

# Reconstructing Network Dynamics of Coupled Discrete Chaotic Units from Data

Irem Topal\* and Deniz Eroglu†

*Faculty of Engineering and Natural Sciences, Kadir Has University, 34083 Istanbul, Turkey*

Reconstructing network dynamics from data is crucial for predicting the changes in the dynamics of complex systems such as neuron networks; however, previous research has shown that the reconstruction is possible under strong constraints such as the need for lengthy data or small system size. Here, we present a recovery scheme blending theoretical model reduction and sparse recovery to identify the governing equations and the interactions of weakly coupled chaotic maps on complex networks, easing unrealistic constraints for real-world applications. Learning dynamics and connectivity lead to detecting critical transitions for parameter changes. We apply our technique to realistic neuronal systems with and without noise on a real mouse neocortex and artificial networks.

*Dynamical networks*, including power grids, food webs, climate networks, and neuron networks, described by dynamical units oscillating on complex networks, are fundamental components of our everyday lives. The ability to regulate network dynamics is crucial for predicting, thus, controlling these systems' behavior to acquire the desired functionality. Neuron networks are an important class of dynamical networks for human wellness since the changes in the interaction can lead to undesired pathological situations. For instance, epileptic seizures are associated with emergent neural network synchronization when the dynamical network parameters change [1]. Consequently, it is vital to anticipate critical transitions to neuronal synchronization and invent predictive technologies to detect early warning signals to prevent potential tragedies [2]. In the case of neuron network dynamics, consisting of intrinsic neuron function and the coupling scheme between neurons, the critical transitions to synchronization are not directly determinable. Therefore, the governing equation must be recovered from the observations of the nodes for forecasting the critical transitions due to parameter changes.

The network dynamics reconstruction from data is a very active research field [3–8]. Various methods were proposed to infer the connectivity matrix under some constraints, such as the need for a system to be at steady-state [9] or requiring prior knowledge about the dynamics [10–12] or the coupling strength [13]. In addition to the studies that reveal the connectivity matrix by control signals or analytical solutions, statistical learning approaches such as compressed sensing were also introduced to learn entire unknown dynamics [14, 15], which also infers the connectivity structure. However, statistical learning techniques are not extendable for large networks or require long time series measurements. A natural question is then whether revealing the network dynamics of weakly interacting chaotic oscillators would be possible using relatively short data without requiring knowledge of the system's nodal behavior and coupling scheme. This question is especially relevant in weak coupling regimes, in which the synchronization regime is unstable and the decay of correlation is exponential for chaotic oscillators, meaning that similarity measures cannot capture the interaction topology.

This Letter reports a dynamical network reconstruction approach from time series observations by integrating mean-field approaches from dynamical systems theory with statistical learning tools. Neural networks are described by chaotic isolated dynamics [16], weakly interacting nodes [17] and interaction through scale-free type networks [18]. Our reconstruction approach assumes that we have the mentioned neuroscientific setting and access to all nodes' data while the local dynamics of the nodes, the coupling function between them and the interaction structure are unknowns. Our methodology accurately identifies them using rather short time series and is independent from the network size, which is important since it is, generally, impossible to have long real-world observations, and real networks are large. Finally, as the reconstruction methodology includes mean-field approximations, the inferred model may not estimate the exact future states of the system due to the chaotic nature of the dynamical units. However, the reconstructed model allows us to predict the emergent collective behavior of dynamical networks considering parameter change, which is crucial to avoid undesired behaviors for real-world applications such as epilepsy seizures.

*Model.* — The network dynamics of weakly coupled and identical  $n$  oscillators with interaction akin to diffusion is described by

$$\mathbf{x}_i(t+1) = \mathbf{f}(\mathbf{x}_i(t)) + \sum_{j=1}^n w_{ij} \mathbf{H}(\mathbf{x}_i(t), \mathbf{x}_j(t)) + \boldsymbol{\eta}_i(t) \quad (1)$$

where  $\mathbf{x}_i \in \mathbb{R}^m$ ,  $\mathbf{f}: \mathbb{R}^m \rightarrow \mathbb{R}^m$  represents the isolated dynamics of nodes and we assume it is chaotic [19].  $\mathbf{H}$

---

\*Electronic address: irem.topal@khas.edu.tr

†Electronic address: deniz.eroglu@khas.edu.tr

is a diffusive coupling function ( $\mathbf{H}(\mathbf{x}, \mathbf{x}) = \mathbf{H}(0) = 0$  and  $\mathbf{H}(\mathbf{x}, \mathbf{y}) = -\mathbf{H}(\mathbf{y}, \mathbf{x})$ ).  $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$  is the adjacency matrix of weighted and directed network where  $w_{ij} \geq 0$  is the interaction strength from node- $j$  to node- $i$ . The noise term,  $\boldsymbol{\eta}_i(t)$ , is uniformly distributed  $\|\boldsymbol{\eta}_i(t)\| \leq \eta_0$  for all nodes where  $\eta_0$  is noise intensity. This network dynamics, Eq. (5), is used to model numerous real-world applications including brain networks [20], power grids [21, 22], superconductors [23], and cardiac pacemaker cells [24].

*Reduction theorem.*— A low-dimensional reduction of Eq. (5) is key for our network dynamics reconstruction approach. The reduction theorem applies a mean-field approach that relies on two main statements: (i) the statistical behavior of nodes' dynamics (frequency distribution of states) must be preserved and (ii) a large portion of nodes must be interacting with at least a few nodes in the network. These statements are satisfied with the given assumptions for the reduction theorem: *chaotic local dynamics* of expanding maps and *weak coupling* (to preserve the nodes' state distribution against fluctuations due to the interactions or external noise) and *scale-free networks* (most of the nodes have small degrees  $k \sim n^\epsilon$ , and some nodes are hubs with degrees  $k \sim n^{\frac{1}{2} + \epsilon}$  where  $\epsilon$  is an arbitrarily small number), which also mimic brain network dynamics. Using the theorem, the coupling term of Eq. (1) can be reduced as follows:

$$\begin{aligned} \sum_{j=1}^n w_{ij} \mathbf{H}(\mathbf{x}_i, \mathbf{x}_j) &\approx k_i \int \alpha \tilde{\mathbf{H}}(\mathbf{x}_i, \mathbf{x}_j) d\mu(\mathbf{x}_j) \\ &= k_i \alpha (\tilde{\mathbf{V}}(\mathbf{x}_i) + \tilde{C}) = k_i \mathbf{V}(\mathbf{x}_i) + C \end{aligned}$$

where  $\mathbf{V}$  is the effective coupling function,  $k_i = \sum_j w_{ij}$  is the incoming degree of node  $i$ ,  $\mathbf{H} = \alpha \tilde{\mathbf{H}}$ ,  $\alpha$  is a multiplier for the coupling function,  $\mu$  is a physical measure of the isolated dynamics,  $C$  is the integration constant and the integral takes into account the cumulative effect of interactions on node- $i$ . As the coupling term is reduced as a function of an invariant measure  $\mu$ , the reduction theorem works for a system in a steady state. Furthermore, to apply this mean-field approach-based reduction theorem, the statistical properties of individual dynamical systems must be preserved, which is satisfied by chaotic oscillators and weak coupling (see Supp. Mat. II and Ref. [25]). Then Eq. (5) can be written as

$$\mathbf{x}_i(t+1) = \mathbf{f}(\mathbf{x}_i(t)) + k_i \mathbf{V}(\mathbf{x}_i(t)) + C + \boldsymbol{\kappa}_i(t) + \boldsymbol{\eta}_i(t) \quad (2)$$

where  $\boldsymbol{\kappa}_i(t)$  is a small fluctuation for an interval of time that is exponentially large and depends on the state of neighbors of the  $i$ th node.

*Reconstruction scheme.*— To learn isolated dynamics  $\mathbf{f}$  and coupling function  $\mathbf{H}$ , we first need to classify nodes regarding their degrees. According to the reduction theorem, nodes with a similar in-degree must have a similar governing equation. To identify the governing equations of each node independently from other nodes, we use sparse regression, particularly the Sparse Identification of Nonlinear Dynamical Systems (SINDy) technique [26]. We denote the data collection of node- $i$  by Eq.(5) as  $\mathcal{X}_i = [\mathbf{x}_i(1), \dots, \mathbf{x}_i(T-1)]^T$  and  $\mathcal{X}'_i = [\mathbf{x}_i(2), \dots, \mathbf{x}_i(T)]^T$ . SINDy performs a sparse regression for the linear equation  $\mathcal{X}'_i = \boldsymbol{\Psi}(\mathcal{X}_i) \boldsymbol{\Xi}_i$  to solve for  $\boldsymbol{\Xi}_i = [\xi_i^1, \dots, \xi_i^p]^T$ , which is a vector of coefficients that defines the dynamics, where  $\boldsymbol{\Psi} = [\psi_1, \dots, \psi_p]$  represents a library of basis functions and is applied to  $\mathbf{x}_i$  as

$$\boldsymbol{\Psi}(\mathcal{X}_i) = \begin{array}{c} \xrightarrow{\text{candidate functions of } \mathbf{x}} \\ \left[ \begin{array}{cccc} \psi_1(\mathbf{x}_i(1)) & \psi_2(\mathbf{x}_i(1)) & \dots & \psi_p(\mathbf{x}_i(1)) \\ \psi_1(\mathbf{x}_i(2)) & \psi_2(\mathbf{x}_i(2)) & \dots & \psi_p(\mathbf{x}_i(2)) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(\mathbf{x}_i(T-1)) & \psi_2(\mathbf{x}_i(T-1)) & \dots & \psi_p(\mathbf{x}_i(T-1)) \end{array} \right] \downarrow \text{time} \end{array}$$

where  $p$  is the number of candidate functions in the library  $\boldsymbol{\Psi}$ . (For a detailed description for the basis library, see Supp. Mat. Sec. VIII.) Sparse regression's goal is to determine the dynamics with a small number of functions in  $\boldsymbol{\Psi}$  by finding active coefficients in  $\boldsymbol{\Xi}_i$  (see Supp. Mat. Sec. VII). Consequently, we obtain a predicted model for each node only using the associated node's own data, and we expect to learn similar models for the nodes with similar in-degree  $k_i$ . A distance matrix is obtained by normalized Euclidean distance to classify the predicted models  $d_{ij} = (\sum_{k=1}^p \frac{1}{V_k} |\xi_i^k - \xi_j^k|^2)^{1/2}$ , where  $|\cdot|$  is absolute value,  $V_k$  is the variance of the predicted coefficients of the  $k$ th function in  $\boldsymbol{\Psi}$ . Assume  $\boldsymbol{\Xi}_i$  and  $\boldsymbol{\Xi}_j$  are two predicted models of nodes- $i$  and  $j$ , which are presented as a linear combination of some functions within the library  $\boldsymbol{\Psi}$ . We get smaller  $d_{ij}$  for a similar pair of nodes  $i$  and  $j$ , while  $d_{ij}$  will be large for distinct nodes, such as a low-degree node and a hub. An example computation of  $d_{ij}$  can be found in Supp. Mat. Sec. VIII. The histogram  $P(D)$  is obtained by the row-sum of the distance matrix  $D_i = \sum_j d_{ij}$ , which provides an excellent classification for model similarities in terms of their degrees (Fig. 1 (a)). The low-degree nodes are expected to be located in the highest bin of the histogram since the network has many low-degree nodes.

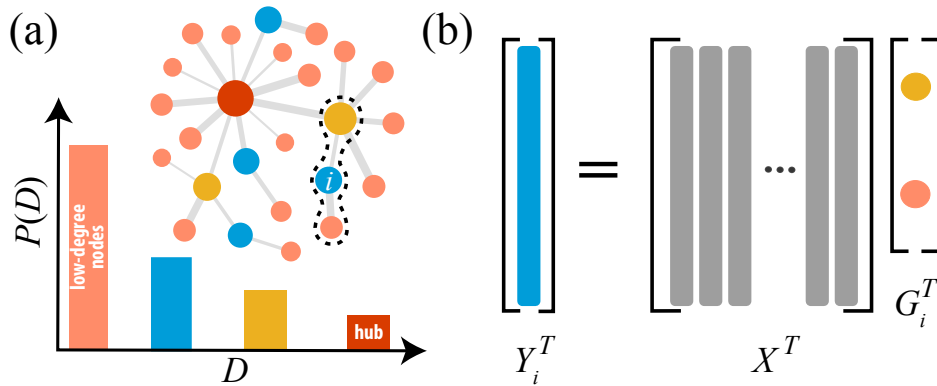


FIG. 1: Illustration of the reconstruction scheme. (a) Performing sparse regression on each observation gives predicted models for each node. Nodes with the same in-degree are reconstructed with the same predicted models, which allow us to classify nodes concerning their in-degrees. As low-degree nodes are abundant and represent the isolated dynamics with a negligible noise, learning  $\mathbf{f}$  is possible. Discarding the local dynamics,  $\mathbf{f}$ , from the hub's data gives the dominant coupling effect on the hub. Therefore, the coupling function  $\mathbf{H}$  can be learned. (b) After learning  $\mathbf{f}$  and  $\mathbf{H}$ , the problem is defined as a linear problem for each node by subtracting the local dynamics, which obtains the remaining interaction effect for each node. Sparse regression on the remaining interaction dynamics of node- $i$  where  $i = 1, \dots, n$  entirely reconstructs the dynamical networks. Nonzero  $\mathbf{G}_i^T$  elements are the incoming connections for  $i$ th node (see Supp. Mat. Sec. II for a step by step scheme).

The models recovered for low-degree nodes are determined as our  $\mathbf{f}$  with a negligible fluctuation  $\kappa_i$ . Contrarily, the distance  $D_i$  is expected to be large for the hub nodes as they are the rarest. Therefore, hubs are located in the lowest bin of the histogram  $P(D)$  (Fig. 1 (a)). Note that the success of the reconstruction depends on the separability of the low-degree nodes and hubs with respect to their degrees, which means the network topology plays an important role here [2], which is further illustrated in Supp. Mat. Sec. II.

