(U_i^C - U_i^D)/\kappa]\}^{-1}$, where U_i^C and U_i^D are the payoffs obtained while adopting the cooperative and defective strategies, respectively. The probability of choosing a cooperative strategy is $P_i^{1 \rightarrow 0} = 1 - P_i^{0 \rightarrow 1}$. For PDG, we have $U_i^C = k_i - n_i$ and $U_i^D = b(k_i - n_i)$. For SG, we have $U_i^C = (k_i - n_i) + (1 - r)n_i$ and $U_i^D = (1 + r)(k_i - n_i)$, where n_i denotes the number of defective neighbors. In our simulations, nodal states 0 and 1 denote cooperation and defection, respectively, and we set $b = 1.2$, $r = 0.7$, and $\kappa = 2$ for all cases of the evolutionary game dynamics.

Supplementary Note 2: Model and real-world networks

TABLE 6

Description of the 9 real-world networks used in our study (N - number of nodes; L - number of edges).

Type	Index	Name	N	L	Description
Trust	1	College Student [4, 5]	32	96	Social network
	2	Prison Inmate [4, 5]	67	182	Social network
Protein	4	Protein-1 [5]	95	213	Protein network
	5	Protein-2 [5]	53	123	Protein network
	6	Protein-3 [5]	99	212	Protein network
Food Web	12	Seagrass [9]	49	226	Food Web
	13	Grassland [10]	88	137	Food Web
	14	Ythan [10]	135	601	Food Web
Circuits	3	s208a [6]	122	189	Logic circuit

Supplementary Table 7 lists the names, types, sizes of the real-world networks used for topology reconstruction in the main text.

Supplementary Note 3: SDBM as a generative model for all types of dynamics

Supplemental Figs. 1-12 show the comparison between the conditional probabilities of the original system and of the approximator using its self-generated state configuration time series for all types of dynamics studied except for the non-Markovian classic PDG and SG, whose conditional probability time series are difficult to compute. We observe that in about half of the cases, namely, SDBM itself, Ising Glauber, SQ-SG, SQ-PDG, the Kirman model, CP, and SIS, the reconstructed SDBM approximators are capable of serving as a generative model of the original time series data, so the reconstructed SDBMs can be used to simulate the long-term behavior of the corresponding original systems.

Fig. 50. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for SDBM Dynamics. The conditional probability $P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ time series generated by the original system from the original state configuration time series (a,c) and by the SDBM approximator from the state configuration time series produced by itself (b,d) for ER random (a,b) and BA scale-free (c,d) topology. (e,f) The mean and standard deviations of $p = P[x_i(t+1) = 1 | \mathbf{X}_i^R(t)]$ for each node i for all time steps for ER random (e) and BA scale-free (f) networks obtained from the original system (red circles) and the approximator (blue squares), where $1 \leq i \leq N$ (the x-axis).

Fig. 51. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Ising Glauber (IS) Dynamics. Legends are identical to that in Fig. 50.

Fig. 52. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Self-Questioning Snowdrift Game (SQ-SG) Dynamics. Legends are identical to that in Fig. 50.

Fig. 53. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Self-Questioning Prisoner's Dilemma Game (SQ-PDG) Dynamics. Legends are identical to that in Fig. 50.

Fig. 54. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Minority Game (MG) Dynamics. Legends are identical to that in Fig. 50.

Fig. 55. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Voter (VT) Dynamics. Legends are identical to that in Fig. 50.

Fig. 56. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Majority Vote (MV) Dynamics. Legends are identical to that in Fig. 50.

Fig. 57. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Link-Update Voter (LV) Dynamics. Legends are identical to that in Fig. 50.

Fig. 58. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Language Model (LM). Legends are identical to that in Fig. 50.

Fig. 59. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Kirman (KM) Dynamics.

Legends are identical to that in Fig. 50.

Fig. 60. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Contact Process (CP) Model.

Legends are identical to that in Fig. 50.

Fig. 61. Comparison Between the Conditional Probabilities of the Original System and the Approximator Using Its Self-Generated State Configuration Time Series for Susceptible-Infected-Susceptible (SIS) Disease-Spread Model. Legends are identical to that in Fig. 50.

REFERENCES

- [1] Glauber, R. J. Time-dependent statistics of the ising model. *J. Math. Phys. (N.Y.)* **4**, 294 (1963).
- [2] Liggett, T. M. *Interacting Particle Systems* (Springer, New York, 1985).
- [3] de Oliveira, M. J. Isotropic majority-vote model on a square lattice. *J. Stat. Phys.* **66**, 273 (1992).
- [4] Challet, D. & Zhang, Y.-C. Emergence of cooperation and organization in an evolutionary game. *Physica A* **246**, 407–418 (1997).
- [5] Challet, D., Marsili, M., Zhang, Y.-C. *et al.* Minority games: interacting agents in financial markets. *OUN Catalogue* (2013).
- [6] Paczuski, M., Bassler, K. E. & Corral, A. Self-organized networks of competing boolean agents. *Phys. Rev. Lett.* **84**, 3185–3188 (2000).
- [7] Zhou, T., Wang, B.-H., Zhou, P.-L., Yang, C.-X. & Liu, J. Self-organized boolean game on networks. *Phys. Rev. E* **72**, 046139 (2005).
- [8] Huang, Z.-G., Zhang, J.-Q., Dong, J.-Q., Huang, L. & Lai, Y.-C. Emergence of grouping in multi-resource minority game dynamics. *Sci. Rep.* **2**, 703 (2012).
- [9] Sood, V. & Redner, S. Voter model on heterogeneous graphs. *Computational Economics* **94**, 178701 (2005).
- [10] Barrat, A., Barthélemy, M., & Vespignani, A. *Dynamical Processes on Complex Networks* (Cambridge University Press, Cambridge, England, 2008).
- [11] Castellano, C., Fortunato, S. & Loreto, V. Statistical physics of social dynamics. *Rev. Mod. Phys.* **81**, 591 (2009).
- [12] Vazquez, F., Castelló, X. & Miguel, M. S. Agent based models of language competition: Macroscopic descriptions and order-disorder transitions. *J. Stat. Mech.* **04**, 04007 (2010).
- [13] Abrams, D. M. & Strogatz, S. H. Modelling the dynamics of language death. *Nature (London)* **424**, 900 (2003).
- [14] Kirman, A. Ants, rationality, and recruitment. *The Quarterly Journal of Economics* **108**, 137 (1993).

