






RESEARCH ARTICLE | APRIL 04 2024

Deep-learning reconstruction of complex dynamical networks from incomplete data EP FREE

Special Collection: [Data-Driven Models and Analysis of Complex Systems](#)

Xiao Ding ; Ling-Wei Kong ; Hai-Feng Zhang  ; Ying-Cheng Lai 

 Check for updates

Chaos 34, 043115 (2024)

<https://doi.org/10.1063/5.0201557>



Chaos

Special Topic: Nonautonomous Dynamics
in the Climate Sciences

[Submit Today](#)

 AIP
Publishing

 AIP
Publishing

Deep-learning reconstruction of complex dynamical networks from incomplete data

Cite as: Chaos 34, 043115 (2024); doi: 10.1063/5.0201557

Submitted: 31 January 2024 · Accepted: 19 March 2024 ·

Published Online: 4 April 2024



View Online



Export Citation



CrossMark

Xiao Ding,¹  Ling-Wei Kong,²  Hai-Feng Zhang,^{1,a)}  and Ying-Cheng Lai² 

AFFILIATIONS

¹The Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Mathematical Science, Anhui University, Hefei 230601, China

²School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA

Note: This paper is part of the Focus Issue: Data-Driven Models and Analysis of Complex Systems.

^{a)} **Author to whom correspondence should be addressed:** haifengzhang1978@gmail.com

ABSTRACT

Reconstructing complex networks and predicting the dynamics are particularly challenging in real-world applications because the available information and data are incomplete. We develop a unified collaborative deep-learning framework consisting of three modules: network inference, state estimation, and dynamical learning. The complete network structure is first inferred and the states of the unobserved nodes are estimated, based on which the dynamical learning module is activated to determine the dynamical evolution rules. An alternating parameter updating strategy is deployed to improve the inference and prediction accuracy. Our framework outperforms baseline methods for synthetic and empirical networks hosting a variety of dynamical processes. A reciprocity emerges between network inference and dynamical prediction: better inference of network structure improves the accuracy of dynamical prediction, and vice versa. We demonstrate the superior performance of our framework on an influenza dataset consisting of 37 US States and a PM_{2.5} dataset covering 184 cities in China.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0201557>

Three important topics in the network science, i.e., link prediction, network reconstruction, and dynamics prediction, have been separately considered in the past, inspired by the fact that, in many real situations, only some incomplete data regarding the time series of partial nodes or the partial structure of networks are observed. One natural question is: how to use the incomplete data to infer the network structure as well as predict the dynamics simultaneously. Evidently, the raised problem is more meaningful and more challenging because it means that the three seemingly independent problems should be solved at the same time. To this end, we develop a collaborative deep-learning framework for network inference and dynamical prediction. Experiments demonstrate that our method outperforms the baseline method for both tasks for different networks (artificial and empirical) and different dynamics (discrete and continuous). Our proposed framework is able to build a bridge between network inference and dynamical prediction, i.e., the good network structure can improve the accuracy of dynamical prediction, and the good dynamical rule can improve the accuracy of network inference.

I. INTRODUCTION

A variety of real-world systems, such as biological, chemical, social, and the climate systems, are complex networks composed of nonlinear dynamical units,^{1,2} and the emergent collective dynamics and their reliance on the network structure have been an area of active research.^{3,4} However, from a structural point of view, the detailed connecting topology of the network is often unknown, hindering our ability to predict the dynamical behaviors that can take place on the network.^{5,6} A viable approach to deciphering the topological structure is through time-series data.⁷ From the point of view of network dynamics, learning the rules and predicting the dynamical evolution of the networked system from data are important for applications.^{8,9} A tacit assumption in most existing works is that time-series data from all nodes in the network are available, which typically does not hold in practical situations. To reconstruct the network structure and predict the dynamical processes from partial, and often severely incomplete data and information is, thus, realistically relevant and significant, but the problem is challenging and remains largely outstanding.