We obtain the cumulative coupling effect on the hub by discarding learned isolated dynamics contribution from the identified hub node's data as  $\mathcal{X}'_h - \mathbf{f}(\mathcal{X}_h)$  where  $h$  denotes the hub node. We fit a function to the cumulative coupling effect on the hub and learn the coupling function  $\mathbf{H}$  with a possible linear shift due to the integration constant  $C$  (Supp. Mat. Sec. II). The size of the linear shift can be easily estimated using  $\mathbf{H}(\mathbf{0}) = \mathbf{0}$ . Inferring the interaction function is vital to reveal the network [27], and learning  $\mathbf{H}$  from such reduced dynamics increases the feasibility of our approach. Introducing the Laplacian matrix,  $\mathbf{L}$  with  $L_{ij} = \delta_{ij}k_i - w_{ij}$  where  $\delta_{ij}$  is the Kronecker delta ( $\delta_{ii} = 1$  and  $\delta_{ij} = 0$  if  $i \neq j$ ) and assuming that  $\mathbf{H}$  is a linear function, we can rewrite Eq. (5) in a compact form as

$$\mathbf{X}(t+1) = \mathbf{F}(\mathbf{X}(t)) - (\mathbf{L} \otimes \mathbf{H})(\mathbf{X}(t)), \quad (3)$$

where  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ ,  $\mathbf{F}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_n)]^T$  and  $\otimes$  is the Kronecker product (See Supp. Mat. Sec. I for the derivation in terms of Laplacian matrix and [28]). Defining  $\mathbf{Y}(t) = \mathbf{X}(t+1) - \mathbf{F}(\mathbf{X}(t))$ , Eq. (8) can be written as  $\mathbf{Y} = \mathbf{G}\mathbf{X}$  where  $\mathbf{G} = -(\mathbf{L} \otimes \mathbf{H})$ . Finally, we complete the reconstruction by learning sparse matrix  $\mathbf{G} \in \mathbb{R}^{mn \times mn}$ , by solving the linear equation  $\mathbf{Y}^T = \mathbf{X}^T \mathbf{G}^T$  using sparse regression, namely Least Absolute Shrinkage and Selection Operator (LASSO)[29], as suggested in Ref. [30]. LASSO adds the  $\ell_1$  regularization penalty to the least-squares loss function to find the sparse coefficients (the links), and it is assessed as a compressed sensing approach [31]. Note that the linear equation can also be solved with  $\ell_2$ -norm for long time series, however, as we are interested in short data (in the case of the length of the time series  $T < mn$ ) the compressed sensing approach must be employed [31]. Consequently, we learn the connectivity matrices  $\mathbf{G}$  and  $\mathbf{L}$  as seen in Fig. 1 (b). It is also important to note that learning the equations of all nodes by a single sparse regression without the reduction theorem is only possible for relatively small networks. The library extension, due to network size, causes a statistically correlated data matrix that quickly fails on reconstruction [32, 33]. A detailed discussion can be found in Supp. Mat. Sec. XII.

*Mouse neocortex reconstruction.*— A weighted and directed neural network (987 nodes and 1536 edges), representing a mouse neocortex [34, 35], is considered (Supp. Mat. Sec. IV). To mimic neurons, we used electrically coupled Rulkov maps akin to diffusion as

$$\begin{aligned} u_i(t+1) &= \frac{\beta}{1 + u_i(t)^2} + v_i(t) - \sum_j L_{ij} u_j + \eta_i \\ v_i(t+1) &= v_i(t) - \nu u_i(t) - \sigma \end{aligned}$$

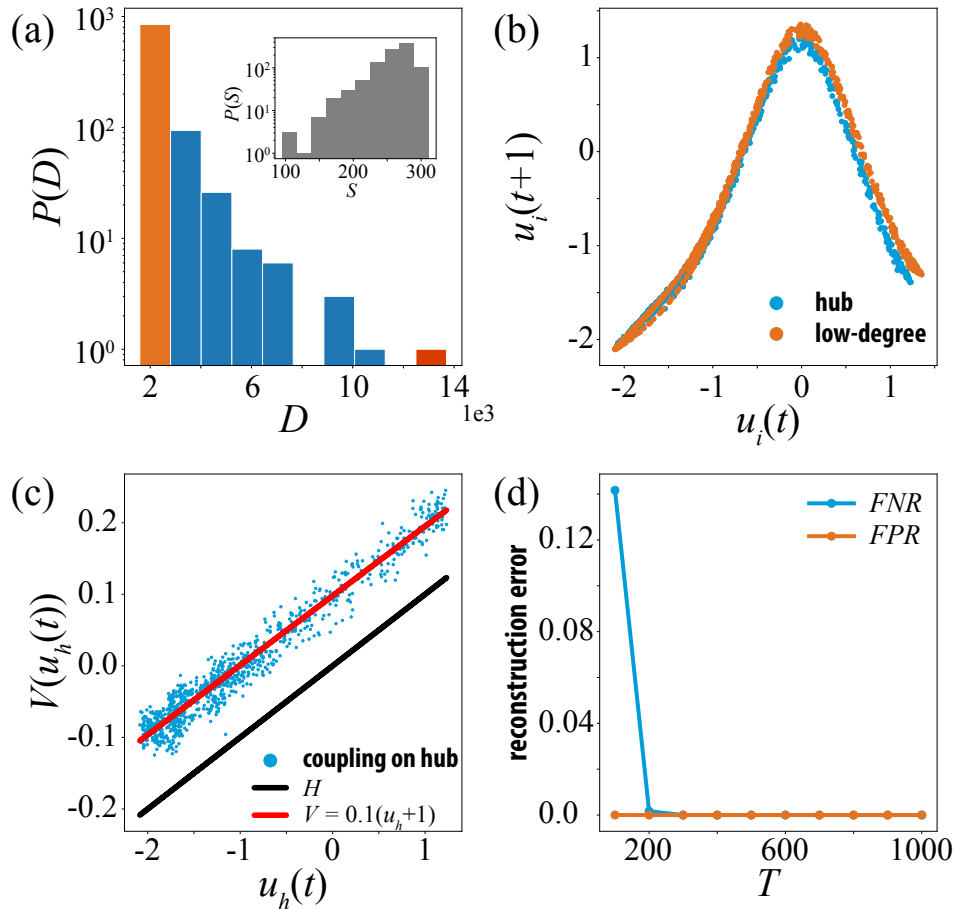


FIG. 2: Reconstruction procedure for weakly electrically coupled Rulkov maps on a real mouse neocortex network. (a) Node-similarity histogram determines the low-degree nodes (orange bar) and the hub (red bar). Inset: Histogram  $P(S)$  presents the correlations between the original time series. It is impossible to infer the connectivity structure from the correlations due to the chaotic nature of the Rulkov maps. (b) The return maps of a low-degree node and the hub are slightly different due to weak coupling effect. (c) The effective coupling,  $V(u)$ , shifts through the horizontal direction due to the integral constant 1. (d)  $FNR$  for different lengths of time series.  $FPR$  are zero for all time series lengths.

where the fast variable  $u_i$  is the membrane potential and the slow variable  $v_i$  is the ion concentration variation [36]. The constant parameters  $\beta = 4.1$  and  $\nu = \sigma = 0.001$  are fixed for chaotic bursting dynamics [37].

*Noise-free case.*— Following the reconstruction scheme, the nodes are classified using the similarity histogram Fig. 2(a) for  $\eta_0 = 0$ , while the pairwise Pearson correlations do not show any information about the degrees (inset in Fig. 2(a)). The difference between return maps of a low-degree node and the hub is illustrated in Fig. 2(b), and the comparison against an isolated node dynamics is given in Supp. Mat. Sec. V-A. Effective coupling  $\mathbf{V}(\mathbf{x})$  is found approximately  $[0.1(u+1), 0]$ , meaning that  $\alpha$  is 0.1 and the linear shift is 1 on  $u$ -variable due to  $\mathbf{H}(\mathbf{0}) = \mathbf{0}$  (Fig. 2(c)). Finally, we learn the network topology by solving the linear equation  $\mathbf{Y} = \mathbf{GX}$  using the learned  $\mathbf{f}$  and  $\tilde{\mathbf{H}}$ . We measure the reconstruction error using the fraction of the false negatives (positives) out of the positives (negatives),  $FNR$  ( $FPR$ ). The  $FNR$  ( $FPR$ ) equals 0 for perfect reconstruction (Supp. Mat. Sec. III). Here, we use the ground truth Laplacian matrix to assess the accuracy of the reconstruction. When the ground truth is not available, the learned model can be evaluated by the cross-validation techniques (Supp. Mat. Sec. XI). The reconstruction error is found to be almost zero for a data length larger than  $T > 200$  (Fig. 2(d)). Furthermore, a systematical evaluation of our approach according to penalty terms and time series lengths is performed. When the number of nodes  $n$  times the dimension of the local dynamics  $m$  exceeds the time series length  $nm > T$ , it corresponds to an underdetermined linear problem. Even for short data ( $T \approx 200$ ), a successful reconstruction is possible with a small penalty term when  $mn = 1974$  (Fig. 3(a)). Note that if the problem is overdetermined  $nm < T$ , then  $\ell_2$ -norm regression can be used for faster computations (Supp. Mat. Sec. IX). Furthermore, we provide reconstruction analysis for Hénon map



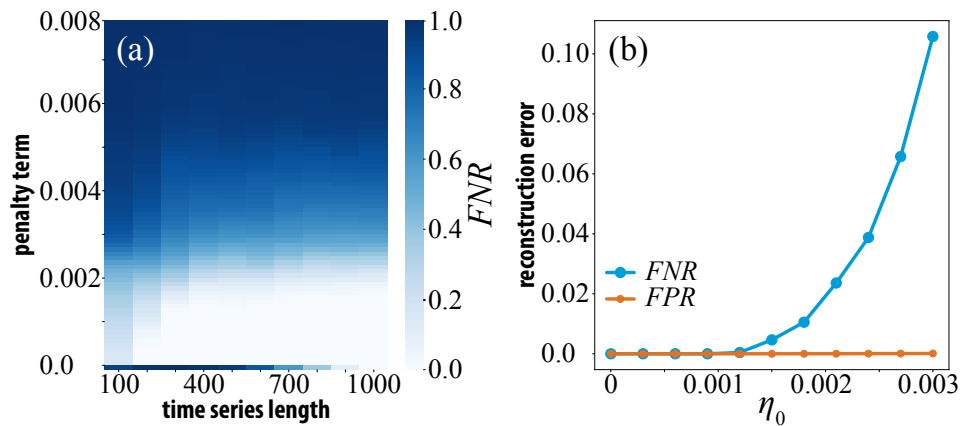


FIG. 3: (a) The reconstruction performance for different time series lengths and a series of penalty terms as  $FNR$ .  $FPR$  always equals 0 for this case. (b) Noise effect on reconstruction performance on real network as  $FNR$  and  $FPR$ .

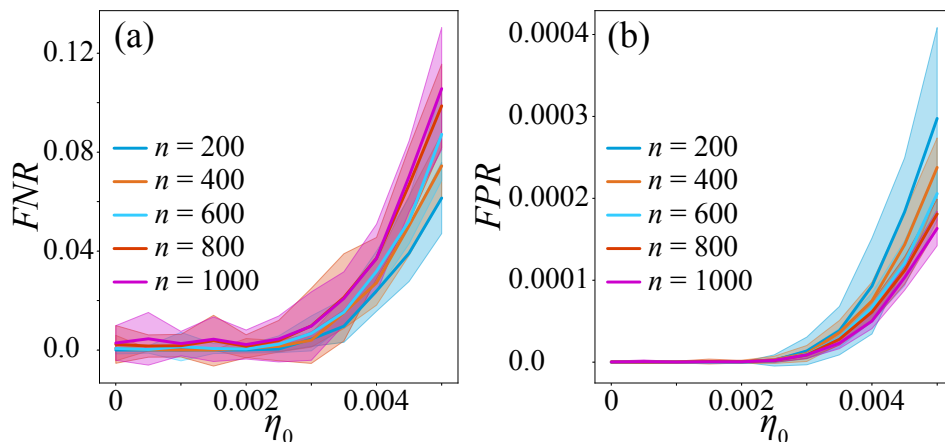


FIG. 4: Average noise effect on reconstruction performance illustrated using (a)  $FNR$  and (b)  $FPR$  for 50 realizations of simulations using random scale-free networks of system sizes  $n = 200, 400, 600, 800$  and  $1000$ . Time series length is fixed as 500 during all the simulations. The shaded regions represent the corresponding standard deviations.

and the Tinkerbell map in Supp. Mat. Sec. V. We also perform our procedure on a macaque monkey visual cortex network [35, 38]. This network is not scale-free; however, assuming the local dynamics and the hub node are known, we reconstructed the network dynamics (see Supp. Mat. Sec. VI).

*Noise effect on reconstruction performance.*— To measure the robustness of our methodology against noise, we systematically perform the reconstruction procedure for various  $\eta_0$  values (Eq. (5)) on synthetic as well as real mouse neocortex networks. The reconstruction approach is robust to small noise intensities for the real-world example (Fig. 3(b)). The average robustness of the reconstruction procedure over 50 different directed and weighted random scale-free networks is given in Fig. 4. The scale-free networks are generated using the algorithm in Ref. [39], and weights are assigned uniformly from the interval  $[0.8, 1.2]$ . The algorithm generates undesired self-loops and multiple edges, so first we remove them. As the system size grows, the reconstruction performance reduces with respect to  $FNR$  for increasing noise intensity  $\eta_0$  (Fig. 4(a)), since the noise becomes more dominant than the weak coupling, which prohibits learning  $\mathbf{H}$  using the coupling effect. As similar to real-world application,  $FPR$  results are also negligibly small for the noise induced reconstruction case (Fig. 4(b)). We also performed experiments by generating denser networks where the reconstruction technique fails, therefore, the network sparsity is crucial for the reconstruction (see Supp. Mat. Sec. X).