- [15] Alfarano, S., Lux, T. & Wagner, F. Estimation of agent-based models: The case of an asymmetric herding model. *Computational Economics* **26**, 19 (2005).
- [16] Bailey, N. T. J. *The Mathematical Theory of Infectious Diseases* (Griffin, London, 1975).
- [17] Antsaklis, P. J. & Michel, A. N. *Linear Systems* (McGraw-Hill, 1997).
- [18] Castellano, C. & Pastor-Satorras, R. Non-mean-field behavior of the contact process on scale-free networks. *Phys. Rev. Lett.* **96**, 038701 (2006).
- [19] Volz, E. & Meyers, L. A. Epidemic thresholds in dynamic contact networks. *J. R. Soc. Interface* **6**, 233–241 (2009).
- [20] Axelrod, R. & Hamilton, W. D. The evolution of cooperation. *Science* **211**, 4489 (1981).
- [21] Szabó, G. & Tóke, C. Evolutionary prisoners dilemma game on a square lattice. *Phys. Rev. E* **58**, 69 (1998).
- [22] Chen, Y.-Z., Huang, Z.-G., Wang, S.-J., Zhang, Y. & Wang, Y.-H. Diversity of rationality affects the evolution of cooperation. *Phys. Rev. E* **79**, 055101(R) (2009).
- [23] Chen, Y.-Z. & Lai, Y.-C. Optimizing cooperation on complex networks in the presence of failure. *Phys. Rev. E* **86**, 045101(R) (2012).
- [24] Sugden, R. *The Economics of Rights Co-operation and Welfare* (Blackwell, Oxford, UK, 1986).
- [25] Gao, K., Wang, W.-X. & Wang, B.-H. Self-questioning games and ping-pong effect in the ba network. *Physica A* **380**, 528–538 (2007).
- [26] Duijn, M. A. J. V., Huisman, M., Stokman, F. N., Wasseur, F. W. & Zeggelink, E. P. H. Evolution of sociology freshmen into a friendship network. *J. Math. Sociol* **27**, 153–191 (2003).
- [27] Milo, R. *et al.* Superfamilies of evolved and designed networks. *Science* **303**, 1538–1542 (2004).
- [28] Christian, R. R. & Luczkovich, J. J. Organizing and understanding a winters seagrass foodweb network through effective trophic levels. *Ecol. Modelling* **117**, 99–124 (1999).
- [29] Dunne, J. A., Williams, R. J. & Martinez, N. D. Food-web structure and network theory: the role of connectance and size. *Proc. Natl. Acad. Sci.* **99**, 12917–12922 (2002).

- [30] Milo, R. *et al.* Network motifs: Simple building blocks of complex networks. *Science* **298**, 824–827 (2002).

B

THE PARADOX OF CONTROLLING COMPLEX NETWORKS: CONTROL INPUTS VERSUS ENERGY REQUIREMENT

THE PARADOX OF CONTROLLING COMPLEX NETWORKS: CONTROL INPUTS VERSUS ENERGY REQUIREMENT

Supplementary Note 1: Analytical calculation of the in- and out-degree distributions

The in- and out-degree distributions of a directed complex network under connection bias probability $\lambda \equiv P_b$ can be obtained analytically.

Defining k_S and k_L to be the numbers of nodes in the neighborhood of a node with degree k , whose degrees are smaller or larger than k , respectively, we have

$$k_S = k \sum_{k'=k_{\min}}^k P(k'|k) = \frac{k}{\langle k \rangle} \sum_{k_{\min}}^k k' P(k) = \frac{k}{\langle k \rangle} \int_{k_{\min}}^k k' \cdot C k'^{-\gamma} dk' = \frac{Ck}{\langle k \rangle} \int_{k_{\min}}^k k'^{1-\gamma} dk', \quad (\text{B.1})$$

and

$$k_L = k \sum_{k'=k}^{k_{\max}} P(k'|k) = \frac{Ck}{\langle k \rangle} \int_k^{k_{\max}} k'^{1-\gamma} dk' \stackrel{\gamma \geq 2}{=} \frac{Ck^{3-\gamma}}{\langle k \rangle(\gamma - 2)} = Ak^{3-\gamma}, \quad (\text{B.2})$$

where

$$A = \frac{C}{\langle k \rangle(\gamma - 2)}. \quad (\text{B.3})$$

Therefore,

$$k_S = k - k_L. \quad (\text{B.4})$$

We then have

$$k_{\text{out}} = \lambda k_S + (1 - \lambda)k_L = \lambda(k - k_L) + (1 - \lambda)k_L = (1 - 2\lambda)k_L + \lambda k = (1 - 2\lambda)Ak^{3-\gamma} + \lambda k, \quad (\text{B.5})$$

and

$$k_{\text{in}} = (1 - \lambda)k_S + \lambda k_L = (1 - \lambda)k + (2\lambda - 1)Ak^{3-\gamma}.$$

The quantities $P(k_{\text{out}})$ and $P(k_{\text{in}})$ can be derived from

$$P(k_{\text{out}}) dk_{\text{out}} = P(k) dk \quad \text{and} \quad P(k_{\text{in}}) dk_{\text{in}} = P(k) dk, \quad (\text{B.6})$$

which yield

$$P(k_{\text{out}}) = P(k) \frac{dk}{dk_{\text{out}}} \text{ and } P(k_{\text{in}}) = P(k) \frac{dk}{dk_{\text{in}}}. \quad (\text{B.7})$$

Setting $k = f_1^{-1}(k_{\text{out}})$ and $k = f_2^{-1}(k_{\text{in}})$, we can obtain the distributions.

Using

$$k_{\text{out}} = f_1(k) \text{ and } k_{\text{in}} = f_2(k),$$

we obtain

$$P(k_{\text{out}}) = P(k) \frac{1}{\frac{dk_{\text{out}}}{dk}} = P(f_1^{-1}(k_{\text{out}})) \frac{1}{f_1'(f_1^{-1}(k_{\text{out}}))}, \quad (\text{B.8})$$

and

$$P(k_{\text{in}}) = P(f_2^{-1}(k_{\text{in}})) \frac{1}{f_2'(f_2^{-1}(k_{\text{in}}))}. \quad (\text{B.9})$$

In general, it is difficult to obtain an explicit expression. However, for some specific values of λ or γ , analytical results are available.

Case I: $\lambda = 0.5$.

In this case, we have

$$k_{\text{out}} = k_{\text{in}} = \frac{k}{2}, \quad (\text{B.10})$$

and

$$k = 2k_{\text{out}} = 2k_{\text{in}}. \quad (\text{B.11})$$

Thus

$$P(k_{\text{out}}) = 2P(2k_{\text{out}}) = 2C(2k_{\text{out}})^{-\gamma} = 2^{1-\gamma}Ck_{\text{out}}^{-\gamma}.$$

Akin to $P(k_{\text{out}})$, we have

$$P(k_{\text{in}}) = P(k_{\text{out}}) = 2^{1-\gamma}Ck_{\text{in}}^{-\gamma}.$$

Case II: $\lambda = 0$.

$$k_{\text{out}} = Ak^{3-\gamma} \Rightarrow k = \frac{k_{\text{out}}^{\frac{1}{3-\gamma}}}{A}. \quad (\text{B.12})$$

We then have

$$P(k_{\text{out}}) = C \left(\frac{k_{\text{out}}}{A} \right)^{\frac{1}{3-\gamma}(-\gamma)} \frac{dk}{dk_{\text{out}}} = C \left(\frac{k_{\text{out}}}{A} \right)^{\frac{\gamma}{\gamma-3}} \frac{1}{3-\gamma} \left(\frac{k_{\text{out}}}{A} \right)^{\frac{1}{3-\gamma}-1} \frac{1}{A} = \frac{C}{3-\gamma} A^{\frac{1}{3-\gamma}} k_{\text{out}}^{\frac{2}{\gamma-3}}.$$