Previously, a number of data-driven approaches have been developed to infer network structure,^{7,10} which include correlation-based methods,¹¹ causal analysis,¹² compressed sensing,¹³ and statistical inference methods.^{14–17} A related problem is link prediction, where the partial network structure is observed and the aim is to predict the potential or future links that may emerge.^{18,19} A variety of methods have been proposed, including similarity-based methods,^{20,21} maximum likelihood methods,^{22,23} and machine learning methods.^{24,25} In terms of dynamical prediction, a variety of methods have been developed. For example, early methods relied on fitting the parameters for specific dynamical models^{26,27} or finding the nonlinear equations.^{28,29} Predicting the network dynamics is harder because the interactions among the nodes generating the observed dynamical behaviors are typically highly correlated and nonlinear. More recently, machine learning, particularly graph neural networks (GNNs), has been exploited for predicting network dynamics^{5,30} with significant applications, such as predicting the influenza-like illness (ILI),³¹ traffic flow,³² stability of power grids,³³ and the dynamical rules underlying climate change.³⁴ Despite the success of machine learning-based predictions in network dynamics, most existing works made the assumption that the network structure and the full historical time-series data are known.

For network inference and dynamical prediction, data collection generally falls into two categories: structural and time-series data, where the former represents prior knowledge about the network topology while the latter are dynamical data collected from a known set of nodes in the network. Both categories of data can be incomplete or partial. For example, consider a toy network as characterized by the adjacency matrix in Fig. 1(a) and suppose that there is an unknown or “hidden” node (labeled as the bottom-right corner) from which no measurement can be taken, so all the links from other five nodes in the network as represented by the gray boxes along the right and bottom edges in the adjacency matrix are unknown, as illustrated in Fig. 1(b). In this case, the extent of “incompleteness” of the available information is not severe as only the links connected to the hidden node are unknown while the connectivity among all other nodes in the network is known. For this type of “weak” incompleteness in the available structural data, recent works have demonstrated that deep learning can be effective for simultaneously inferring the network structure and predicting the dynamics.^{35–37}

In contrast to the above problem scenarios, a more challenging structural data scenario is illustrated in Fig. 1(c), where the connection relationships in the network are almost completely unknown in the sense that only a small group of observed edges is known and the existence or non-existence of any link is uncertain. To define and contrast the two incomplete data scenarios quantitatively, we use the numbers of unknown parameters that need to be estimated, which are $O(N)$ and $O(N^2)$ for a network of size N in Figs. 1(b) and 1(c), respectively. In all likelihood, to solve the problem of network inference and dynamical prediction in situations where the number of unknown parameters scales quadratically with the network size is significantly more challenging than in cases where the relation is linear. Furthermore, the existing method³⁶ assumed that only the time-series data of hidden nodes are missing, while the time-series data of other observed nodes are known. On the contrary, we assume that the missing time-series data of nodes can be

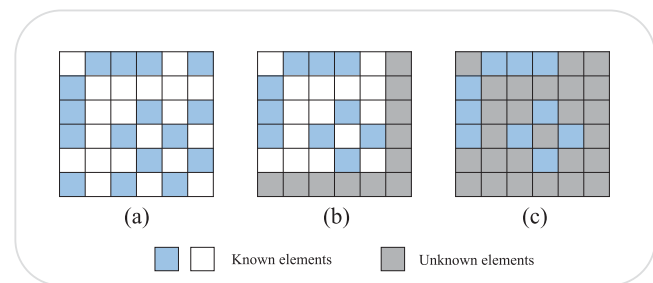


FIG. 1. Illustration of two types of data incompleteness associated with network inference and dynamical prediction. (a) The adjacency matrix of a toy network of six nodes, where the blue and blank boxes indicate an actual link and the non-existence of a link, respectively. (b) A “benign” partial-data scenario where one node (the bottom-right matrix element) is hidden, so the connections between this node and all other nodes in the network are unknown (the gray boxes). In this case, the number of structural parameters to be estimated scales linearly with the network size. (c) A severely incomplete data scenario in which all nodes are hidden and only a handful of links are known. In this case, the number of structural parameters to be determined scales quadratically with the network size.

independent of the data of network structure, and such an assumption is more general. Therefore, in order to address this challenge, we need to develop more effective methods.