*Prediction of emergent behavior.*— Although we recover the true network structure, the reduction theory approximates the isolated dynamics  $\mathbf{f}$  and coupling function  $\mathbf{H}$  with small bounded fluctuations. As the local dynamics is chaotic and we consider possible noise effect, time evolution forecasting of the given system can diverge from the

original system. However, as we recovered the network dynamics with high accuracy for the given data as,

$$\mathbf{X}(t+1) = \mathbf{F}(\mathbf{X}(t)) + \gamma \mathbf{G}\mathbf{X}(t), \quad (4)$$

where  $\gamma$  is the coupling control parameter and initially it is  $\gamma = 1$  for the reconstructed model. Then, it is possible to detect critical coupling strength factor  $\gamma_c$  to predict the emergent behavior of the dynamical network by fully analytical techniques if the reconstructed coupling function is identity matrix [40]. For general coupling functions, master stability function [41] or connection graph method [42] can be performed for the detection (see Supp. Mat. Sec. XIII).

*Conclusions.*— The network is fully recovered by our approach in the setting where  $\mathbf{f}$  is chaotic, the network is scale-free and the coupling is weak. Because of the weak coupling between nodes and the chaotic nature of the local dynamics, the correlation between measured time series decays exponentially; therefore, it is impossible to reconstruct such complex systems by conventional methods. Cutting-edge autonomous statistical learning techniques also fail when the network size is large. The key idea in our procedure is splitting the model equation into parts by reduction theorem and inferring each unknown ( $\mathbf{f}$ ,  $\mathbf{H}$  and  $\mathbf{W}$ ) one by one using sparse recovery. Although the reduction theorem is not established for the general chaotic discrete maps, we showed its validity on various maps. Our approach guarantees the full reconstruction for the noise-free case and small noise intensity even for relatively short time series with no limitations on the network size. However, the quality of reconstruction decreases for increasing noise and the destructive effect of the noise also increases with increasing system size. Finally, obtaining the network dynamics allows one to predict the emergent behavior under parameter changes. There are available regression-based approaches that can learn network topology using short time series [8]; however, it is impossible to detect the critical transitions with only the connectivity. The ability to detect such transitions is crucial for applications such as a transition to collective behavior in the brain network, which can lead to undesired implications. Thus, it is desirable to put forward precautionary norms to avert potential disasters.

*Data and code availability.*—The data we used in this study can be regenerated by running the code which is publicly available on GitHub [43].

We are indebted to Tiago Pereira, Matteo Tanzi, Sajjad Bakrani, Arash Rezaeinazhad, Thomas Peron and Jeroen Lamb for enlightening discussions. This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. 118C236. D.E. acknowledges support from the BAGEP Award of the Science Academy.

- 
- [1] K. Schindler, H. Leung, C. E. Elger, and K. Lehnertz, *Brain* **130**, e65 (2007).
  - [2] D. Eroglu, M. Tanzi, S. van Strien, and T. Pereira, *Phys. Rev. X* **10**, 021047 (2020).
  - [3] T.-T. Gao and G. Yan, *Nat. Comput. Sci* **2**, 160 (2022).
  - [4] W.-X. Wang, Y.-C. Lai, and C. Grebogi, *Phys. Rep.* **644**, 1 (2016).
  - [5] M. Nitzan, J. Casadiego, and M. Timme, *Sci. Adv.* **3**, e1600396 (2017).
  - [6] T. Stankovski, T. Pereira, P. V. E. McClintock, and A. Stefanovska, *Rev. Mod. Phys.* **89**, 045001 (2017).
  - [7] M. Timme and J. Casadiego, *J. Phys. A* **47**, 343001 (2014).
  - [8] J. Casadiego, M. Nitzan, S. Hallerberg, and M. Timme, *Nat. Commun.* **8**, 1-10 (2017).
  - [9] T. S. Gardner, D. D. Bernardo, D. Lorenz, and J. J. Collins, *Science* **301**, 102 (2003).
  - [10] M. Timme, *Phys. Rev. Lett.* **98**, 224101 (2007).
  - [11] S. G. Shandilya and M. Timme, *New J. Phys.* **13**, 013004 (2011).
  - [12] D. Yu, M. Righero, and L. Kocarev, *Phys. Rev. Lett.* **97**, 188701 (2006).
  - [13] J. Ren, W. X. Wang, B. Li, and Y. C. Lai, *Phys. Rev. Lett.* **104**, 058701 (2010).
  - [14] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and M. A. F. Harrison, *Europhys. Lett.* **94**, 48006 (2011).
  - [15] W.-X. Wang, Y. C. Lai, C. Grebogi, and J. Ye, *Phys. Rev. X* **1**, 021021 (2011).
  - [16] H. Korn and P. Faure, *C. R. Biol* **326**, 787 (2003).
  - [17] A. J. Preyer and R. J. Butera, *Phys. Rev. Lett.* **95**, 138103 (2005).
  - [18] G. Werner, *Frontiers in Physiology* **1**, 15 (2010).
  - [19] B. Barzel and A. L. Barabási, *Nat. Phys.* **9**, 673 (2013).
  - [20] E. M. Izhikevich, *Dynamical systems in neuroscience* (MIT press, 2007).
  - [21] F. Dörfler, M. Chertkov, and F. Bullo, *Proc. Natl. Acad. Sci. U.S.A.* **110**, 2005 (2013).
  - [22] A. E. Motter, S. A. Myers, M. Anghel, and T. Nishikawa, *Nat. Phys.* **9**, 191 (2013).
  - [23] S. Watanabe and S. H. Strogatz, *Phys. D* **74**, 197 (1994).
  - [24] A. T. Winfree, *The Geometry of Biological Time*, vol. 12 (Springer Science & Business Media, 2001).
  - [25] T. Pereira, S. van Strien, and M. Tanzi, *J. Eur. Math. Soc.* **22**, 2183 (2020).
  - [26] S. L. Brunton, J. L. Proctor, J. N. Kutz, and W. Bialek, *Proc. Natl. Acad. Sci. U.S.A.* **113**, 3932 (2016).
  - [27] Y. Y. Liu and A. L. Barabási, *Rev. Mod. Phys.* **88**, 035006 (2016).
  - [28] T. Pereira, D. Eroglu, G. B. Bagci, U. Tirnakli, and H. J. Jensen, *Phys. Rev. Lett.* **110**, 234103 (2013).

- [29] R. Tibshirani, *J. R. Stat. Soc., B: Stat. Methodol* **58**, 267 (1996).
- [30] X. Han, Z. Shen, W. X. Wang, and Z. Di, *Phys. Rev. Lett.* **114**, 028701 (2015).
- [31] E. J. Candes, J. K. Romberg, and T. Tao, *Commun. Pure Appl. Math.* **59**, 1207 (2006).
- [32] M. Novaes, E. R. dos Santos, and T. Pereira, *Phys. D* **424**, 132895 (2021).
- [33] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, *Journal of Open Source Software* **5**, 2104 (2020).
- [34] N. Kasthuri, K. J. Hayworth, D. R. Berger, R. L. Schalek, J. A. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, T. R. Jones, *et al.*, *Cell* **162**, 648 (2015).
- [35] J. T. Vogelstein, E. Perlman, B. Falk, A. Baden, W. G. Roncal, V. Chandrashekhar, F. Collman, S. Seshamani, J. L. Patsolic, K. Lillaney, *et al.*, *Nat. Methods* **15**, 846 (2018).
- [36] Q. Wang, G. Chen, and M. Perc, *PLoS One* **6**, e15851 (2011).
- [37] N. F. Rulkov, *Phys. Rev. E* **65**, 041922 (2002).
- [38] N. T. Markov, J. Vezoli, P. Chameau, A. Falchier, R. Quilodran, C. Huissoud, C. Lamy, P. Misery, P. Giroud, S. Ullman, *et al.*, *J. Comp. Neurol.* **522**, 225 (2014).
- [39] B. Bollobás, C. Borgs, J. Chayes, and O. Riordan, in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Society for Industrial and Applied Mathematics, 2003), SODA '03, p. 132–139.
- [40] D. Eroglu, J. S. Lamb, and T. Pereira, *Contemp. Phys.* **58**, 207 (2017).
- [41] L. M. Pecora and T. L. Carroll, *Phys. Rev. Lett.* **80**, 2109 (1998).
- [42] V. N. Belykh, I. V. Belykh, and M. Hasler, *Phys. D* **195**, 159 (2004).
- [43] <https://github.com/lffrrnt/ReconstructingNetworkDynamics/>.
- [44] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and C. Grebogi, *Phys. Rev. Lett.* **106**, 154101 (2011).
- [45] P. Mehta, M. Bukov, C. H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, *Phys. Rep.* **810**, 1 (2019).
- [46] D. Donoho, *IEEE Trans. Inf. Theory* **52**, 1289 (2006).
- [47] S. L. Brunton, J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2019).
- [48] L. Huang, Q. Chen, Y.-C. Lai, and L. M. Pecora, *Phys. Rev. E* **80**, 036204 (2009).
- [49] P. Schultz, T. Peron, D. Eroglu, T. Stemler, G. M. Ramírez Ávila, F. A. Rodrigues, and J. Kurths, *Phys. Rev. E* **93**, 062211 (2016).

# Reconstructing network dynamics of coupled discrete chaotic units from data:

## Supplemental Material

Irem Topal and Deniz Eroglu

Faculty of Engineering and Natural Sciences, Kadir Has University, 34083 Istanbul, Turkey

### Contents

<b>I. Connectivity matrix in terms of Laplacian</b>	2
<b>II. Network Reconstruction Scheme on a toy model</b>	2
<b>III. Assessing the performance</b>	4
<b>IV. Mouse neo-cortex network</b>	6
<b>V. Applications on mouse neo-cortex network</b>	6
A. Rulkov map	6
B. Hénon map	7
1. $u$ -component coupling	7
2. $u$ -component coupling with sinusoidal function	8
C. Tinkerbell Map	9
<b>VI. Macaque monkey visual cortex network</b>	10
<b>VII. Sparse optimization</b>	10
<b>VIII. Library of basis functions</b>	11
<b>IX. Comparison between sparse regression methods</b>	12
<b>X. Effect of network sparsity on reconstruction performance</b>	14
<b>XI. Cross validation of inferred Laplacian matrices</b>	14
<b>XII. Why is the reduction theorem crucial to reconstruct large networks?</b>	15
<b>XIII. General Diffusive Coupling and Master stability function</b>	16
<b>References</b>	17

## I. CONNECTIVITY MATRIX IN TERMS OF LAPLACIAN

Network dynamics with diffusive interaction is given by,

$$\mathbf{x}_i(t+1) = \mathbf{f}(\mathbf{x}_i(t)) + \sum_{j=1}^n w_{ij} [\mathbf{H}(\mathbf{x}_j) - \mathbf{H}(\mathbf{x}_i)] \quad (5)$$

where  $\mathbf{x} \in \mathbb{R}^m$ ,  $\mathbf{f}: \mathbb{R}^m \rightarrow \mathbb{R}^m$  is chaotic dynamics of isolated nodes and  $\mathbf{H}$  is a diffusive coupling function [1].  $\mathbf{W} = [w_{ij}] \in \mathbb{R}^{n \times n}$  is a weighted and directed adjacency matrix. Diffusive nature of the interaction allows represent the coupling in terms of the Laplacian matrix.

$$\begin{aligned} \sum_{j=1}^n w_{ij} [\mathbf{H}(\mathbf{x}_j) - \mathbf{H}(\mathbf{x}_i)] &= \sum_{j=1}^n w_{ij} \mathbf{H}(\mathbf{x}_j) - \mathbf{H}(\mathbf{x}_i) \sum_{j=1}^n w_{ij} \\ &= \sum_{j=1}^n w_{ij} \mathbf{H}(\mathbf{x}_j) - k_i \mathbf{H}(\mathbf{x}_i) \\ &= \sum_{j=1}^n (w_{ij} - \delta_{ij} k_i) \mathbf{H}(\mathbf{x}_j) \end{aligned} \quad (6)$$

where  $k_i = \sum_j w_{ij}$  is the incoming degree of node  $i$  and  $\delta_{ij}$  is the Kronecker delta. By introducing the Laplacian matrix,  $\mathbf{L}$  with  $L_{ij} = \delta_{ij} k_i - w_{ij}$  we obtain,

$$\mathbf{x}_i(t+1) = \mathbf{f}(\mathbf{x}_i(t)) - \sum_{j=1}^n L_{ij} \mathbf{H}(\mathbf{x}_j) \quad (7)$$

Then we can rewrite Eq. (7) in a compact form as,

$$\mathbf{X}(t+1) = \mathbf{F}(\mathbf{X}(t)) - (\mathbf{L} \otimes \mathbf{H})(\mathbf{X}(t)), \quad (8)$$

where  $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T$ ,  $\mathbf{F}(\mathbf{X}) = [\mathbf{f}(\mathbf{x}_1), \dots, \mathbf{f}(\mathbf{x}_n)]^T$  and  $\otimes$  is the Kronecker product [2].

## II. NETWORK RECONSTRUCTION SCHEME ON A TOY MODEL

We present a step-by-step reconstruction algorithm using weakly coupled chaotic maps on a directed and weighted network of size  $n = 20$  Fig. 5(a). The network is heterogeneous; the low-degree nodes are abundant, and the hub is the rarest, as illustrated in the network's in-degree distribution Fig. 5(b). The network has Rulkov maps interacting with each other through their  $u$ -components diffusively:

$$\begin{aligned} u_i(t+1) &= \frac{\beta}{1 + u_i(t)^2} + v_i(t) - \sum_{j=1}^n L_{ij} u_j(t) \\ v_i(t+1) &= u_i(t) - \nu u_i(t) - \sigma \end{aligned} \quad (9)$$

where the constant parameters  $\beta = 4.1$  and  $\nu = \sigma = 0.001$  are fixed for chaotic bursting dynamics. We first simulate the system 15000 time steps and discard the first 14000 steps as a transient. We should note that the local dynamics (Rulkov map), the coupling function ( $u$ -coupling) and the connectivity matrix ( $L_{ij}$ ) are all unknowns during the procedure; however, we present every piece of information in this toy model to show the method rigorously.