Case III: $\lambda = 1$.

$$k_{\text{in}} = Ak^{3-\gamma} \Rightarrow k = \frac{k_{\text{in}}^{\frac{1}{3-\gamma}}}{A}, \quad (\text{B.13})$$

which yields

$$P(k_{\text{in}}) = P(k_{\text{out}}) = \frac{C}{3-\gamma} A^{\frac{1}{3-\gamma}} k_{\text{in}}^{\frac{2}{\gamma-3}}. \quad (\text{B.14})$$

Case IV: $\gamma = 3$. For example, for BA model, we have

$$k_{\text{out}} = (1 - 2\lambda)A + \lambda k, \quad (\text{B.15})$$

and

$$k = \frac{k_{\text{out}} + (2\lambda - 1)A}{\lambda}. \quad (\text{B.16})$$

$$P(k_{\text{out}}) = C \left(\frac{k_{\text{out}} + (2\lambda - 1)A}{\lambda} \right)^{-\gamma} \frac{1}{\lambda} = C \lambda^{\gamma-1} \left[k_{\text{out}} + (2\lambda - 1) \frac{C}{\langle k \rangle} \right]^{-\gamma} = C \lambda^{\gamma-1} \left[k_{\text{out}} + (2\lambda - 1)k_{\text{min}} \right]^{-\gamma}.$$

Similarly, we have

$$k_{\text{in}} = (2\lambda - 1)A + (1 - \lambda)k, \quad (\text{B.17})$$

and

$$k = \frac{k_{\text{in}} + (1 - 2\lambda)A}{1 - \lambda}, \quad (\text{B.18})$$

so

$$P(k_{\text{in}}) = C \left[\frac{k_{\text{in}} + (1 - 2\lambda)A}{1 - \lambda} \right]^{-\gamma} \frac{1}{1 - \lambda} = C(1 - \lambda)^{\gamma-1} [k_{\text{in}} + (1 - 2\lambda)k_{\text{min}}]^{-\gamma}. \quad (\text{B.19})$$

Supplementary Note 2: Additional numerical results

2.1 Condition number and control error.

The correlation between the condition number C_W and the control error e_X is shown in Supplementary Fig. 62. We observe that, within a certain range of C_W , an approximate scaling relation exists between C_W and e_X , shown in panels (a), (c), (e), and (g). However, the scaling disappears outside the shown C_W range. The reason is that, outside the C_W range, the Gramian matrix W is ill conditioned, leading to considerable errors when computing the matrix inverse. In principle, the scaling regime can be extended with improved computational precision, but not indefinitely.

2.2 More on structural and practical controllability measures.

Supplementary Figure 63 shows the measure of the structural controllability, n_D , and the measure of the practical controllability, $P(\bar{C}_W)$, versus P_b for ER random and BA scale-free networks of size $N = 200$. We see that the structural and practical controllability cannot be simultaneously optimized irrespective of the network size.

2.3 More on control energy power law distribution.

Supplementary Figure 64 shows the robustness of the control energy power-law distribution against varying network size for both random [(a)] and scale-free [(b)] topology.

Fig. 62. Condition Number C_W Versus Control Error e_X for Random and Scale-Free Networks. Network size is $N = 100$ for (a-d) and 200 for (e-h), average degree is $\langle k \rangle = 6$ for ER random networks [(a),(b),(e), and (f)] and 8 for BA scale-free networks [(c), (d), (g), and (h)]. Directional link probability between any pair of nodes is $P_b = 0.1$. Panels (a),(c),(e), and (g) show the scaling relation between the condition number C_W and control error e_X . Panels (b), (d), (f), and (h) show the fraction R_{CW} of the networks with a certain C_W number. The scaling relation holds within some C_W - e_X region with boundaries specified as the black dashed lines. The e_X values are not physically meaningful outside the boundaries that are defined according to the precision limit of computation. The thresholds of C_W and e_X used in the computations are 10^{12} and 10^{-4} , respectively, which are indicated as the blue dashed lines. The threshold values are chosen to lie within the physical boundaries so that the calculations for all C_W values are meaningful.

Fig. 63. Practical Controllability Measures for Directed Networks. Measure of the practical controllability, $P(\bar{C}_W)$, for (a) ER random networks and (b) BA scale-free networks of size $N = 200$.

Fig. 64. Distributions of Control Energy for Practically Controllable Networks Under Different Network Sizes. Panels (a) and (b) are for random networks with $\langle k \rangle = 6$ and scale-free networks with $\langle k \rangle = 8$, respectively, for two values of the network size ($N = 100$ and $N = 200$), where we set $\bar{C}_W = 10^{12}$ and $t_f = 1$. In all cases, we observe an algebraic (power-law) scaling behavior.

Supplementary Note 3: Control Energy of One-Dimensional String

As shown in Supplementary Fig. 65, the energy required to control a unidirectional 1D string nearly overlaps with that of a bidirectional one with identical weights. In fact, if the chains are not too long, the relative difference in the energy between the two case are within the same order of magnitude. Here we provide an analytical calculation of the control energy for a bidirectional 1D chain network.

The energy E is given by

$$E(t_f) = \mathbf{x}_0^T \cdot H^{-1} \cdot \mathbf{x}_0, \quad (\text{B.20})$$

where $H \equiv e^{-At_f} \cdot W \cdot e^{-A^T t_f}$, $\mathbf{x}_0$ is the initial state of the network, and W is the Gramian matrix. Note that $\mathbf{x}_0$ can have arbitrary values but fixed after initial time. We stress that, again, for the discussion in this paper, $\mathbf{x}_0$ is an arbitrary but fixed value. Since H is positive definite and symmetric, its inverse H^{-1} can be decomposed in terms of its eigenvectors as $H^{-1} = Q\Lambda Q^T$, where $Q = [q_1, q_2, \dots, q_N]$ is composed of the orthonormal eigenvectors that satisfy $QQ^T = Q^T Q = I$, and $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_N\}$ is the diagonal eigenvalue matrix of H^{-1} in descending order. Numerically, we find that λ_1 is typically much larger than other eigenvalues. We thus have

$$E(t_f) = \mathbf{x}_0^T Q\Lambda Q^T \mathbf{x}_0 = \sum_{i=1}^n \lambda_i (q_i^T \mathbf{x}_0)^2 \approx \lambda_1 (q_1^T \mathbf{x}_0)^2. \quad (\text{B.21})$$

Since $\mathbf{x}_0$ can be chosen arbitrarily, in rare cases, for example, if we set $\mathbf{x}_0 = q_1$, then Eq. (B.21) becomes

$$E(t_f) \approx \lambda_1 (q_1^T q_1)^2 = \lambda_1. \quad (\text{B.22})$$

For an undirected network, the adjacency matrix A is positive definite and symmetric. We can decompose A into the form $A = VSV^T$, where the columns of V constitute the orthonormal eigenvectors of A and $S = \text{diag}\{s_1, s_2, \dots, s_N\}$ is the diagonal eigenvalue matrix of A in descending

order. We thus have $H = e^{-At_f} W e^{-A^T t_f} = V e^{-S t_f} V^T W V e^{-S t_f} V^T$. Let

$$\Lambda_H = \text{diag}\{\lambda_{H_1}, \lambda_{H_2}, \dots, \lambda_{H_N}\} = \text{diag}\{1/\lambda_N, 1/\lambda_{N-1}, \dots, 1/\lambda_1\}$$

be the eigenvalue matrix of H in descending order. The energy can thus be expressed as

$$E(t_f) \approx \lambda_1 (q_1^T \mathbf{x}_0)^2 = \lambda_{H_N}^{-1} (q_1^T \mathbf{x}_0)^2.$$

Letting $\Lambda_W = \text{diag}\{\lambda_{W_1}, \lambda_{W_2}, \dots, \lambda_{W_N}\}$ be the eigenvalue matrix of W in descending order.