In this paper, we develop a collaborative deep-learning framework for network inference and dynamical prediction, and design an alternating updating strategy for the parameters. Our contributions are as follows:

- (1) In order to address the incomplete data scenarios (observing partial structure of networks and time-series data of partial nodes), we develop a data-driven collaborative deep-learning framework (CoND) for network inference and dynamical prediction, which consists of three modules: state estimation, network inference, and dynamical prediction. The first two modules aim to recover incomplete time-series data and structural data, respectively, while the third module aims to learn dynamical rules.
- (2) In order to improve the accuracy of inference and prediction, we design a unique loss function for each module to facilitate parameter learning. Furthermore, we adopt an alternating updating strategy to update the parameters of each module. Specifically, during the parameter updating process, the parameters of the network inference and state estimation modules are fixed, and then the parameters of the dynamical learning module are updated, alternately updating the parameters in this way.
- (3) The experimental results on various networks with discrete and continuous dynamics show that our framework outperforms previous methods in network inference and dynamical prediction tasks. We explicitly validate a reciprocity: knowing the network structure better improves dynamical prediction and better learned dynamical rules in turn makes network inference more accurate. Meanwhile, we also demonstrate the effectiveness of CoND in the influenza dataset and the PM_{2.5} dataset.

The rest of the paper is organized as follows: Sec. II introduces the definition of the problem. In Sec. III, we introduce the proposed CoND framework. Then, we present the experimental setup and

experimental results of two tasks in Sec. IV. Finally, we summarize our conclusions in Sec. V.

II. PROBLEM DEFINITION

In this section, we introduce the definition of the problem is about how to infer network structure and predictive dynamics simultaneously from incomplete data.

Let $G = \{V, E\}$ denote the network to be reconstructed, where $V = \{v_1, v_2, \dots, v_N\}$ and $E = \{e_{ij} = (v_i, v_j) | v_i, v_j \in V\}$ are the sets of nodes and edges, respectively. The adjacency matrix of G is A , where $A_{ij} = A_{ji} = 1$ if nodes i and j are connected, and $A_{ij} = A_{ji} = 0$ if they are not connected. The network dynamics are described by $X^{t+1} = \varphi(X^t, A)$, where $X^t = (X_1^t, X_2^t, \dots, X_N^t)$ denotes the state of all N nodes at time t . The size N of the network (the total number of nodes) is assumed to be known.

Given that the information about the network structure is incomplete and only time series of partial nodes can be observed, we divide G into two components: the observed structure G_o and the hidden structure G_m , with the corresponding adjacency matrices A_o and A_m , respectively, so the full adjacency matrix A can be written as $A = A_o + A_m$. Likewise, X recording the states of all nodes can be divided into two parts: the states X_o of the observed nodes and the states X_m of the unobserved nodes. Given A_o and X_o at time t , the estimated dynamical state X_o at time $t + 1$ can be written as

$$\hat{X}_o^{t+1} = \varphi_\gamma(X_o^t \oplus X_m^t(\alpha), A_o + A_m(\beta)), \quad (1)$$

where α , β , and γ are three parameter sets to be determined, which will be used to estimate the states of nodes, infer the network structure and predict the dynamics, respectively.

Our goal is to infer the full network G as represented by A and learn the network dynamical rules φ based on the incomplete structure described by A_o and the incomplete time-series data X_o . Therefore, we develop deep-learning framework to address this optimization problem.

III. DESCRIPTIONS OF COND FRAMEWORK

In this section, we introduce the details of the three modules and the objective functions in our proposed CoND framework.

The CoND framework consists of three modules: state estimation, network inference, and dynamical learning, as illustrated in Fig. 2. The first two modules aim to recover incomplete data, i.e., time-series data of nodes and the network structure, while the third module aims to learn the dynamical rules. The functionalities of the three modules are described here.

A. The state estimation module

The state estimation module is designed to estimate the states of the unobserved nodes and optimize the parameter set α . The input to this module is the N -dimensional network state vector X^t at time t , where each entry of X^t is a learnable parameter, denoted as α_i^t , on which the state X_i^t of node i depends,

$$X_i^t(\alpha) = f(\alpha_i^t), \quad (2)$$

where α_i^t represents the state learning parameters of node i at time t and f is a function that maps the parameter values to a range of state

values. For example, for a binary state model, the mapping function can be chosen to be a Sigmoid function. The estimated complete time series can be obtained by combining the estimated missing node states with the observed node states: $\hat{X}^t(\alpha) = X_o^t \oplus X_m^t(\alpha)$. As shown in Fig. 2, the input is the states of the observed nodes X_o^t , and the estimated complete node state vector $\hat{X}^t(\alpha)$ can be obtained through this module.