Fig. 5(c) shows the 2-dimensional data coming from the hub and one of the low degree nodes. We present the return maps of the same nodes in Fig. 5(d), the dispersion of the hub node due to the dominant coupling effect can be seen. On the other hand, we compute the pairwise Pearson correlation coefficients  $s_{ij}$  between the time series, see Fig. 5(e). Then we obtain the histogram  $P(S)$ , Fig. 5(f), by the row-sum of the correlation matrix in Fig. 5(e). Given the chaotic nature of the Rulkov maps, correlations between time series does not give any information about the network connectivity.

The reconstruction procedure is given by the following steps:

### 1. Classification of the nodes.

The separability of low-degree nodes and hubs is crucial since the reconstruction procedure is not applicable if we cannot classify which nodes are low-degree or hubs at the beginning of the reconstruction recipe. The classification leads us to learn the isolated dynamics  $\mathbf{f}$  and the coupling function  $\mathbf{H}$  as follows: we, first, assume that the coupling is *weak*, meaning that if we classify the low-degree nodes, we can consider their dynamics as local dynamics  $\mathbf{f}$  with some small fluctuations. If we classify which node is the hub and  $\mathbf{f}$ , then we can learn the coupling function  $\mathbf{H}$  by discarding  $\mathbf{f}(\mathbf{x}_h(t))$  from the hub's time series  $\mathbf{x}_h(t+1)$  using the cumulative effect of interactions. Therefore, our first aim is to classify the nodes regarding their degrees.

For the classification, we first learn a governing equation for each time series using sparse regression methods (Supp. Mat. Sec. VII) with a basis composed of candidate functions (Supp. Mat. Sec. VIII). After obtaining the  $n$  predicted models, we measure the difference between the models using normalized Euclidean distance between the coefficients of predicted models,  $d_{ij}$ . A distance matrix  $D = [d_{ij}]$  summarizes the mismatches between the predicted models. Fig. 5(g). The row-sum of the distance matrix  $D_i = \sum_j d_{ij}$  gives a histogram  $P(D)$ , and each bin of the histogram contains nodes with a similar degree (Fig. 5(h)).

It is also important to note that weak coupling is also crucial to avoid possible synchronization in the network dynamics. If the system is synchronized, the time series will be identical; in other words, all the predicted models will be identical. This will make the classification impossible. Therefore, the weak coupling is quite crucial for this algorithm to work as desired.

**2. Learning the local dynamics.** To mimic brain network dynamics, we assume that the connectivity matrix  $\mathbf{W} = [w_{ij}]$  represents a scale-free network whose degree distribution follows a power law; in other words, most of the nodes have small degrees  $k \sim n^\epsilon$ , and some nodes are hubs with degrees  $k \sim n^{\frac{1}{2}+\epsilon}$  where  $\epsilon$  is an arbitrarily small number. Therefore, the highest bin of the histogram (the most abundant frequencies) contains the low-degree nodes' models since there are many low-degree nodes (the orange bar in Fig. 5 (h)). As it is likely to have nodes having incoming links,  $k_i > 0$ , in the set of low-degree nodes, the models can contain some small fluctuations. Thus, we average the predicted coefficients of the low-degree nodes' models to infer the local dynamics as accurately as possible.

**3. Learning the coupling function.** Using the scale-free network topology assumption as in the previous section, we understand that the lowest bin of the histogram (the rarest frequencies) contains the most different models, which belong to the hubs (the blue bar in Fig. 5(h)). Then discarding the local dynamics of the hub ( $\mathbf{f}(\mathbf{x}_h(t))$ ) from the hub measurement ( $\mathbf{x}_h(t+1)$ ) gives the dominant coupling effect, which we aim to learn. Fig. 5(i) presents the remaining coupling effect. We fit a function to the data and learn an approximate coupling function plus an integration constant arising from the reduction theorem. To be more precise with the integration constant, for instance, if the diffusive coupling function is given as  $h(x, y) = \phi(y) - \phi(x)$  then the effective coupling  $V$  for the expending maps is  $V = \int h(x, y) d\mu(y) = -\phi(x) + C$  where  $C$  is the integration constant, which we call a *possible linear shift* for the numerically recovered effective coupling  $V$ . This is easy to estimate from the fitted coupling (Fig. 5 (i)) since we assume that  $\phi(0) = 0$ . Therefore, the appearing linear shift is the integration constant  $C$ , and its value can vary for different dynamical networks. Furthermore, the effective coupling is an approximation, which becomes exact when the network model is considered as a directed star graph with the link directions from low-degree nodes to the hub at the thermodynamical limits as  $n \rightarrow \infty$ . We have an approximated effective coupling for tree-like graphs, as scale-free graphs, under our assumptions related to node degrees (mentioned above). The size of the bounded fluctuation term  $\kappa_i$  in the reduction (Eq. (2) in the main text) can be considered as the measure of approximation.

Here, the weak coupling and chaotic dynamics assumptions take an important role again for more general networks to preserve the statistical behavior of the nodes' dynamics. Although using chaotic oscillators helps to keep the state distribution of the states stable, this property can be destroyed by a large coupling. Therefore, if the network is denser, then the coupling constant should be scaled to a smaller coupling strength for our reduction theorem to work. For instance, if the hubs are rare in a scale-free network, the coupling strength must be scaled by  $n^{1/2}$ . If all the nodes are hubs, in other words, if the network is all-to-all connected, then the coupling strength must be scaled by  $n$ . In our work, we scaled the weights in  $\mathbf{W}$  with  $n^{1/2}$ , as we considered scale-free networks (containing many low-degree nodes and rare hubs) since the work focuses on reconstructing brain networks. It is also important to remind that, for the node classification step, the synchronization of the nodes disallows the reconstruction since the machine learning-based methods cannot distinguish the data source between synchronized oscillators. Consequently, the chaotic oscillators also play an important role here since if we have periodic and identical oscillator networks, any positive coupling strength can synchronize the model, which is undesired for the reconstruction part. Therefore, the weak coupling strength and the chaotic dynamical regime are crucial assumptions for the theory and also for the numerical steps to learn the coupling function  $\mathbf{H}$ .

4. **Learning the connectivity matrix.** After learning the functions  $\mathbf{f}$  and  $\mathbf{H}$ , we define  $\mathbf{Y} = \mathbf{X}(t+1) - \mathbf{F}(\mathbf{X}(t))$ , so Eq. (8) can be written as  $\mathbf{Y} = \mathbf{G}\mathbf{X}$  where  $\mathbf{G} = -(\mathbf{L} \otimes \mathbf{H})$ . Therefore, learning the links becomes a regression problem by solving the linear equation  $\mathbf{Y}^T = \mathbf{X}^T \mathbf{G}^T$ . Here we employ a compressed sensing approach using  $\ell_1$ -norm, called LASSO, to solve the equation since we are interested in reconstructing large network dynamics from short data. In other words, we aim to solve the problem when the length of the time series  $< m \times n$  where  $m$  is the dimension of the local dynamics and  $n$  is the system size. The problems under this condition are called underdetermined regression problems. Finally, we identify the exact Laplacian matrix of the network by solving the linear equation (Fig. 1 (j)).

Note that, in the previous steps of the procedure, the potential interaction functions of two nodes (or higher-order interactions if the hypernetworks are interested) are not involved in the basis library. Meaning that the size of the library is not grown exponentially. Therefore, the algorithm does not require a longer time series to reconstruct the network dynamics as the network size,  $n$ , is increasing. In this last step, we also only sparsely solve a linear equation where  $\mathbf{G} \in \mathbb{R}^{mn \times mn}$ , which is a square matrix. The discussion on the relation between the length of the time series and the sparsity can be found in Supp. Mat. Sec. XII.

### III. ASSESSING THE PERFORMANCE

We use two standard indices to assess the reconstruction performance of our procedure: the fraction of the false negatives out of the positives ( $FNR$ ) and the fraction of false positives out of the negatives ( $FPR$ ). These metrics correspond to two types of errors in network reconstruction: missing a link where a link indeed exists (false negatives) or assigning a link between two nodes when they are not connected (false positives):

$$FNR = \frac{FN}{P} \text{ and } FPR = \frac{FP}{N}$$

where  $FN$  evaluates errors on the non-zero terms ( $P$ ) of the correct matrix (underestimation) and  $FP$  evaluates errors of the zero terms ( $N$ ) (overestimation). We count  $FN$  and  $FP$  links by  $\sum_{i,j} \Theta(|L_{ij} - \hat{L}_{ij}| > \epsilon)$  to assess the accuracy of the link strengths where  $\hat{L}_{ij}$  is the predicted Laplacian and  $\epsilon$  is a tolerance value set as 10 times smaller than the smallest nonzero value in  $\mathbf{L}$ . We use a fixed tolerance value as  $\epsilon = 0.0001$  for all analyses.

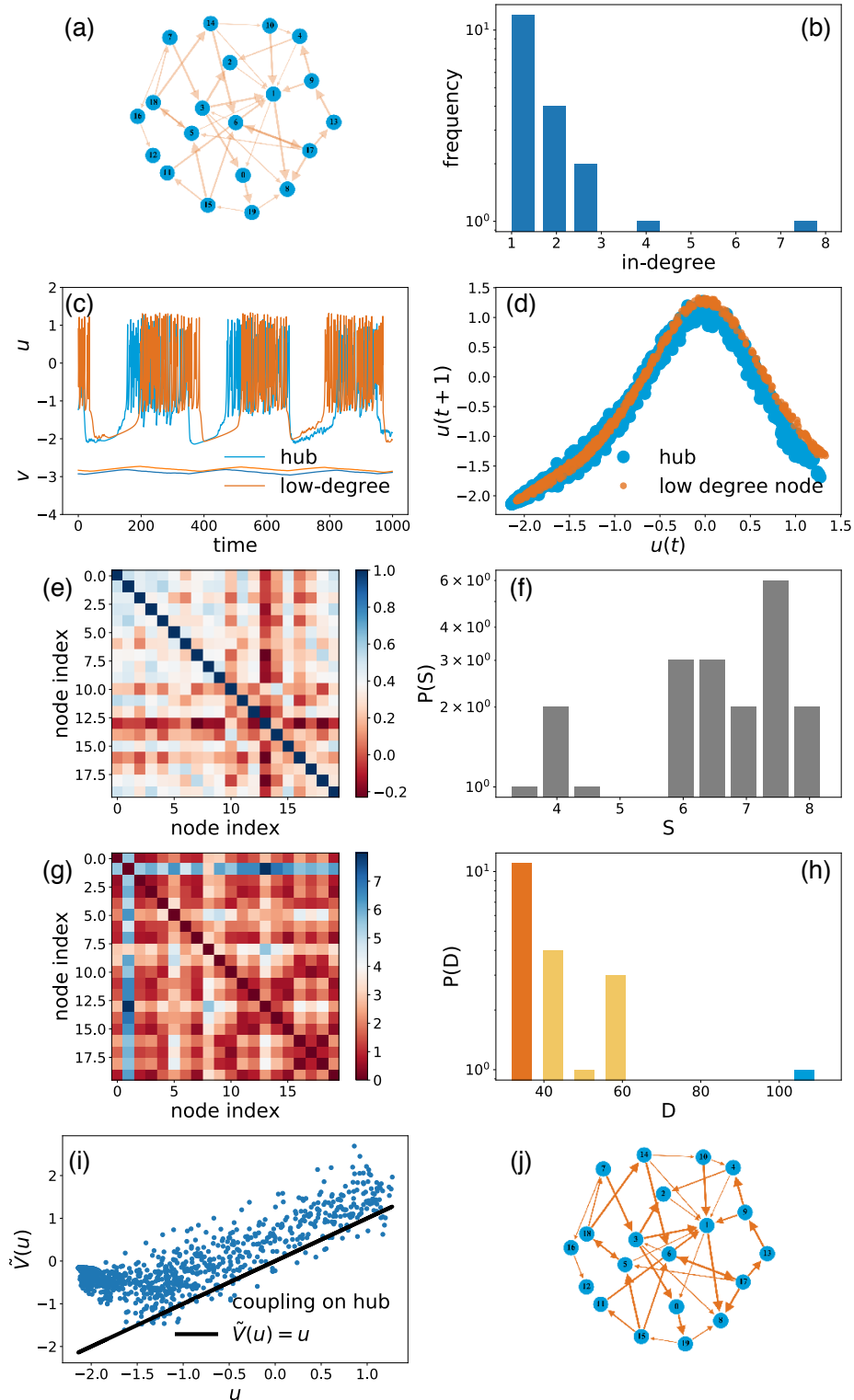


FIG. 5: (a) A directed and weighted network. As the network is unknown initially, its links were illustrated indistinctly. (b) Corresponding in-degree distribution of the network. (c) Time series of a low-degree node and the hub. (d) The return maps of these time series, the hub's data is dispersed according to the low-degree node due to the coupling effect accumulated on the hub. (e) Pairwise Pearson correlation matrix of the time series, which shows no similarity between, even connected, nodes dynamics due to the chaos. (f) The histogram obtained by the row-sum of the time series similarity matrix (subplot (e)) does not reflect any information about the network's original degree distribution; therefore, it is impossible to identify a low-degree node or the hub. (g) Pairwise distances between predicted models were obtained in the first step of our reconstruction procedure. Nodes with similar in-degrees show high similarity. (h) The histogram of the row-sum of the distance matrix (subplot (g)), which does also not contain the exact in-degree distribution of the network (subplot (b)), however, allows for the low-degree and hub node identification. The abundant models (the highest bin in (h)) represent the low-degree nodes, and these models represent the local dynamics, **f**. The most distinct model (the lowest bin in (h)) belongs to the hub node. We learn the coupling function by filtering the local dynamics from the hub data. (i) The remaining coupling effect on the hub is plotted, and the reduction theorem recovers the coupling function. (j) After recovering the local dynamics,



#### IV. MOUSE NEO-CORTEX NETWORK

We choose a neuronal connectivity consisting 1029 nodes representing a small volume of a young adult mouse neocortex [3]. The network is directed and contains multi-edges between some nodes. We considered the number of these multi edges between nodes as weights. Furthermore, the network contains some disconnected parts, we removed them and kept only the giant component. The resulting weighted and directed network has  $n = 987$  nodes and  $m = 1536$  links, which is illustrated in Fig. 6 (a) with its in-degree distribution in Fig. 6 (b). We downscale entries of the connectivity matrix by a small constant 0.1, and then normalize by in-degree of hub,  $k_h$ , so that the network is not synchronized.