We can approximate the eigenvalue of H by W , which has been numerically validated: $\Lambda_H \approx \Lambda_W$.

We thus have

$$\varepsilon \approx \lambda_1 (q_1^T \mathbf{x}_0)^2 = \lambda_{H_N}^{-1} (q_1^T \mathbf{x}_0)^2 \approx \lambda_{W_N}^{-1} (q_1^T \mathbf{x}_0)^2. \quad (\text{B.23})$$

Since orthonormal transform does not alter the eigenvalues of a given matrix, we have $\Lambda_H = e^{-S t_f} \Lambda_W e^{-S t_f}$.

For an undirected chain, the adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & \ddots & & \\ & 1 & \ddots & 1 & \\ & & \ddots & 0 & 1 \\ & & & 1 & 0 \end{bmatrix}_{N \times N},$$

control matrix is $B = [1, 0, \dots, 0]^T$, and eigenvalues and eigenvectors of A are

$$s_i = 2 \cos \left(\frac{\pi}{N+1} i \right), \quad i = 1, \dots, N, \quad (\text{B.24})$$

$$V_j^{(i)} = \sqrt{\frac{2}{N+1}} \sin \left(\frac{\pi}{N+1} i j \right), \quad i, j = 1, \dots, N. \quad (\text{B.25})$$

Recall that $H = V e^{-S t_f} \left(\int_0^{t_f} e^{S t} V^T B B^T V e^{S t} dt \right) e^{-S t_f} V^T$. Substituting this in Eqs. (B.24) and (B.25), after some algebraic manipulation, we obtain

$$H = \frac{1}{N+1} V W P W V^T, \quad (\text{B.26})$$

where

$$W = \begin{bmatrix} \sin(\theta) & & & \\ & \sin(2\theta) & & \\ & & \ddots & \\ & & & \sin(N\theta) \end{bmatrix}_{N \times N}$$

and

$$P_{jk} = \int_0^{2t_f} e^{-[\cos(j\theta) + \cos(k\theta)]t} dt$$

with $\theta = \pi/(N+1)$, $j, k = 1, \dots, N$.

As a result, we have $S = \text{diag}\{2\cos(\frac{\pi}{N+1}), 2\cos(\frac{2\pi}{N+1}), \dots, 2\cos(\frac{N\pi}{N+1})\}$. The minimum eigenvalue of H is given by

$$\lambda_{H_N} = e^{-2\cos(\frac{N\pi}{N+1})t_f} \lambda_{W_N} e^{-2\cos(\frac{N\pi}{N+1})t_f} = e^{-4\cos(\frac{N\pi}{N+1})t_f} \lambda_{W_N} = 1/\lambda_1. \quad (\text{B.27})$$

The Rayleigh-Ritz theorem can be used to bound P as:

$$\lambda_{P_N} \leq \frac{y^T P y}{y^T y} \leq \lambda_{P_1}, \quad (\text{B.28})$$

where $y = [y_1, y_2, \dots, y_N]^T$ is an arbitrary nonzero column vector, λ_{P_N} and λ_{P_1} are the maximal and minimal eigenvalues of P , respectively. Letting $T = 2t_f$, we have

$$\begin{aligned} y^T P y &= (y_1 \cdots y_N) \left[\int_0^T e^{-[\cos(j\theta) + \cos(k\theta)]\tau} d\tau \right]_{N \times N} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \\ &= \sum_{j,k=1}^N y_j y_k \int_0^T e^{-[\cos(j\theta) + \cos(k\theta)]\tau} d\tau \\ &= \sum_{j,k=1}^N \langle y_j e^{-[\cos(j\theta)t]}, y_k e^{-[\cos(k\theta)t]} \rangle \\ &= \left\langle \sum_{j=1}^N y_j e^{-[\cos(j\theta)t]}, \sum_{j=1}^N y_j e^{-[\cos(j\theta)t]} \right\rangle, \end{aligned} \quad (\text{B.29})$$

Fig. 65. Comparison Between Energies Required to Control a Unidirectional and a Bidirectional 1D Chains: (a) energies required to control a unidirectional chain E_{uni} (red) and a bidirectional one E_{bi} (blue) versus chain length L , and (b) the relative energy difference $|E_{\text{uni}} - E_{\text{bi}}|/E_{\text{uni}}$ versus chain length L .

with $\langle f, g \rangle \equiv \int_0^T f g \, d\tau$.

Letting $b_j = e^{-\cos(j\theta)}$ and performing a Taylor expansion of b_j^t around $t = 0$, we obtain

$$b_j^t = \sum_{k=0}^{N-1} [-\cos(j\theta)]^k \frac{t^k}{k!} + [-\cos(j\theta)]^N \frac{t_j^N}{N!} \quad (\text{B.30})$$

with $t_j \in [0, T]$. Now letting

$$q_j(t) = \sum_{k=0}^{N-1} [-\cos(j\theta)]^k \frac{t^k}{k!},$$

we have $b_j^t = q_j(t) + [-\cos(j\theta)]^N \cdot (t_j^N/N!)$. Consequently, the numerator in the Rayleigh quotient

can be expressed as

$$\begin{aligned}
y^T P y &= \left\langle \sum_{j=1}^N y_j b_j^t, \sum_{j=1}^N y_j b_j^t \right\rangle \\
&= \left\langle \sum_{j=1}^N \left(y_j q_j(t) + y_j [-\cos(j\theta)]^N \frac{t_j^N}{N!} \right), \sum_{j=1}^N \left(y_j q_j(t) + y_j [-\cos(j\theta)]^N \frac{t_j^N}{N!} \right) \right\rangle \\
&= \left\langle \sum_{j=1}^N y_j q_j(t), \sum_{j=1}^N y_j q_j(t) \right\rangle + 2 \left\langle \sum_{j=1}^N y_j q_j(t), \sum_{j=1}^N y_j [-\cos(j\theta)]^N \frac{t_j^N}{N!} \right\rangle \\
&\quad + \left\langle \sum_{j=1}^N y_j [-\cos(j\theta)]^N \frac{t_j^N}{N!}, \sum_{j=1}^N y_j [-\cos(j\theta)]^N \frac{t_j^N}{N!} \right\rangle \\
&\leq \underbrace{\left\langle \sum_{j=1}^N y_j q_j(t), \sum_{j=1}^N y_j q_j(t) \right\rangle}_{\text{Denote as } K_1} + \underbrace{2 \frac{T^N}{N!} \sum_{k=1}^N |y_k| \left(\left| \left\langle \sum_{j=1}^N y_j q_j(t), 1 \right\rangle \right| \right)}_{\text{Denote as } K_2} \\
&\quad + \underbrace{\left(\frac{T^N}{N!} \right)^2 \sum_{j,k=1}^N |y_j y_k| T}_{\text{Denote as } K_3}. \tag{B.31}
\end{aligned}$$