B. The network inference module

The network inference module requires a functional formula of $A_m(\beta)$ to learn the parameter set β . As the connectivity between a pair of nodes is a logical variable (either connected or not), it is difficult to update the parameters characterizing the network structure through backpropagation. To overcome this difficulty, we use the Gumbel-Softmax method³⁸ to infer the candidate adjacency matrix. By this approach, the sampling process of a discrete distribution is simulated through a continuous function so that the distributions produced by the real and the simulated sampling process are the same. Specifically, the candidate adjacency matrix $A_m(\beta)$ inferred by the Gumbel-Softmax approach is defined as

$$(A_m(\beta))_{ij} = \sigma((\log(\beta_{ij}) + \xi_{ij})/\tau), \quad (3)$$

where ξ_{ij} is a random number generated by the Gumbel distribution, σ is the Softmax function, τ is a temperature parameter for adjusting the sharpness of the sampling process. Inspired by the idea of residual connection,³⁹ we use the candidate adjacency matrix $A_m(\beta)$ as the residual term, the complete network structure can be estimated by combining the learned adjacency matrix and the observed structure: $\hat{A}(\beta) = A_o + A_m(\beta)$. As shown in Fig. 2, the input is the observation structure A_o , and the complete network structure $\hat{A}(\beta)$ can be obtained through this module. This not only preserves the information of the original graph but also avoids excessively modifying the topological structure of the graph.

C. The dynamical learning module

In the dynamical learning module, the nonlinear function φ_γ characterizing the dynamical rules needs to be defined. Taking advantage of the strong representation and learning capabilities of GNNs for network dynamics, we use it to represent φ_γ . As shown in Fig. 2, the estimated complete node state vector $\hat{X}^t(\alpha)$ and the complete network structure $\hat{A}(\beta)$ are the inputs, and the state of the observed node at time $t + 1$ can be predicted through Eq. (1), which can be written as

$$\hat{X}_o^{t+1} = \text{GNN}(\hat{X}^t(\alpha), \hat{A}(\beta), \gamma), \quad (4)$$

where γ is the parameter set of the GNN model. In fact, our goal is to understand the dynamical rules by learning this parameter set of the GNN model. We test a number of GNN models, including Graph Convolutional Network (GCN),⁴⁰ Graph SAmple and aggreGAtE (SAGE),⁴¹ and Graph Isomorphism Network (GIN)⁴² (a detailed description of different GNN models is provided in Appendix A).