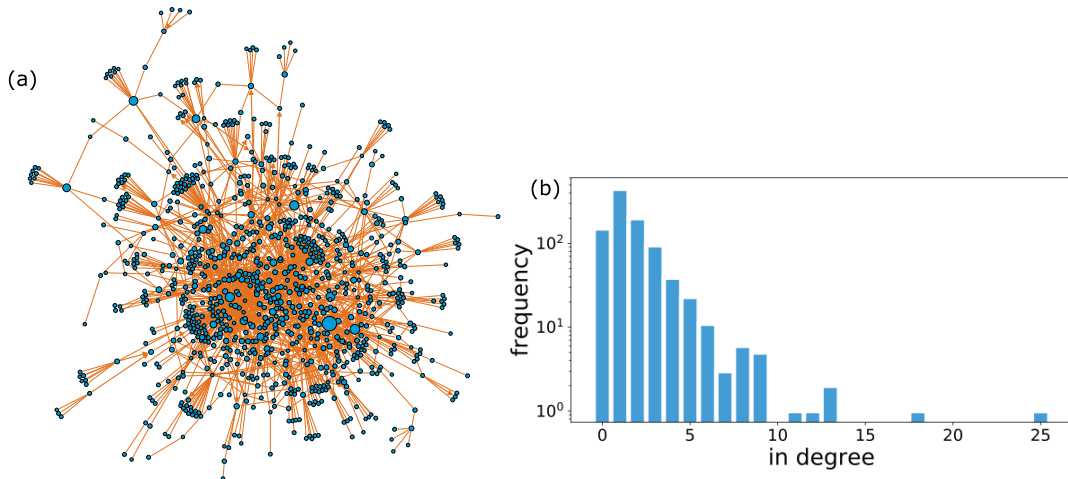


FIG. 6: (a) Pre-processed directed and weighted mouse neo-cortex network. The sizes of the nodes denote their in-coming degrees and the thickness of the arrows denote their weights. (b) In-degree distribution of the associated real network.

#### V. APPLICATIONS ON MOUSE NEO-CORTEX NETWORK

This section presents the results of the reconstruction procedure on various dynamical systems, namely Rulkov map, Hénon map and Tinkerbell map. For all the systems, we use mouse neo-cortex network which described in Section IV.

##### A. Rulkov map

We have already presented Rulkov map results in the main text. In our reconstruction scheme, we assume that a low-degree node can be taken as an isolated node to estimate local dynamics  $\mathbf{f}$ . Here, we compare the return maps of a low-degree node and a hub node's dynamics with an isolated Rulkov map under the effect of weak coupling (Fig 7).

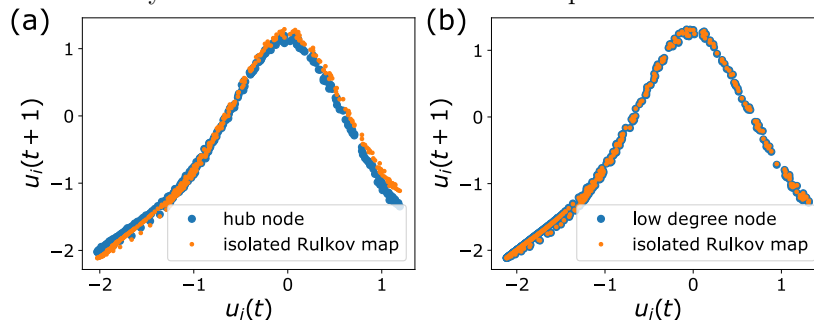


FIG. 7: a) The return map of the hub's data and isolated Rulkov map to show the weak coupling effect. b) The return map of one of the low-degree nodes shows that these nodes oscillate almost with the isolated local dynamics.

## B. Hénon map

### 1. $u$ -component coupling

We use Hénon map as  $\mathbf{f}$  in Eq. (7), a discrete-time dynamical system exhibit chaotic behavior:

$$\begin{aligned} u_i(t+1) &= 1 - \alpha u_i(t)^2 + v_i(t) - \sum_{j=1}^n L_{ij} u_j(t) \\ v_i(t+1) &= \beta u_i(t) \end{aligned} \quad (10)$$

where  $\alpha = 1.4$  and  $\beta = 0.3$  are fixed during the simulations. Hénon maps are coupled through their  $u$ -components weakly. Initial positions are taken from the interval  $[0, 0.1]$  randomly uniformly for each map. We iterate over 11000 steps and discard the first 10000 steps as a transient. Our procedure reveals the local dynamics as *Hénon map*, interaction function as *diffusive coupling* for  $u$ -component, and finally, the correct Laplacian matrix from time series data.

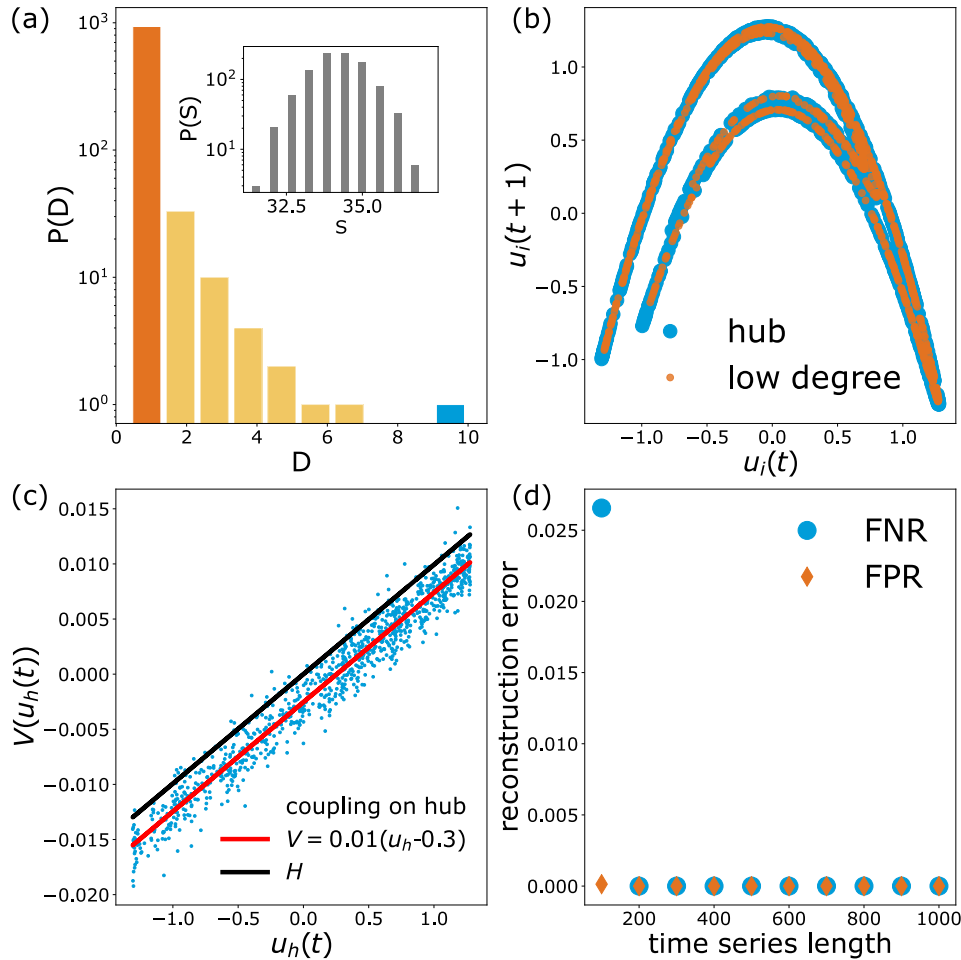


FIG. 8: Reconstruction procedure for weakly  $u$ -coupled Hénon maps on a real network. First, we learn a model for each node's time series data to classify them. (a) The histogram  $P(D)$  based on the pairwise Euclidean distance matrix. While the highest bin of the histogram gives us  $\mathbf{f}$ , the lowest bin identifies the hub. The inset histogram emphasizes that the pairwise correlations of the time series observations do not provide any information about the network connectivity due to the chaotic dynamics. (b) The return maps of a low degree node and hub show the coupling effect slightly due to weak coupling. In (c), we plot the dominant coupling effect on the hub to see the model of  $\mathbf{H}$ . (d) shows  $FNR$  and  $FPR$  for different lengths of time series. 200 data points are enough for full reconstruction of the network dynamics with 987 nodes.

2.  $u$ -component coupling with sinusoidal function

We use the same  $\mathbf{f}$  but different coupling function  $\mathbf{H}$  in this example:

$$\begin{aligned} u_i(t+1) &= 1 - \alpha u_i(t)^2 + v_i(t) - \sum_{j=1}^n L_{ij} \sin(2\pi u_j(t)) \\ v_i(t+1) &= \beta u_i(t) \end{aligned} \quad (11)$$

Initial positions are taken from the interval  $[0, 0.01]$  randomly uniformly for each map. We iterate over 12000 steps and discard the first 10000 steps as a transient. Fig. 9 shows the reconstruction results of sine-coupled Hénon maps. We simply added trigonometric functions to the candidate function library in the first step of our procedure and fully reconstructed the network connections by learning the local dynamics as *Hénon map* and the coupling function as *sine function*.

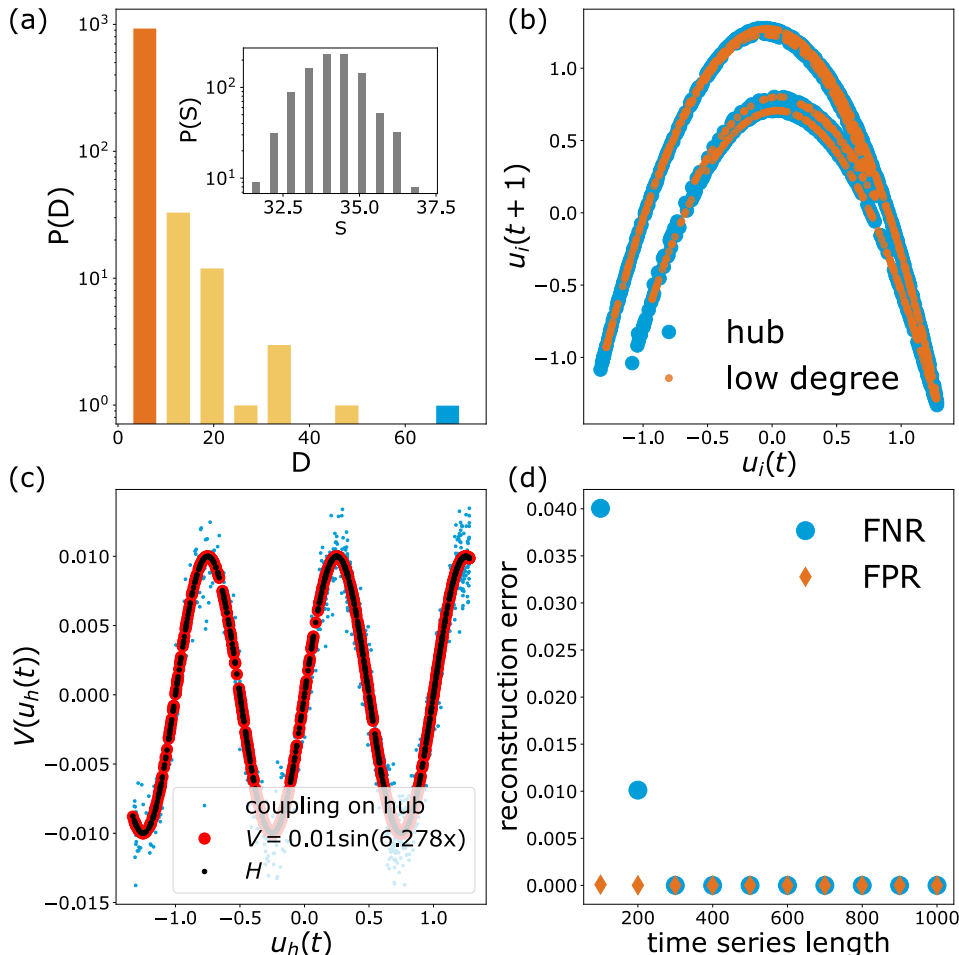


FIG. 9: Reconstruction procedure for weakly sin  $u$ -coupled Hénon maps on a real network. First, we learn a model for each node's time series data to classify them. (a) The histogram  $P(D)$  based on the pairwise Euclidean distance matrix. While the highest bin of the histogram gives us  $\mathbf{f}$ , the lowest bin identifies the hub. The inset histogram emphasizes that the pairwise correlations of the time series observations do not provide any information about the network connectivity due to the chaotic dynamics. (b) The return maps of a low degree node and hub show the coupling effect slightly due to weak coupling. In (c), we plot the the dominant coupling effect on the hub to see the model of  $\mathbf{H}$ . (d) shows  $FNR$  and  $FPR$  for different lengths of time series. 300 data points are enough for full reconstruction of the network dynamics with 987 nodes.