Since $y = [y_1, y_2, \dots, y_N]^T$ is an arbitrary nonzero column vector, for each N and T , we can choose $y = y_m$ insofar as K_1 and K_2 are relatively small compared with K_3 . We can normalize $y_m^T y$ to arrive at

$$\lambda_{P_N} \leq \frac{y_m^T P y_m}{y_m^T y_m} = \left(\frac{T^N}{N!} \right)^2 \sum_{j,k=1}^N |y_{m_j} y_{m_k}| T \sim \mathcal{O} \left(\frac{T^{2N}}{(N!)^2} \right), \tag{B.32}$$

where λ_{P_N} is the smallest eigenvalue of P . Recall that P is symmetric and positive definite, using Cholesky decomposition we can obtain its factorization [1] as $P = LL^T$, where L is the lower triangular matrix with its diagonal filled with square roots of eigenvalues of P . Therefore, Eq. (B.26) can be written as $H = \frac{1}{N+1} VWLL^T WV^T$. Since orthonormal transform does not change the eigenvalues of a matrix, H has the same eigenvalues as $R = \frac{1}{N+1} WLL^T W = \frac{1}{N+1} WL(WL)^T$. Suppose $\Lambda_P = \text{diag}\{\lambda_{P_1}, \lambda_{P_2}, \dots, \lambda_{P_N}\}$ is the diagonal eigenvalue matrix of P in descending order. The j th eigenvalue of R satisfies

$$\lambda_{R_j} = \frac{1}{N+1} \lambda_{P_j} (\sin k\theta)^2 \leq \frac{1}{N+1} \lambda_{P_j},$$

where j and k run from 1 to n . Therefore, assuming that $\mathbf{x}_0$ is arbitrary and fixed, the control energy $E(t_f)$ can then be approximated as

$$E(t_f) \sim \mathcal{O}(\lambda_{H_N}^{-1}) = \mathcal{O}\left((N+1)\frac{(N!)^2}{t_f^{2N}}\right). \quad (\text{B.33})$$

Fig. 66. Correlation Between Network Control Energy and the Smallest Eigenvalue of H -Matrix. Network size is $N = 100$, directional link probability between any pair of nodes is $P_b = 0.1$, and average degree is (a) $\langle k \rangle = 6$ for ER random networks and (b) $\langle k \rangle = 8$ for BA scale-free networks.

Supplementary Note 4: Correlation between network control energy and smallest eigenvalue of H -matrix

Strong correlation between the average network control energy, $\langle E \rangle$, and the smallest eigenvalue of the H -matrix, $\lambda_{H_N}^{-1}$, for ER random and BA scale-free networks can be observed in Supplementary Fig. 66, indicating that the network control energy is essentially determined by the smallest eigenvalue of its H -matrix.

Supplementary Note 5: LCC-skeleton and double-chain interaction models

5.1 LCC-skeleton model.

Referring to Fig. 4 in the main text, we assume that the control chains are independent of each other so that E_2 is negligible as compared with E_1 . Each control chain is effectively a 1D string. Due to the exponential increase in energy via chain length increment, E_1 can be regarded as the sum of control energies associated with the set of unidirectional 1D strings, to which the contribution of the LCC dominates. The required energy to control the full network can thus be approximated as that required to control all LCCs,

$$E = E_1 + E_2 \approx E_1 \approx m \cdot E_L \approx m \cdot \lambda_{H_L}^{-1}, \quad (\text{B.34})$$

where E_L denotes the energy required to control an LCC, λ_{H_L} is the smallest eigenvalue of the LCC's H matrix H_L , and m denotes the degeneracy (multiplicity) of the LCC, as shown in Fig. 4(b) in the main text. Results presented in Fig. 3(b) of the main text demonstrate a positive correlation between E and $m \cdot E_L$, reinforcing the idea the independent LCCs are the key topological structure dictating the energy required to control the whole network. In particular, if a network contains long LCCs (as can be determined straightforwardly by maximum matching from the structural controllability theory [2]), there is high likelihood that it cannot be practically controlled as practically the required energy would diverge.

Reasoning from an alternative standpoint, an arbitrary combination of D_C and m effectively represents a network, as shown in Fig. 4(b) in the main text, and the entire network ensemble can be represented by the ensemble of all possible combinations of LCCs. In the LCC ensemble, the quantities D_C and m emerge according to their probability density functions, $P_{D_C}(D_C)$ and $P_m(m)$, respectively, and the appearance of an arbitrary pair of D_C and m is determined by their joint probability density function $P(D_C, m)$. Consequently, the distribution of the energy required

to control the original network can be characterized accurately by the distribution of the energy required to control the LCC skeleton in the corresponding ensemble.

Supplementary Figure 67(a) shows the distribution of the control diameter D_C , essentially the length distribution of LCCs. The probability density function decays approximately exponentially with D_C , so we write

$$P_{D_C}(D_C) = a \cdot e^{-b \cdot D_C}, \quad (\text{B.35})$$

where a and b are positive constants. Using the relationship between E_L and D_C [e.g., Fig. 3(a) in the main text], we have

$$E_L \approx A \cdot e^{B \cdot D_C} \Rightarrow D_C \approx \frac{1}{B} \ln \frac{E_L}{A}, \quad (\text{B.36})$$

where A and B are positive constants. The probability density function of E_L can then obtained as

$$P_L(E_L) = P_{D_C}\left(\frac{1}{B} \ln \frac{E_L}{A}\right) \cdot \left| \frac{d\left(\frac{1}{B} \ln \frac{E_L}{A}\right)}{dE_L} \right| \approx \frac{a}{B} A^{\frac{b}{B}} \cdot E_L^{-(1+\frac{b}{B})}. \quad (\text{B.37})$$

In the ER random network ensemble, the probability density of LCC degeneracy m for networks with $D_C > 2$ also exhibits an exponential decay, as shown in Supplementary Fig. 67(b):

$$P_m(m) = c \cdot e^{-g \cdot m}, \quad (\text{B.38})$$

where c and d are positive constants.

Since the control energy depends monotonously on the control diameter D_C , the energy dependence on m can be revealed by examining the correlation between D_C and m , which can in general be either positive or negative. From Eq. (B.38), we see that $P_m(m)$ increases exponentially with m , implying a positive correlation:

$$D_C = s_1 + s_2 \cdot m, \quad (\text{B.39})$$

where s_1 and s_2 are positive constants. This form of relation ensures that $P_m(m)$ has the form in Eq. (B.38). Positive correlation, however, means that the number of LCCs increases with its length,

Fig. 67. Probability Distribution of Control Diameter and Its Degeneracy. (a) Distribution of D_C (blue) and the probability that an LCC of length D_C appears in the random network ensemble versus D_C . (b) Probability distribution of LCC-degeneracy m for networks with $D_C > 2$.

which is unphysical for random networks. These arguments suggest that a contradiction can arise if we assume either positive or negative correlation between D_C and m . A natural resolution is that these two quantities are independent of each other. Since E_L , the energy required to control a chain of length L , is determined mainly by the control diameter D_C , E_L and m can be assumed to be independent of each other so that their joint probability density function can be expressed as $P(E_L, m) \approx P_L(E_L) \cdot P_m(m)$.