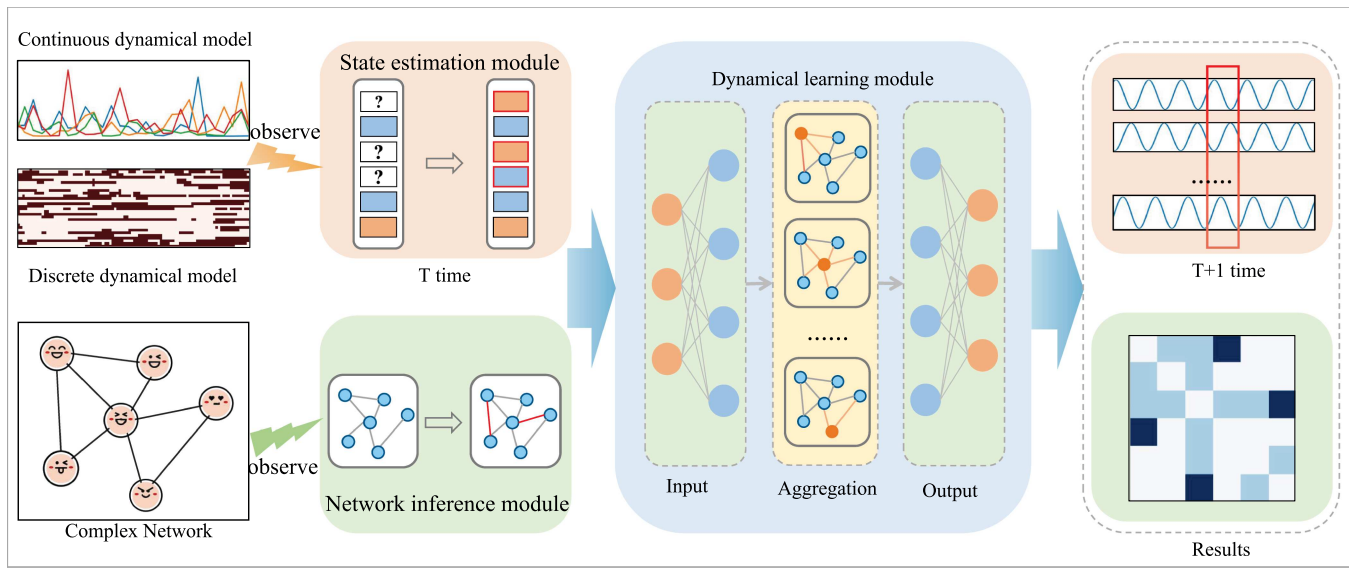


FIG. 2. Proposed collaborative deep-learning framework for network inference and dynamical prediction. The framework consists of three main modules: (1) a state estimation module is used to evaluate the states of the missing nodes (box with symbol “?”) at the observed time step T , (2) a network inference module is used to infer missing edges (red edges) by the Gumbel-Softmax approach based on the observed adjacency matrix, and (3) a dynamical learning module is used to predict the state of the node at the next time step ($T + 1$) through the graph neural network based on inputs from the other two modules (i.e., the state of the node at T and the inferred adjacency matrix). The three modules work “collaboratively” through the strategy of alternating parameter updates to accomplish the network inference and dynamical prediction tasks.

D. Objective function

We design a unique objective function for each module to update its parameters. Specifically, the parameter sets α and β in the state estimation and network inference modules are optimized by minimizing the following loss function:

$$L(\alpha) = \sum_{k=1}^K \sum_{t=0}^{T-1} (L_1(X_o^{t+1}, \hat{X}_o^{t+1}) + L_2(X_o^t, X_o^t(\alpha))), \quad (5)$$

$$L(\beta) = \sum_{k=1}^K \sum_{t=0}^{T-1} L_1(X_o^{t+1}, \hat{X}_o^{t+1}) + \lambda ||\hat{A}_m(\beta)||_1, \quad (6)$$

where L_1 is the Cross Entropy (CE) for discrete dynamics or the Mean Square Error (MSE) for continuous dynamics, L_2 is the MSE loss, and K is the number of batches of samples. Equation (5) incorporates the loss between the observed node states and the predicted values with the goal of better learning the states of the missing nodes. Equation (6) adds λ to the loss function to sparsify the learned network to avoid overfitting and we set $\lambda = 1/N^2$. By fixing the input of $\hat{X}^t(\alpha)$ and the learned $\hat{A}(\beta)$, the parameter set γ in φ_γ is learned by minimizing the following loss function:

$$L(\gamma) = \sum_{k=1}^K \sum_{t=0}^{T-1} L_1(X_o^{t+1}, \hat{X}_o^{t+1}). \quad (7)$$

Equations (5)–(7) stipulate that the difference between the estimated and real values of the observed states should be minimized. By designing different loss functions for each module, it is to enable sufficient learning for each module to improve its performance, thereby achieving the accuracy of network inference and dynamical prediction. Once the objective functions in the three learning modules have been specified, training can start. The optimization process consists of alternatively updating the three modules until the three loss functions converge. We express the training process in the form of pseudocode as in Algorithm 1.