### C. Tinkerbell Map

Tinkerbell map coupled through their  $u$ -component diffusively is given by:

$$\begin{aligned} u_i(t+1) &= u_i(t)^2 - v_i(t)^2 + au_i(t) + bv_i(t) - \sum_{j=1}^n L_{ij}u_j(t) \\ v_i(t+1) &= 2u_i(t)v_i(t) + cu_i(t) + dv_i(t) \end{aligned} \quad (12)$$

where  $a = 0.9, b = -0.6013, c = 2.0$  and  $d = 0.5$  are fixed during the simulations to ensure a fully chaotic regime. Data is generated from  $u$ -coupled Tinkerbell maps by iterating over 11000 steps and dropping the first 10000 time steps as a transient. We successfully reveal the local dynamics of the Tinkerbell map, interaction function and finally the Laplacian matrix from time series.

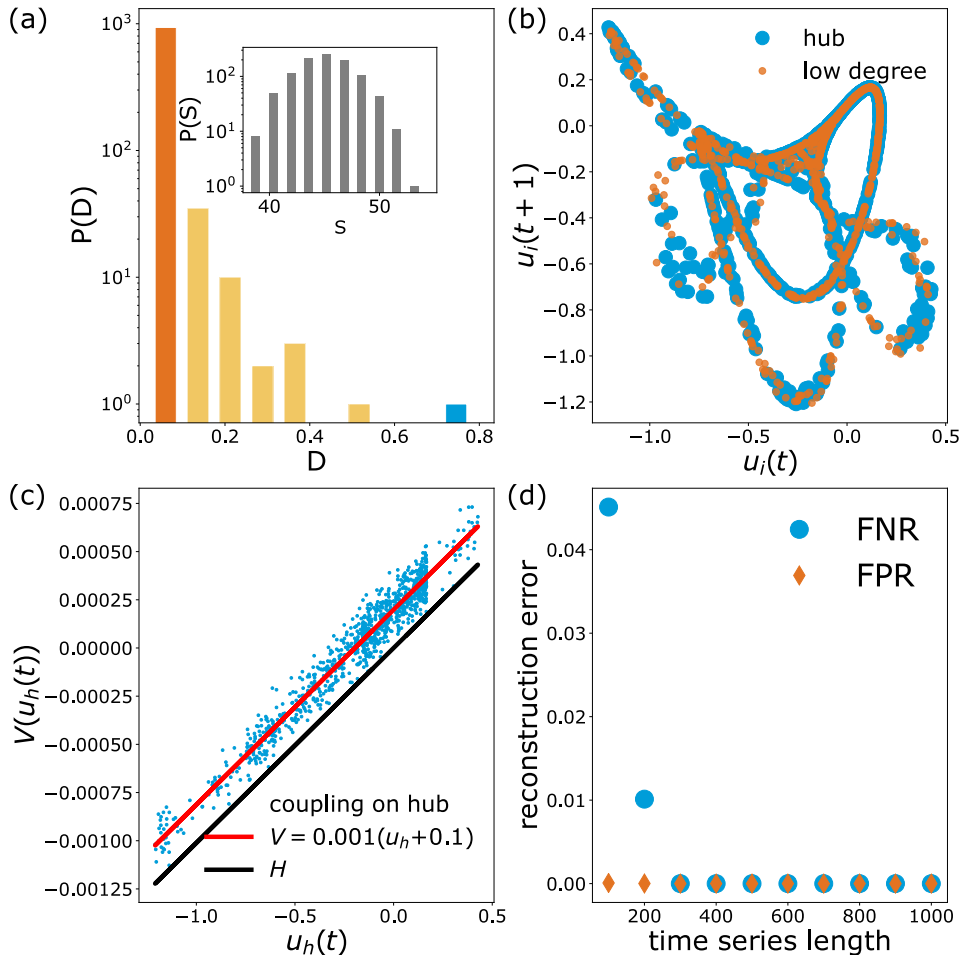


FIG. 10: Reconstruction procedure for weakly  $u$ -coupled Tinkerbell maps on a real network. First, we learn a model for each node's time series data to classify them. (a) The histogram  $P(D)$  based on the pairwise Euclidean distance matrix. While the highest bin of the histogram gives us  $\mathbf{f}$ , the lowest bin identifies the hub. The inset histogram emphasizes that the pairwise correlations of the time series observations do not provide any information about the network connectivity due to the chaotic dynamics. (b) The return maps of a low degree node and hub show the coupling effect slightly due to weak coupling. In (c), we plot the dominant coupling effect on the hub to see the model of  $\mathbf{H}$ . (d) shows  $FNR$  and  $FPR$  for different lengths of time series. 300 data points are enough for full reconstruction of the network dynamics with 987 nodes.

As seen in figures 8(c), 9(c) and 10(c), the horizontal shift between the fitted curve on the data (red) and the predicted  $H$  curve (black) varies for each dynamical system. This observation is compatible with the reduction theorem, which is analytically proven only for expanding maps [4]. We show that even if the theorem is not rigorously

proven for the Rulkov map, Hénon map or Tinkerbell map, the methodology accurately works. For the coupling,  $H(x, y) = y - x$ ,  $v(x) = \int H(x, y) dm(y)$  is found as  $-x$  plus a constant. These constants are found as 1 for coupled Rulkov maps,  $-0.3$  for coupled Hénon maps and  $0.1$  for coupled Tinkerbell maps. For sinusoidal coupling,  $H(x, y) = \sin 2\pi y - \sin 2\pi x$ ,  $v(x) = \int H(x, y) dm(y)$  is found as  $-\sin 2\pi x$  plus a constant and it is 0 for sin-coupled Hénon maps.

## VI. MACAQUE MONKEY VISUAL CORTEX NETWORK

We tried our procedure on a real monkey visual cortex network that is not a scale-free type [5, 6]. The network consists of 91 nodes and 581 undirected and unweighted edges, which is illustrated in Fig. 11(a) with its degree distribution in Fig. 11(b). It is impossible to classify nodes by degree in a network with such a degree distribution, failure to classify nodes results in an unable to learn local dynamics and identify the hub. So, we assume we know the local dynamics and which node is the hub. When we continue with this preliminary information, the reduction theorem allows us to learn the coupling function smoothly and see full reconstruction.

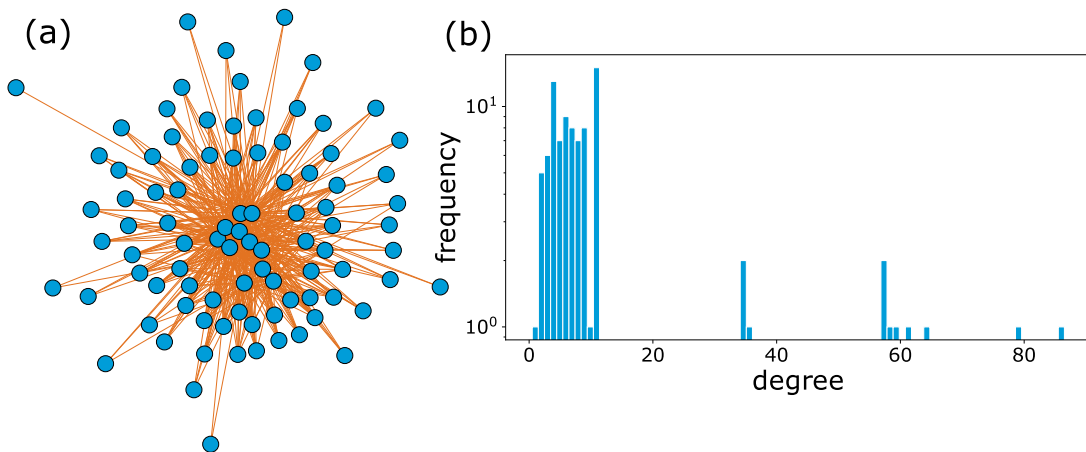


FIG. 11: (a) Undirected and unweighted macaque monkey visual-cortex network. (b) Degree distribution of the associated real network.

## VII. SPARSE OPTIMIZATION

The model for the dynamical system is given by the function  $\mathbf{g}$ :

$$\mathbf{g}(\mathbf{x}) \approx \sum_{k=1}^p \psi_k(\mathbf{x}) \xi^k \quad (13)$$

where  $\Psi(\mathbf{x}) = [\psi_1(\mathbf{x}), \psi_2(\mathbf{x}), \dots, \psi_p(\mathbf{x})]$  is the library of basis functions. Only a few active terms characterize data that comes from natural systems on a well-chosen basis; most of the coefficients  $\xi^k$  are zero. To find interpretable models sparsity concept is beneficial. This idea was used in a compressed sensing framework for the first time as the sparsity requirement is satisfied [7]. If we have prior knowledge to decide the basis, sparse regression is instrumental in avoiding overfitting and noise [8]. In our reconstruction procedure, we use sparse regression in more than one step.

The least absolute shrinkage and selection operator (LASSO) [9] is a sparse regression method that uses  $\ell_1$ -norm to promote sparsity. Sequentially-thresholded least-squares (STLS) is another method implemented in the Python package of SINDy [10]. In STLS, we start with an ordinary least squares solution which results in overfitting the time series at each point. We repeat least squares to the values obtained by thresholding the residues below a particular cut-off value called sparsity parameter at each sequence. Finally, an interpretable dynamical model is obtained.

### VIII. LIBRARY OF BASIS FUNCTIONS

Although prior knowledge about the dynamical system is always helpful in constructing a basis library, it is impossible to access the information for all cases. Nevertheless, many functions can be accurately estimated using high-order polynomials. For instance, in this work, we mainly studied with the Rulkov maps defined as follows,

$$\begin{aligned} u(t+1) &= \frac{\beta}{1+u(t)^2} + v(t) \\ v(t+1) &= v(t) - \nu u(t) - \sigma \end{aligned} \quad (14)$$

Using a basis library containing only polynomials, it is possible to reconstruct an imperfect model for the Rulkov map, which generates quite an accurate time series to the original one (Fig. 12). For the Rulkov map, only the non-polynomial term is the rational one,  $\frac{1}{1+u^2}$ , which is absent in the library. However, this rational term can be expanded in a power series as follows,

$$\frac{1}{1+x^2} = \sum_{n=0}^{\infty} (-x^2)^n = 1 - x^2 + x^4 - x^6 + x^8 + \dots \quad (15)$$

therefore, it is possible to model the Rulkov map with polynomials roughly. As the power expansion's convergent interval is  $[-1, 1]$ , the predicted model diverges when the orbit is out of the given interval. However, such convergence problems can be solved by normalizing the time series into the associated space. The interesting dynamics for our study is the bursting regime which can be modeled without the normalization Fig. 12 (b); therefore, we skip this step, but the model becomes very complicated with many higher-order polynomial terms.

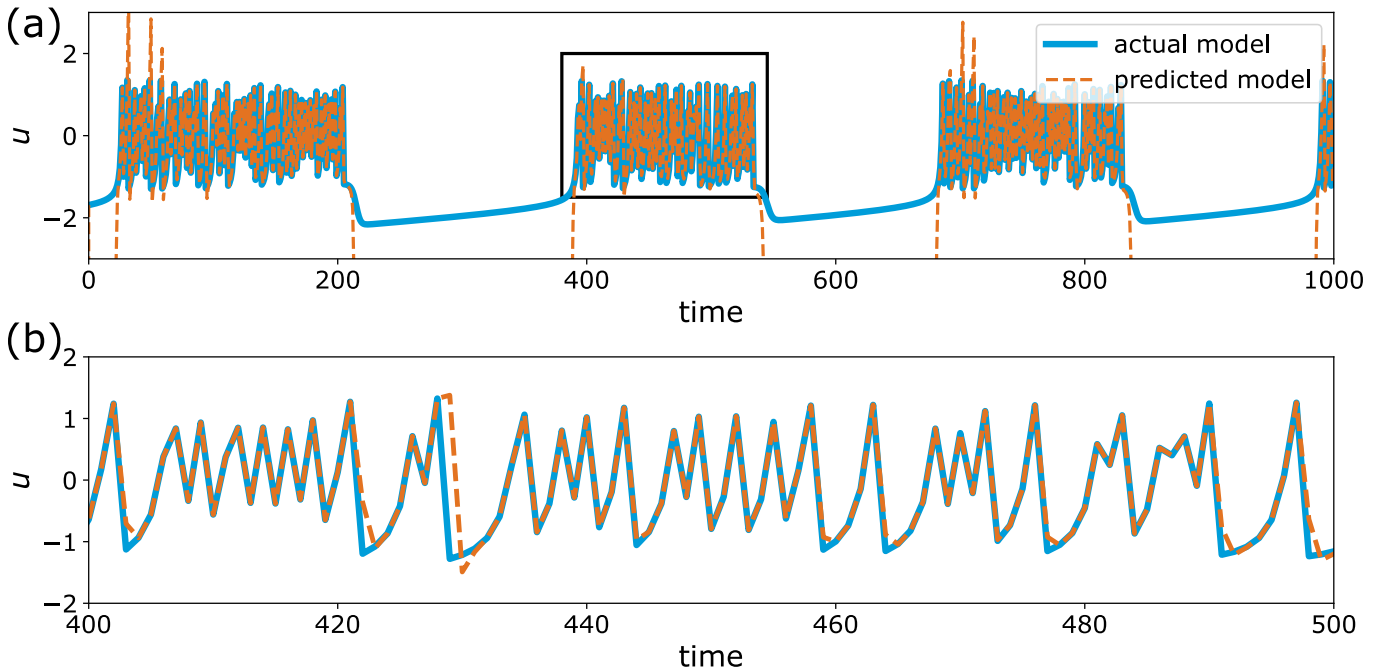


FIG. 12: The predicted time series of a reconstructed Rulkov model using a candidate functions library containing only polynomials. a) predicted  $u$ -variable of the model b) a bursting regime package zoomed from the black rectangle in a).