Having obtained $P_L(E_L)$ and $P_m(m)$, we can calculate the cumulative probability distribution function of the estimated control energy $E = m \cdot E_L$ required to control the original network. We have

$$\begin{aligned}
 F_E(E) &= P(m \cdot E_L < E) = \int_0^\infty \left[\int_0^{\frac{E}{E_L}} P(E_L, m) \cdot dm \right] dE_L \\
 &= \int_0^\infty \left[\int_0^{\frac{E}{E_L}} P_L(E_L) \cdot P_m(m) \cdot dm \right] dE_L \approx \frac{caA^{\frac{b}{B}}}{gB} \cdot \left\{ -\frac{b}{B} - \left[\Gamma\left(\frac{b}{B}\right) - \Gamma\left(\frac{b}{B}, gE\right) \right] \cdot (gE)^{-\frac{b}{B}} \right\},
 \end{aligned} \tag{B.40}$$

where $\Gamma(\frac{b}{B})$ and $\Gamma(\frac{b}{B}, gE)$ are the Gamma and incomplete Gamma function, respectively. Thus,

the probability density function of E can then be expressed as

$$P_E(E) = \frac{dF_E(E)}{dE} \approx \frac{caA^{\frac{b}{B}}}{gB} \cdot \left\{ -\frac{e^{-gE}}{E} + \left[\Gamma\left(\frac{b}{B}\right) - \Gamma\left(\frac{b}{B}, gE\right) \right] \cdot (gE)^{-(1+\frac{b}{B})} \right\}, \quad (\text{B.41})$$

where the first term $-e^{-gE}/E$ can be neglected due to the typically large value of E . Since we observe numerically that the difference between the two Gamma functions is approximately constant:

$h(\Gamma) \equiv \Gamma(\frac{b}{B}) - \Gamma(\frac{b}{B}, gE) \approx 1.7$, we can simplify Eq. (B.41) as

$$P_E(E) \approx C \cdot E^{-(1+\frac{b}{B})}, \quad (\text{B.42})$$

where $C = [\frac{caA^{\frac{b}{B}}}{gB} \cdot g^{-(2+\frac{b}{B})} \cdot h_\Gamma]$ is a positive constant. Equation (B.42) indicates a power-law distribution of the control energy, providing an analytical explanation to the numerically discovered energy distribution for practically controllable networks, as exemplified in Fig. 3 in the main text. To get a rough idea about the value of the power-law scaling exponent, say we take $B \approx 2$ and $b \approx 1$ (typical numerical values). A theoretical estimate of the power-law exponent is thus $1 + b/B \approx 1.5$, which is consistent with the value obtained from results from direct numerical simulation. The fact that the distribution of E_L is power law with the identical exponent provides additional support for our assumption that the LCC degeneracy m plays little role in determining the control energy. It is the combination of the exponential decay in the probability distribution of the control diameter [cf., Eq. (B.35)] and the exponential increase in the energy required to control LCC with its length [cf., Eq. (B.36)] that gives rise to the power-law energy distribution of the LCCs, which ultimately leads to the power-law distribution in the actual energy required to control the original random network.

We see that the control diameter of a network is a key quantity determining the required control energy. The topological diameter, on the other hand, is a fundamental quantity characterizing, for example, the small-world structure of the network [3]. An interesting issue concerns the relation between the control and topological diameters. In particular, if the network has a large diameter,

does it mean that its control diameter must be large as well? This issue has been addressed, with the finding that there is little correlation between the two types of diameters.

5.2 Double-chain interaction model.

Our analysis of the LCC-skeleton model predicts power-law distribution of the required energy for practically controllable networks, which agrees qualitatively with numerics. However, in the model interactions among the coexisting chains are ignored. In a physical system, interactions among the basic components usually plays an important role in determining the system's properties. To obtain a more accurate estimate of the behaviors of the control energy, we need to include the interactions among the chains. The necessity is further justified as there are discrepancies between the actual control energy and that from the LCC-skeleton model, as exemplified in Fig. 3(b) in the main text. In particular, there is an approximately continuous distribution in the energy required to control the actual network, but the distribution of the energy from the LCC-skeleton model tends to aggregate into a number of subintervals, each corresponding to a certain value of the control diameter associated with an LCC. Thus, in order to reproduce the numerically obtained energy distributions, we must incorporate the interactions among the LCCs into the model. However, including the interactions makes analysis difficult, as there are typically a large number of interacting pairs of chains. To gain insight into the role played by the interactions, it is useful to focus on the relatively simple case of two interacting chains.

Our double-chain interaction model is constructed, as follows. Consider two identical unidirectional chains, denoted by $C1$ and $C2$, each of length D_C . Every node in $C1$ connects with every node in $C2$ with probability p , all links between the two chains are unidirectional. A link points to $C2$ from $C1$ with probability $p_{1 \rightarrow 2}$ and the probability for a link in the opposite direction is $p_{2 \rightarrow 1} = 1 - p_{1 \rightarrow 2}$. By changing the connection rate p and the directional bias $p_{1 \rightarrow 2}$, we

Fig. 68. Distribution of control energy in a double-chain interaction models. (a) Two chains of $D_C = 5$ (red) and $D_C = 6$ (black), where the upper panel is a schematic illustration of two LCCs of identical length $D_C = 5$ interacting with each other via some random links (green) between them. (b) Two chains with their lengths randomly chosen from 3 – 6, where the upper panel shows the case of two interacting chains of length 5 and 3. The longer chain (red) plays the role of LCC, while the shorter chain is a non-LCC (orange).

can simulate and characterize various interaction patterns between the two chains. To be concrete, we generate an ensemble of 10000 interacting double-chain networks, each with $2 \cdot D_C$ nodes and multiple randomized interchain links as determined by the parameters p and $p_{1 \rightarrow 2}$. As shown in Supplementary Fig. 68(a), the distribution of the control energy displays a remarkable similarity to that for random networks, in that a power-law scaling behavior emerges with the exponent about 1.5. A striking result is that the energy distributions from the double-chain interaction model are much more smooth than those from the LCC-skeleton model, indicating the key role played by the interchain interactions in spreading out the control energies that are clustered when the interactions are absent. The power-law distribution holds robustly with respect to variations in the parameters p and $p_{1 \rightarrow 2}$. In addition, to reveal the role of the interaction between an LCC and a non-LCC chain in the control energy, we randomly pick their lengths from $[3, 6]$ with equal probability, where the longer chain acts as an LCC. Again, we observe a strong similarity between the energy distributions from random networks and from this model, as shown in Supplementary Fig. 68(b), suggesting a universal pattern followed by pair interactions, regardless of the length of the chains. In particular, interactions between two chains, LCC or not, have similar effect on the control-energy distribution. These results indicate that the double-chain interaction model captures the essential physical ingredients of the energy distribution in controlling complex networks.