First, the parameters α , β , and γ are initialized (line 1). Then, in each epoch, the dynamical learning module is optimized when the parameters of network inference module and state estimation module are fixed, and the parameter γ is updated by Eq. (7) (lines 4–10). Then, the parameter set α of the state estimation module is updated by Eq. (5) (lines 11–18), and finally, the parameter set β of the network inference module is updated by Eq. (6) (lines 19–26). In each epoch, the three modules are trained alternately, and three modules are constantly optimized in each stage. In order to speed up the running efficiency of the model, we use batch processing to train the model. The batch size is set as P , and the gradient descent algorithm is used to update the parameters α , β and γ step by step with different learning rates lr_α , lr_β , and lr_γ until the end of training. In this paper, we set $P = 9$, and the length of time series is 100 per batch, $D_1 = 30$, $D_2 = 20$, $D_3 = 20$, and epoch = 10. In addition, we set $lr_\alpha = 0.1$, $lr_\beta = 0.1$, and $lr_\gamma = 0.001$, respectively.

ALGORITHM 1. CoND framework.

Input: A_o : the observed adjacency matrix, D_1 : dynamical learning module training rounds, D_2 : state estimation module training rounds, D_3 : network inference module training rounds, P : the number of batch, lr_α : state estimation module learning rate, lr_β : network inference module learning rate, lr_γ : dynamical learning module learning rate.

Output: α, β, γ .

```

1: Initialize parameter  $\alpha, \beta, \gamma$ .
2: for each  $i$  in epochs do
3:    $\hat{X}(\alpha) = X_o \oplus X_m(\alpha), \hat{A}(\beta) = A_o + A_m(\beta)$ 
4:   for  $m = 1, \dots, D_1$  do #Dynamical learning module.
5:     for batch = 1, ...,  $P$  do
6:        $Y_o^{batch} = \varphi_\gamma(\hat{A}(\beta), \hat{X}^{batch}(\alpha))$ .
7:     end for
8:     Calculate Loss ( $\{X_o^1, \dots, X_o^P\}, \{Y_o^1, \dots, Y_o^P\}$ ).
9:      $\gamma = \gamma - lr_\gamma \cdot \Delta\gamma$ .
10:  end for
11:  for  $n = 1, \dots, D_2$  do #State estimation module.
12:     $\hat{X}(\alpha) = X_o \oplus X_m(\alpha)$ 
13:    for batch = 1, ...,  $P$  do
14:       $Y_o^{batch} = \varphi_\gamma(\hat{A}(\beta), \hat{X}^{batch}(\alpha))$ .
15:    end for
16:    Calculate Loss ( $\{X_o^1, \dots, X_o^P\}, \{Y_o^1, \dots, Y_o^P\}, \hat{X}_m(\alpha)$ ).
17:     $\alpha = \alpha - lr_\alpha \cdot \Delta\alpha$ .
18:  end for
19:  for  $n = 1, \dots, D_3$  do #Network inference module.
20:     $\hat{A}(\beta) = A_o + A_m(\beta)$ 
21:    for batch = 1, ...,  $P$  do
22:       $Y_o^{batch} = \varphi_\beta(\hat{A}(\beta), \hat{X}^{batch}(\alpha))$ .
23:    end for
24:    Calculate Loss ( $\{X_o^1, \dots, X_o^P\}, \{Y_o^1, \dots, Y_o^P\}, \hat{A}_m(\beta)$ ).
25:     $\beta = \beta - lr_\beta \cdot \Delta\beta$ .
26:  end for
27: end for

```

IV. EXPERIMENTAL RESULTS

In this section, we introduce network and dynamical process data, baseline methods, evaluation metrics, and experimental results on different networks and dynamical processes.

TABLE I. Basic information of real networks.

Network	Nodes	Edges	Average degree
Dolphins	62	159	5.1290
Word	112	425	7.5893
Ca-netsci	379	914	4.8232
Email	1133	5451	9.6222

A. Network data

In this section, we introduce the synthesis network and real networks.

1. Synthesis network

The scale network is generated by the Barabási–Albert (BA) networks.⁴³ We add a new node to the network and connect it to an existing node with $m = 2$. Erdős–Rényi (ER) networks⁴⁴ are randomly connected with a probability of 0.02. In Watts–Strogatz (WS) networks,⁴⁵ each node has four neighbors and reconnects randomly with probability 0.2. Here, the network size is $N = 200$.