In order to avoid complicated models containing many polynomial terms and possible inaccurate predictions, we prepared a standard library including common functions, such as geometric series and trigonometric functions. Although unreasonably increasing the number of functions in the library is undesired, adding such nonpolynomial terms generally terminates a large number of polynomials. Thus, our computational cost mostly becomes cheaper using the following standard candidate library:

$$\Psi = \left[ \mathbf{1}, \mathcal{P}_d(\mathbf{x}), \left\{ \sin(i\mathbf{x}), \cos(i\mathbf{x}) \right\}_{i=1}^r, \left\{ \frac{1}{\mathbf{x}^i}, \frac{1}{1 \pm \mathbf{x}^i}, \frac{1}{(1 \pm \mathbf{x})^i} \right\}_{i=1}^q \right]$$

where  $d$  is the degree of the polynomials, and for instance  $\mathcal{P}_2(\mathbf{x})$  is:

$$\mathcal{P}_2(\mathbf{x}) = [u, v, u^2, uv, v^2].$$

Our main strategy is to start  $d = r = q = 1$ , and increase them slowly to estimate the model with the minimum number of candidate functions. The first step of our reconstruction algorithm aims to learn the local dynamics by classifying the inferred model equations due to their similarity. Here, we illustrate the reconstructed models on the mouse neocortex network using coupled Rulkov maps (Table I) through a set of selected nodes' with respect to their degrees  $k_i$  ( $k_0 = 0$ ,  $k_{46} = k_{136} = 4$ , and  $k_{218} = 26$ ). As node-0 has no incoming links, we reconstructed only the Rulkov map dynamics. Node-46 has 4 incoming links, the inferred coefficients for it do not reflect the exact governing equation, but one can say that its governing equations are similar to node-136, which also has 4 incoming links. The most *distinct* model belongs to node-218, which is the hub of this particular network.

node id	1	$u$	$v$	$\dots$	$\cos(u)$	$\frac{1}{1-u}$	$\frac{1}{1+u^2}$
0	-0.000	-0.000	1.000	$\vdots$	0.000	-0.000	4.100
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
46	0.000	-0.015	1.005	$\vdots$	0.005	0.000	4.093
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
136	-0.000	-0.015	1.006	$\vdots$	0.000	-0.000	4.101
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
218	-0.714	0.098	0.783	$\vdots$	0.016	0.001	4.075

TABLE I: Table for the inferred coefficients of the corresponding candidate functions for some of the selected nodes. We present only nonzero functions for any node.

The first step of the reconstruction algorithm is to classify nodes, which uses a distance between nodes' coefficients. We define the distance metric as,  $d_{ij} = (\sum_{k=1}^p \frac{1}{V_k} |\xi_i^k - \xi_j^k|^2)^{1/2}$  where  $|\cdot|$  is absolute value,  $V_k$  is the variance of the predicted coefficients of the  $k$ -th function in  $\Psi$ . As an example, we compute the pairwise distances of the learnt models given in Table I:

node id	0	46	136	218
0	0.000	2.830	2.105	18.351
46	2.830	0.000	1.925	16.808
136	2.105	1.925	0.000	16.979
218	18.351	16.808	16.979	0.000

TABLE II: Pairwise distances of the predicted models for some of the selected nodes.

As numerically shown in Table II,  $d_{ij}$  is small for the nodes with the same or similar degrees, such as the distance is 1.925 between 46 ( $k_{46} = 4$ ) and 136 ( $k_{136} = 4$ ), while the distance is large for different predicted models, such as the distance is 18.351 between 46 ( $k_{46} = 4$ ) and 218 ( $k_{218} = 26$ ).

## IX. COMPARISON BETWEEN SPARSE REGRESSION METHODS

We compare two optimizers to perform sparse regression; LASSO and STLS.

LASSO is defined by the equation [11]:

$$\hat{\mathbf{w}}_{LASSO}(\lambda) = \underset{\mathbf{w} \in \mathbb{R}^m}{\operatorname{argmin}} (||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 + \lambda ||\mathbf{w}||_1) \quad (16)$$

for a linear problem  $\mathbf{y} = \mathbf{X}\mathbf{w}$  where  $\mathbf{y} \in \mathbb{R}^n$ ,  $\mathbf{X} \in \mathbb{R}^{n \times m}$  and  $\lambda$  is *penalty term* and uses  $\ell_1$ -norm to penalize the weights.

In STLS, Ridge regression is used and defined by the equation [11]:

$$\hat{\mathbf{w}}_{Ridge}(\lambda) = \underset{\mathbf{w} \in \mathbb{R}^m}{\operatorname{argmin}} (|\mathbf{X}\mathbf{w} - \mathbf{y}|_2^2 + \lambda|\mathbf{w}|_2^2) \quad (17)$$

and  $\lambda$  is fixed as 0.05. Sparsity is obtained by masking out elements of  $\hat{\mathbf{w}}$  that are below a given threshold. This threshold is called as *sparsity parameter* and it is the single hyper-parameter of STLS. In the final step of our reconstruction approach, we find a sparse solution for  $\mathbf{w}$ , it corresponds to  $\mathbf{L}$  in our case. From the algorithmic point of view, fine-tuning of the hyper-parameters, penalty term for LASSO and sparsity parameter for STLS, is essential to improve the accuracy. These two parameters are not equivalent, however we can make a similar interpretation between them. The length of the time series determines the problem type since our network size is fixed in the experiments. The time series length lower than the network size ( $T < nm = 987 \times 2 = 1974$ ) correspond to an under-determined case for the real network we use.

Fig. 13 presents the reconstruction performance for different time series lengths and two hyperparameters as *FNR* and *FPR*. We compare *FNR* values to evaluate the performance of LASSO and STLS in Fig. 13(a) and Fig. 13(c), since *FPR* values are always very small. The results show that LASSO overcomes STLS for shorter time series. Since the  $\ell_1$ -norm penalized solution, is a generalization of compressive sensing approach, it provides a unique solution for underdetermined cases [12]. On the other hand, if there is long enough data, STLS becomes a fast alternative to the LASSO from an algorithmic point of view and works very well for our problem. We took one under-determined case ( $T = 300$ ) to compare the performance of two different sparsity promoting methods in Fig. 14. If we have limited data, we see full reconstruction by LASSO but not by STLS.

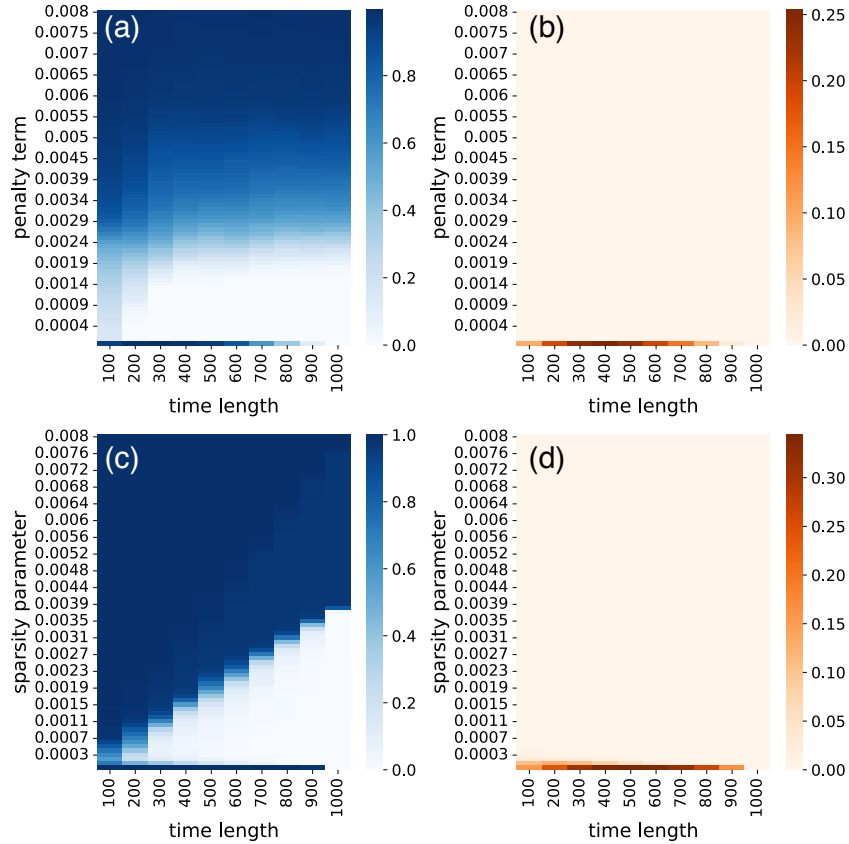


FIG. 13: (a) *FNR* and (b) *FPR* with respect to a list of penalty terms for LASSO. (c) *FNR* and (d) *FPR* with respect to a list of sparsity parameters for STLS. 10 different lengths of time series are used to compare the performance.



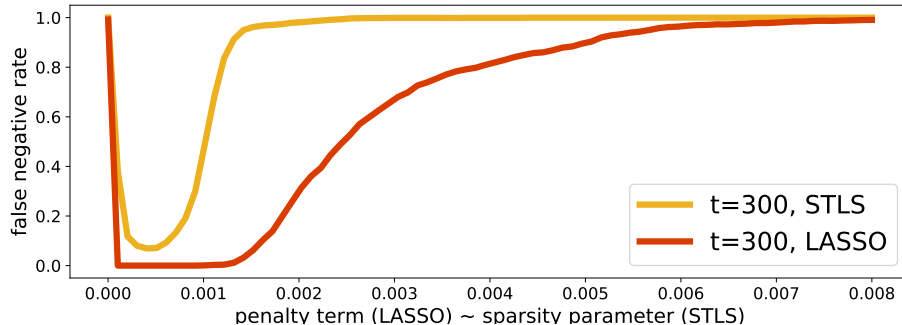


FIG. 14:  $FNR$  with respect to a list of penalty term for Lasso and sparsity parameter for STLS. ( $T = 300$ ) case is used as a short time series example. There is no *full reconstruction* region for STLS.

## X. EFFECT OF NETWORK SPARSITY ON RECONSTRUCTION PERFORMANCE

In addition to the lack of sufficient data, the sparsity condition of the Laplacian matrix should be satisfied to solve the linear regression problem within the compressed sensing framework. We present results in this direction based on synthetical networks. The algorithm we use to generate directed scale-free networks has three probabilistic parameters:  $\alpha$  and  $\gamma$  are the probabilities of adding a new node connected to an existing node chosen randomly according to the in-degree and out-degree distribution, respectively.  $\beta$  is the probability of adding an edge between two existing nodes. We use two different parameter setting as  $[\alpha = 0.41, \gamma = 0.05, \beta = 0.54]$  and  $[\alpha = 0.2, \gamma = 0.5, \beta = 0.3]$  to obtain Laplacian matrices at different sparsity levels. The former (default) setting's power exponent is approximately 2 and generates denser networks, while the latter's power exponent is approximately 1 and generates sparser networks. As seen in Fig. 15, we see full reconstruction in sparser networks since the Laplacian matrix meets the sparsity condition for the compressive sensing approach [13].

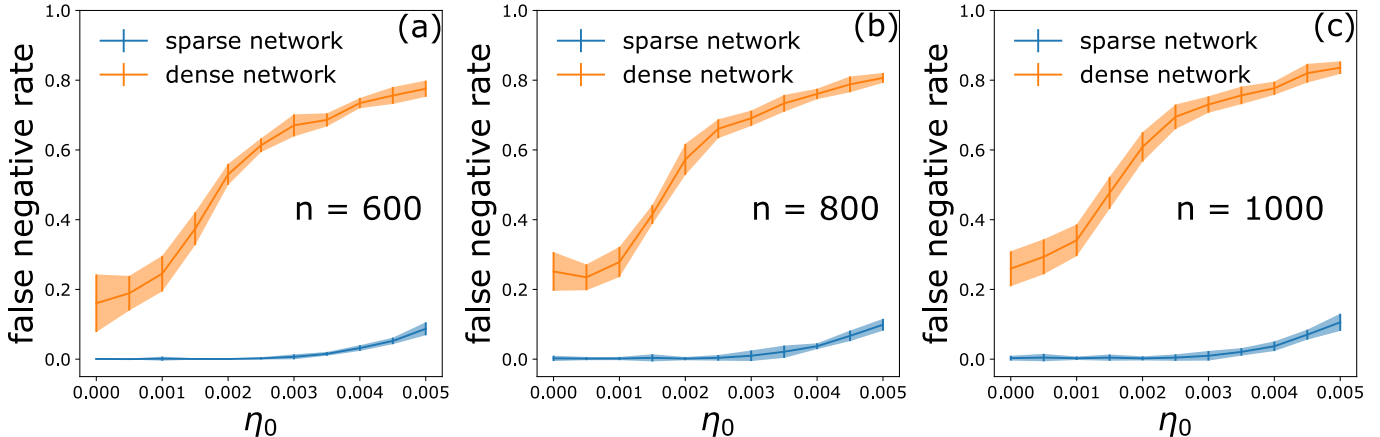


FIG. 15:  $FNR$  against noise for sparse and dense networks of sizes (a)  $n = 600$ , (b)  $n = 800$  and (c)  $n = 1000$ . All points show an average error over 10 different realizations of scale free networks and shaded regions present standard deviations. Time series length is 500, which corresponds to underdetermined cases for all three network sizes.

## XI. CROSS VALIDATION OF INFERRED LAPLACIAN MATRICES

In the main text, we use the ground-truth Laplacian matrix to show the performance of the reconstruction for various hyper-parameters and data lengths. However, it is unlikely to know the true matrix to evaluate the methodology's success in real-world problems. Therefore, it is important to assess the reconstruction outcome when the real matrix is absent using a cross-validation technique. To illustrate the cross-validation approach, we obtained *predicted time series* of length 500 using inferred Laplacian matrices for various hyper-parameters (penalty term)  $\lambda = 0.008, 0.003$  and  $0.0003$ . Out of those three parameters, the outcome for  $\lambda = 0.0003$  gives the best-fit, Fig. 16(a).