Supplementary Note 6: Energy optimization of a cascade parallel R-C circuit network

6.1 Network representation of a circuit system

We consider a cascade parallel R-C circuit consisting of three identical resistors and capacitors as an example to illustrate how the circuit can be abstracted into a directed network, as shown in Supplementary Fig. 69. For convenience, we set $R_1 = R_2 = R_3 = R$ and $C_1 = C_2 = C_3 = C$, and denote the currents through R_1 , R_2 , and R_3 as $i_1(t)$, $i_2(t)$, and $i_3(t)$, respectively. The equations of the circuit are

$$\left\{ \begin{array}{l} u(t) = i_1(t)R + u_1(t) \\ u_1(t) = i_2(t)R + u_2(t) \\ u_2(t) = i_3(t)R + u_3(t) \\ C \frac{du_1(t)}{dt} = i_1(t) - i_2(t) \\ C \frac{du_2(t)}{dt} = i_2(t) - i_3(t) \\ C \frac{du_3(t)}{dt} = i_3(t) \end{array} \right. \quad (\text{B.43})$$

After some algebraic manipulation, we have

$$\left\{ \begin{array}{l} \frac{du_1(t)}{dt} = -\frac{2}{RC}u_1(t) + \frac{1}{RC}u_2(t) + \frac{1}{RC}u(t) \\ \frac{du_2(t)}{dt} = \frac{1}{RC}u_1(t) - \frac{2}{RC}u_2(t) + \frac{1}{RC}u_3(t) \\ \frac{du_3(t)}{dt} = \frac{1}{RC}u_2(t) - \frac{1}{RC}u_3(t), \end{array} \right. \quad (\text{B.44})$$

which can be written as

$$\begin{pmatrix} \frac{du_1(t)}{dt} \\ \frac{du_2(t)}{dt} \\ \frac{du_3(t)}{dt} \end{pmatrix} = \begin{pmatrix} -\frac{2}{RC} & \frac{1}{RC} & 0 \\ \frac{1}{RC} & -\frac{2}{RC} & \frac{1}{RC} \\ 0 & \frac{1}{RC} & -\frac{1}{RC} \end{pmatrix} \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{pmatrix} + \begin{pmatrix} \frac{1}{RC} \\ 0 \\ 0 \end{pmatrix} u(t). \quad (\text{B.45})$$

Fig. 69. Controlling and Optimizing a Cascade Parallel R-C Circuit and the Corresponding Network Presentation. (a) A cascade parallel R-C circuit with 3 resistors (R_1 , R_2 , and R_3 , each of resistance 1Ω) and 3 capacitors (C_1 , C_2 , and C_3 , each of capacitance $1F$), where $u(t)$ is the external input voltage, $u_1(t)$, $u_2(t)$, and $u_3(t)$ are the voltages on the capacitors C_1 , C_2 , and C_3 , respectively, $i_1(t)$, $i_2(t)$, and $i_3(t)$ are the currents through the resistors R_1 , R_2 , and R_3 , respectively. (b) Network representation of the circuit in (a). (c) Circuit with an extra external current input $i_e(t)$ into the capacitor C_2 . (d) The extra external current input $i_e(t)$ serves as a redundant control input injected into node 2 of the network in (b). There are two driver nodes (yellow) in the network: 1 and 3.

Setting $R = 1\Omega$ and $C = 1F$, we have

$$\begin{pmatrix} \frac{du_1(t)}{dt} \\ \frac{du_2(t)}{dt} \\ \frac{du_3(t)}{dt} \end{pmatrix} = A \cdot \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{pmatrix} + B \cdot u(t), \quad (\text{B.46})$$

where

$$A = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -1 \end{pmatrix} \quad (\text{B.47})$$

is the adjacency matrix of the network representing the circuit, and

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (\text{B.48})$$

is the control input matrix. The circuit has then been transferred into a 3-node bidirectional 1D chain network with adjacency matrix A .

6.2 The implementation of extra control input

Without loss of generality, we inject an extra external current input $i_e(t)$ into the capacitor C_2 , and the circuit equations become:

$$\left\{ \begin{array}{l} u(t) = i_1(t)R + u_1(t) \\ u_1(t) = i_2(t)R + u_2(t) \\ u_2(t) = i_3(t)R + u_3(t) \\ C \frac{du_1(t)}{dt} = i_1(t) - i_2(t) \\ C \frac{du_2(t)}{dt} = i_2(t) - i_3(t) + i_e(t) \\ C \frac{du_3(t)}{dt} = i_3(t) \end{array} \right. \quad (\text{B.49})$$

The state equations are

$$\begin{pmatrix} \frac{du_1(t)}{dt} \\ \frac{du_2(t)}{dt} \\ \frac{du_3(t)}{dt} \end{pmatrix} = A \cdot \begin{pmatrix} u_1(t) \\ u_2(t) \\ u_3(t) \end{pmatrix} + B_e \begin{pmatrix} u(t) \\ i_e(t) \end{pmatrix}, \quad (\text{B.50})$$

where

$$B_e = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \quad (\text{B.51})$$

is the control input matrix of the circuit under the original control input $u(t)$ on node 1 and a redundant control input $i_e(t)$ to node 2. Similarly, the redundant control input can be injected into any capacitor.

It is necessary to keep all other nodes unaffected while introducing exactly one extra control input into the circuit. However, any additional voltage change in any part of the circuit can lead to voltage changes on all the capacitors. A change in the current through a capacitor will not affect the currents in other components of the network, since only the time derivative of its voltage is affected. Thus, a meaningful way to introduce an extra control signal input to one node of a circuit's network is to inject current into one particular capacitor in the circuit.

Supplementary Note 7: The practical controllability of real-world networks

Supplementary Table 7 lists the names and types of the real-world networks studied and Supplementary Table 8 presents more detailed information about the controllability of the 18 real-world networks analyzed in the main text.

TABLE 7

Description of the 18 real-world networks used in the paper (N - number of nodes; M - number of edges).

Type	Index	Name	N	M	Description
Trust	1	College Student [4, 5]	32	96	Social network
	2	Prison Inmate [4, 5]	67	182	Social network
Circuits	3	s208a [6]	122	189	Logic circuit
	4	s420a [6]	252	399	Logic circuit
	5	s838a [6]	512	189	Logic circuit
Citation	6	Small World [3]	233	1988	Stanley Milgram
	7	Kohonen [7]	3772	96	T. Kohonen
Protein	8	Protein-1 [5]	95	213	Protein network
	9	Protein-2 [5]	53	123	Protein network
	10	Protein-3 [5]	99	212	Protein network
Food Web	11	St. Martin [8]	45	224	Food Web
	12	Seagrass [9]	49	226	Food Web
	13	Grassland [10]	88	137	Food Web
	14	Ythan [10]	135	601	Food Web
	15	Silwood [11]	154	370	Food Web
	16	Little Rock [12]	183	2494	Food Web
	17	Baydry [13]	128	2137	Food Web
	18	Florida [13]	128	2106	Food Web

TABLE 8

Practical controllability of real-world networks studied in Ref. [2], where N_D denotes the number of controllers as determined by the structural controllability theory and M^* is the number of augmented driver nodes needed to make the network practically controllable. The densities of the original driver nodes and with M^* augmented drivers included are $n_D = N_D/N$ and $n_D^* = N_D^*/N$, respectively, where $N_D^* = N_D + M^*$; n_D^* is the new measure of controllability; and E^* denotes the new energy.

Type	Name	N	N_D	M^*	n_D	n_D^*	E^*
Trust	Coll. Student	32	6	0	0.19	0.19	7.9×10^{10}
	Prison Inmate	67	9	27	0.13	0.54	2.3×10^7
Electronic Circuits	s208a	122	29	40	0.24	0.57	2.3×10^7
	s420a	252	59	71	0.23	0.52	2.6×10^6
	s838a	512	119	81	0.23	0.39	4.1×10^9
Citation	Small World	233	140	11	0.60	0.65	2.0×10^3
	Kohonen	3772	2114	413	0.56	0.67	9.4×10^4
Protein	Protein-1	95	48	19	0.51	0.71	1.9×10^{10}
	Protein-2	53	13	0	0.25	0.25	3.7×10^9
	Protein-3	99	22	35	0.22	0.58	3.9×10^5
	St. Martin	45	14	9	0.31	0.51	2.7×10^3
Food Web	Seagrass	49	13	7	0.27	0.41	3.6×10^3
	Grassland	88	46	0	0.52	0.52	3.4×10^5
	Ythan	135	69	14	0.51	0.62	2.6×10^3
	Silwood	154	116	12	0.75	0.83	1.5×10^4
	Little Rock	183	99	36	0.54	0.74	5.4×10^3
	Baydry	128	62	34	0.48	0.75	1.7×10^3
	Florida	128	30	69	0.23	0.77	NaN

TABLE 9

In addition to Table 8, when an additional control signal is added to the middle of each LCC [strategy (I)] so that M_{mid}^* extra control inputs are used, the control energy is E_{mid}^* . For comparison, we use M_{mid}^* randomly chosen extra driver nodes and calculate the required control energy $E_{\text{R-mid}}^*$. We also test another strategy in which an extra control signal is applied to each of the m convergent nodes of all LCCs [strategy (II)], in which M_{end}^* additional control inputs are used. The control energy required is denoted as E_{end}^* , with $E_{\text{R-end}}^*$ being the control energy when the same number of additional controllers are chosen randomly. All energies are result of averaging over 1000 combinations of initial and final states. The control diameter D_C of each network is listed in the last column. (See Supplementary Table 7 for detailed description of the 18 networks.)

Type	Name	M_{mid}^*	E_{mid}^*	$E_{\text{R-mid}}^*$	M_{end}^*	E_{end}^*	$E_{\text{R-end}}^*$	D_C
Trust	Coll. Student	3	2.5×10^5	2.4×10^7	4	7.3×10^4	2.7×10^7	4
	Prison Inmate	1	4.6×10^3	2.2×10^7	1	9.7×10^4	1.8×10^7	5
Electronic Circuits	s208a	1	3.0×10^5	1.8×10^7	1	4.0×10^5	2.3×10^7	5
	s420a	11	2.1×10^5	1.3×10^6	10	2.2×10^5	1.5×10^6	4
	s838a	13	3.1×10^6	3.7×10^9	6	3.7×10^6	3.7×10^9	5
Citation	Small World	1	1.9×10^3	1.9×10^3	1	1.9×10^3	1.8×10^3	5
	Kohonen	49	4.0×10^4	9.0×10^4	37	5.1×10^4	9.0×10^4	3
Protein	Protein-1	7	8.5×10^2	1.9×10^9	5	2.2×10^3	1.3×10^5	3
	Protein-2	2	9.0×10^8	1.3×10^9	2	1.2×10^9	2.0×10^9	4
	Protein-3	3	5.1×10^4	3.1×10^5	3	5.3×10^4	3.6×10^5	4
	St. Martin	2	9.2×10^2	2.0×10^3	2	1.5×10^3	1.6×10^3	3
Food Web	Seagrass	3	8.3×10^2	9.8×10^2	2	3.4×10^3	2.4×10^3	3
	Grassland	1	8.6×10^4	3.3×10^5	1	9.2×10^4	3.4×10^5	4
	Ythan	4	1.9×10^3	2.0×10^3	2	2.0×10^3	2.3×10^3	3
	Silwood	3	1.0×10^3	1.8×10^4	2	9.9×10^2	1.2×10^4	3
	Little Rock	48	3.6×10^3	3.6×10^3	48	3.6×10^3	3.6×10^3	2
	Baydry	32	9.6×10^2	9.6×10^2	48	9.6×10^2	9.6×10^2	2
	Florida	1	NaN	240 NaN	1	NaN	NaN	3

REFERENCES

- [1] G. Strang, *Linear Algebra and Its Applications* (Academic Press, 1976).
- [2] Y.-Y. Liu, J.-J. Slotine, A.-L. Barabási, Controllability of complex networks. *Nature (London)* **473**, 167-173 (2011).
- [3] D. J. Watts, S. H. Strogatz, Collective dynamics of small-world networks. *Nature* **393**, 440-442 (1998).
- [4] M. A. J. V. Duijn, M. Huisman, F. N. Stokman, F. W. Wasseur, E. P. H. Zeggelink, Evolution of sociology freshmen into a friendship network. *J. Math. Sociol.* **27**, 153-191 (2003).
- [5] R. Milo, *et al.*, Superfamilies of evolved and designed networks. *Science* **303**, 1538-1542 (2004).
- [6] R. Milo, *et al.*, Network motifs: Simple building blocks of complex networks. *Science* **298**, 824-827 (2002).
- [7] T. A. Davis, Y. Hu, The university of florida sparse matrix collection. *ACM Trans. Math. Software (TOMS)* **38**, 1 (2011).
- [8] D. Baird, J. Luczkovich, R. R. Christian, Assessment of spatial and temporal variability in ecosystem attributes of the st marks national wildlife refuge, apalachee bay, florida. *Estuarine Coast. Shelf Sci.* **47**, 329-349 (1998).
- [9] R. R. Christian, J. J. Luczkovich, Organizing and understanding a winters seagrass foodweb network through effective trophic levels. *Ecol. Modelling* **117**, 99-124 (1999).
- [10] J. A. Dunne, R. J. Williams, N. D. Martinez, Food-web structure and network theory: the role of connectance and size. *Proc. Natl. Acad. Sci.* **99**, 12917-12922 (2002).
- [11] J. Memmott, N. D. Martinez, J. E. Cohen, Predators, parasitoids and pathogens: species richness, trophic generality and body sizes in a natural food web. *J. Animal Ecol.* **69**, 1-15 (2000).
- [12] N. D. Martinez, Artifacts or attributes? effects of resolution on the little rock lake food web. *Ecol. Mono.* **61**, 367-392 (1991).
- [13] R. E. Ulanowicz, D. L. DeAngelis, Network analysis of trophic dynamics in south florida ecosystems. *US Geo. Sur. Prog. South Florida Ecosystem* **114** (2005).