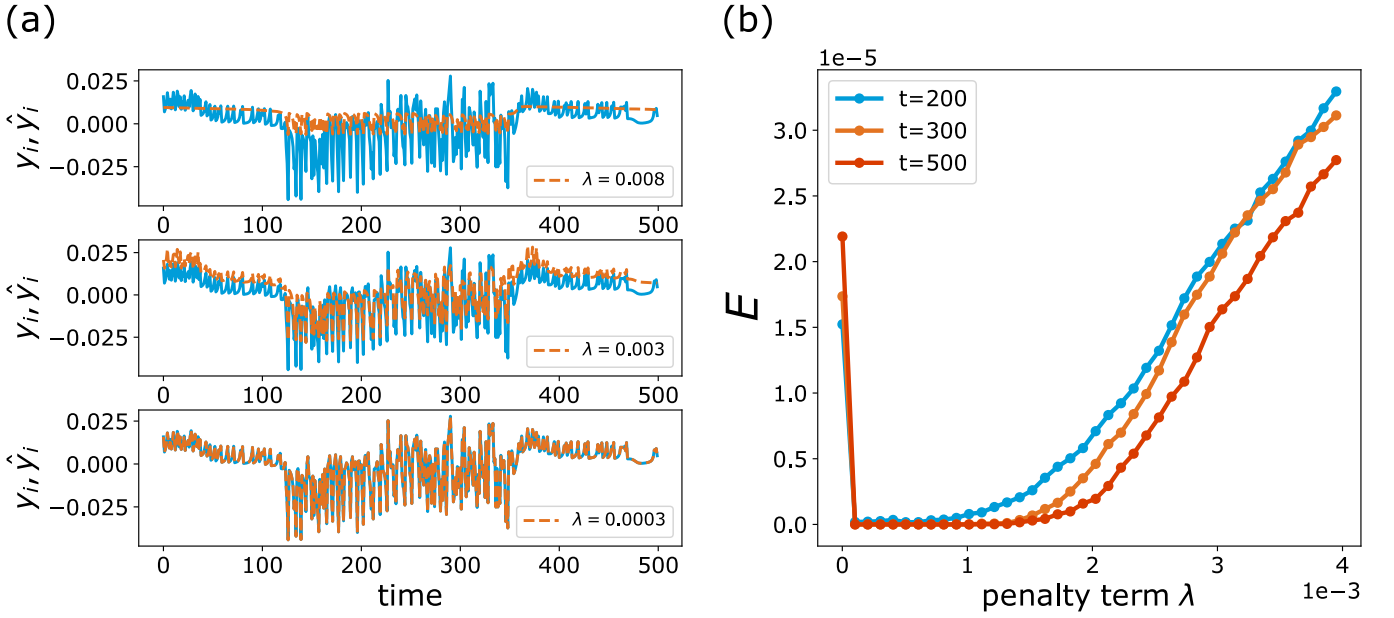


FIG. 16: a) Comparison between measurement of a node and test time series. Each  $\lambda$  parameter corresponds to different connectivity matrices. We simulate the inferred models and obtain test time series. b) Cross validation of the inferred time series based on obtained connectivity matrices for a series of penalty terms.

To find the best hyper-parameters in large network dynamics, we use an error function

$$E = \frac{1}{n} \sum_{i=1}^n \left( \frac{1}{T} \sum_{t=1}^T (y_{i,t} - \hat{y}_{i,t})^2 \right) \quad (18)$$

where  $y_i = x_i(t+1) - f(x_i)$  and  $\hat{y}$  denotes the predicted one. Then we optimize the hyper-parameters, which minimize the error. The parameters yielding the most accurate results can be used as the optimal hyper-parameter (Fig. 16(b)).

## XII. WHY IS THE REDUCTION THEOREM CRUCIAL TO RECONSTRUCT LARGE NETWORKS?

A fully-algorithmic approach is to apply any sparse optimization technique directly to the entire multi-variable time series. Only up to a certain size networks can be recovered by this method. By the reduction theorem, we learn local dynamics from low degree nodes and interaction dynamics from the hub. We are not looking for a sparse linear combination of functions within a massive library of all possible functions. Therefore, there is no limit on network size in our approach. In this section, we will show this with a simple example:

Assume we have a network with 5 nodes. If node 1 has links coming from the nodes 0, 4 and 5 as seen in Fig. 17, then the dynamics of node 1 is written simply in Laplacian matrix form:

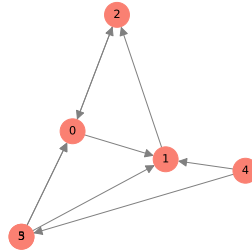


FIG. 17: 5-nodes network as an example.

$$\mathbf{x}_1(t+1) = \mathbf{f}(\mathbf{x}_1(t)) + 3\mathbf{H}(\mathbf{x}_1(t)) - \mathbf{H}(\mathbf{x}_0(t)) - \mathbf{H}(\mathbf{x}_4(t)) - \mathbf{H}(\mathbf{x}_5(t)) \quad (19)$$

where  $\mathbf{x}_i \in \mathbb{R}^m$ . Eq.(19) is in the form of a linear combination of the connections, allowing any sparse regression algorithm to capture the parameters correctly in case of a well-defined basis library. For one single equation, it seems pretty feasible, but for a 5-nodes network, each has  $m$  components, we have  $5m$  equations to be discovered. Suppose we set the basis composed of polynomials up to second-degree and all quadratic combinations of all variables due to a piece of prior knowledge about the coupling function. This example setting has 9 functions for  $m = 3$ . In that case, our basis includes 45 features for this particular example. If we do not have prior knowledge, we use an extensive library that includes more candidate functions. We performed experiments in this direction to see the limitations. The results showed that a purely algorithmic equation-based learning approach fails as the network size, that is, the basis grows. Results of Novaes *et al.* validated our conclusions, they showed that adding new functions to the basis due to increase of network size (up to 20-nodes) effects badly the reconstruction [14]. Furthermore, in the final sparse regression step, our problem turns into a linear equation as  $\mathbf{y} = \mathbf{A}\mathbf{x}$ , where  $\mathbf{y} \in \mathbb{R}^T$ ,  $\mathbf{A} \in \mathbb{R}^{T \times n}$  and  $\mathbf{x} \in \mathbb{R}^n$ . Here,  $\mathbf{y}$  is the data, and the length of the data is  $T$ ,  $n$  is the network size, and we focus on the underdetermined cases ( $T < n$ ) since data availability is always limited. From the perspective of compressed sensing, the measurement matrix  $\mathbf{A}$  should satisfy some conditions to converge the sparsest solution for  $\mathbf{x}$  by  $\ell_1$ -norm. One of these conditions can be summarized as the time series length scales with the network size, so if the measurement matrix has more columns, longer measurements are needed. It is given by the relation:

$$T \approx \mathcal{O}(K \log(n/K))$$

where  $K$  is the network's sparsity measure (number of nonzero entries) [15]. If the number of columns is determined by the basis functions that include all possible pairwise interactions, not the network size, this means the need for a longer time series. On the other hand, the uncertainty of the interaction function is an obstacle to inferring the exact connection matrix [16]. Finding  $\mathbf{H}$  and therefore reducing the dimension of the optimization problem for full reconstruction broadens the applicability and distinguishes our work.

### XIII. GENERAL DIFFUSIVE COUPLING AND MASTER STABILITY FUNCTION

Consider the function  $\mathbf{h} : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m$ . We say that  $\mathbf{h}$  is diffusive if

$$\mathbf{h}(x, x) = \mathbf{h}(0) = 0 \text{ and } \mathbf{h}(x, y) = -\mathbf{h}(y, x)$$

We again consider the model to a general diffusive coupling

$$\mathbf{x}_i(t+1) = \mathbf{f}(\mathbf{x}_i(t)) + \gamma \sum_{j=1}^n w_{ij} \mathbf{h}(\mathbf{x}_j, \mathbf{x}_i) \quad (20)$$

where  $\gamma$  is the coupling strength.

We perform the analysis close to synchronization  $\mathbf{x}_i = \mathbf{s} + \boldsymbol{\xi}_i$  so

$$\mathbf{h}(\mathbf{x}_j, \mathbf{x}_i) = \mathbf{h}(\mathbf{s} + \boldsymbol{\xi}_j, \mathbf{s} + \boldsymbol{\xi}_i) = \mathbf{h}(\mathbf{s}, \mathbf{s}) + D_1 \mathbf{h}(\mathbf{s}, \mathbf{s}) \boldsymbol{\xi}_j + D_2 \mathbf{h}(\mathbf{s}, \mathbf{s}) \boldsymbol{\xi}_i$$

but since the coupling is diffusive

$$D_2 \mathbf{h}(\mathbf{s}, \mathbf{s}) = -D_1 \mathbf{h}(\mathbf{s}, \mathbf{s})$$

we get

$$\mathbf{h}(\mathbf{x}_j, \mathbf{x}_i) = \mathbf{H}(\mathbf{s})(\boldsymbol{\xi}_j - \boldsymbol{\xi}_i) + \mathbf{R}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$$

where  $\mathbf{H}(\mathbf{s}) = D_1 \mathbf{h}(\mathbf{s}, \mathbf{s})$  and  $\mathbf{R}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j)$  contains quadratic terms. Then, the first variational equation about the synchronization manifold

$$\boldsymbol{\xi}_i(t+1) = D\mathbf{f}(\mathbf{s}(t))\boldsymbol{\xi}_i(t) + \gamma \sum_{j=1}^n w_{ij} \mathbf{H}(\mathbf{s})(\boldsymbol{\xi}_j(t) - \boldsymbol{\xi}_i(t)) \quad (21)$$

$$= D\mathbf{f}(\mathbf{s}(t))\boldsymbol{\xi}_i(t) - \gamma \mathbf{H}(\mathbf{s}) \sum_{j=1}^n L_{ij} \boldsymbol{\xi}_j(t). \quad (22)$$

All blocks have the same form which are different only by  $\lambda_i$ , the  $i$ th eigenvalue of  $\mathbf{L}$ . Then we obtain the parametric equation for the modes

$$\mathbf{u}(t+1) = [D\mathbf{f}(\mathbf{s}(t)) - \kappa\mathbf{H}(\mathbf{s}(t))]\mathbf{u}(t)$$

where  $\kappa = \gamma\lambda_i$ . By fixing  $\kappa$ , we compute the maximum Lyapunov exponent  $\Lambda(\kappa)$  as

$$\|\mathbf{u}(t)\| \leq Ce^{\Lambda(\kappa)t}.$$

The map

$$\kappa \mapsto \Lambda(\kappa) \tag{23}$$

is called master stability function. Notice that if  $\Lambda(\kappa) < 0$  when  $\kappa > \gamma_c\lambda_2$  then  $\|\mathbf{u}\| \rightarrow 0$ . For more details on the master stability function Eq. 23, see e.g., [17, 18].

- [1] B. Barzel and A. L. Barabási, *Nature Physics* **9**, 673 (2013), ISSN 17452481.
- [2] T. Pereira, D. Eroglu, G. B. Bagci, U. Tirnakli, and H. J. Jensen, *Phys. Rev. Lett.* **110**, 234103 (2013).
- [3] N. Kasthuri, K. J. Hayworth, D. R. Berger, R. L. Schalek, J. A. Conchello, S. Knowles-Barley, D. Lee, A. Vázquez-Reina, V. Kaynig, and T. R. J. *et al.*, *Cell* **162**, 648 (2015), ISSN 10974172.
- [4] D. Eroglu, M. Tanzi, S. van Strien, and T. Pereira, *Physical Review X* **10**, 1 (2020), ISSN 2160-3308.
- [5] N. T. Markov, J. Vezoli, P. Chameau, A. Falchier, R. Quilodran, C. Huissoud, C. Lamy, P. Misery, P. Giroud, S. Ullman, et al., *Journal of Comparative Neurology* **522**, 225 (2014).
- [6] J. T. Vogelstein, E. Perlman, B. Falk, A. Baden, W. G. Roncal, V. Chandrashekhar, F. Collman, S. Seshamani, J. L. Patsolic, and K. L. *et al.*, *Nature Methods* **15**, 846 (2018), ISSN 15487105.
- [7] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and C. Grebogi, *Physical Review Letters* **106** (2011), ISSN 00319007.
- [8] S. L. Brunton, J. L. Proctor, J. N. Kutz, and W. Bialek, *Proceedings of the National Academy of Sciences* **113**, 3932 (2016), ISSN 10916490.
- [9] R. Tibshirani, *Source: Journal of the Royal Statistical Society. Series B (Methodological)* **58**, 267 (1996).
- [10] B. M. de Silva, K. Champion, M. Quade, J.-C. Loiseau, J. N. Kutz, and S. L. Brunton, *Journal of Open Source Software* **5**, 2104 (2020), URL <https://doi.org/10.21105/joss.02104>.
- [11] P. Mehta, M. Bukov, C. H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, *Physics Reports* **810**, 1 (2019), ISSN 03701573, URL <https://doi.org/10.1016/j.physrep.2019.03.001>.
- [12] D. Donoho, *IEEE Transactions on Information Theory* **52**, 1289 (2006).
- [13] E. J. Candes, J. K. Romberg, and T. Tao, *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59**, 1207 (2006).
- [14] M. Novaes, E. R. dos Santos, and T. Pereira, *Physica D: Nonlinear Phenomena* **424**, 132895 (2021), ISSN 01672789.
- [15] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, 2019).
- [16] Y. Y. Liu and A. L. Barabási, *Reviews of Modern Physics* **88**, 035006 (2016), ISSN 15390756.
- [17] L. Huang, Q. Chen, Y.-C. Lai, and L. M. Pecora, *Phys. Rev. E* **80**, 036204 (2009), URL <https://link.aps.org/doi/10.1103/PhysRevE.80.036204>.
- [18] P. Schultz, T. Peron, D. Eroglu, T. Stemler, G. M. Ramírez Ávila, F. A. Rodrigues, and J. Kurths, *Phys. Rev. E* **93**, 062211 (2016).