2. Real networks

We select four real-world networks, including Dolphins network (the network of connections between dolphins⁴⁶), Word network (the network of commonly make up of adjectives and nouns in the novel “David Copperfield”⁴⁶), Ca-netsci network (the collaborative network of scientists engage in network theory and experimentation⁴⁷), Email network (the communication network make up of Email⁴⁸). The description of real networks is shown in Table I.

B. Dynamical process

In this section, we introduce five discrete processes and three continuous processes.

1. Threshold model

In this model,⁴⁹ when the fraction of active neighbors of a node is greater than the threshold, the node becomes active, where no recovery is allowed. In our work, the threshold is set to 1/2. Since all nodes can quickly converge to a steady state, we randomly initialize the states of all nodes every five times steps.

2. Voter model

In the Voter model,⁵⁰ each node adopts the states of a randomly selected neighbor at each time step. Since all nodes can converge to a stable state, we randomly initialize the states of all nodes every ten times steps.

3. Ising model

The Ising model⁵¹ is a classical paradigm for studying ferromagnetic spin from the microscopic point of view in statistical physics. Here, we set $\beta = 2$ to represent the combined effect of temperature and ferromagnetic interaction.

4. Susceptible-infected-susceptible (SIS) model

In the SIS process,⁴⁹ there are two types of nodes: susceptible (S) and infected (I). At each time step, an infected node can infect its susceptible neighbors at the infection rate λ and it recovers to the susceptible state with the recovery rate μ . In our simulations, the values of λ and μ are set to be 0.5 and the number of data resampling steps is 2.

TABLE II. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_acc metric for five discrete dynamical models (the second column) on synthetic and real networks (the first column). The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Threshold	0.831 ± 0.003	0.967 ± 0.003	0.966 ± 0.003	0.911 ± 0.003	0.910 ± 0.003
	Voter	0.602 ± 0.005	0.741 ± 0.004	0.740 ± 0.004	0.685 ± 0.005	0.684 ± 0.005
	Ising	0.583 ± 0.006	0.689 ± 0.002	0.689 ± 0.002	0.645 ± 0.005	0.646 ± 0.003
	SIS	0.792 ± 0.007	0.845 ± 0.003	0.848 ± 0.005	0.839 ± 0.001	0.840 ± 0.001
	SIR	0.824 ± 0.008	0.836 ± 0.003	0.834 ± 0.001	0.834 ± 0.007	0.835 ± 0.003
ER	Threshold	0.817 ± 0.005	0.967 ± 0.005	0.962 ± 0.004	0.897 ± 0.004	0.897 ± 0.004
	Voter	0.595 ± 0.007	0.697 ± 0.003	0.696 ± 0.003	0.660 ± 0.004	0.660 ± 0.004
	Ising	0.579 ± 0.006	0.666 ± 0.002	0.670 ± 0.002	0.643 ± 0.002	0.643 ± 0.002
	SIS	0.804 ± 0.001	0.840 ± 0.003	0.833 ± 0.003	0.831 ± 0.001	0.830 ± 0.001
	SIR	0.810 ± 0.009	0.827 ± 0.008	0.830 ± 0.003	0.825 ± 0.006	0.819 ± 0.006
WS	Threshold	0.813 ± 0.006	0.973 ± 0.003	0.969 ± 0.002	0.908 ± 0.002	0.908 ± 0.002
	Voter	0.620 ± 0.008	0.741 ± 0.002	0.740 ± 0.001	0.719 ± 0.004	0.718 ± 0.004
	Ising	0.562 ± 0.002	0.675 ± 0.002	0.680 ± 0.001	0.653 ± 0.003	0.652 ± 0.003
	SIS	0.880 ± 0.006	0.895 ± 0.001	0.893 ± 0.002	0.892 ± 0.001	0.892 ± 0.001
	SIR	0.850 ± 0.004	0.873 ± 0.002	0.871 ± 0.001	0.872 ± 0.005	0.869 ± 0.003
Dolphins	Threshold	0.897 ± 0.007	0.960 ± 0.003	0.956 ± 0.004	0.870 ± 0.004	0.870 ± 0.003
	Voter	0.702 ± 0.007	0.743 ± 0.005	0.749 ± 0.005	0.687 ± 0.004	0.686 ± 0.004
	Ising	0.645 ± 0.007	0.694 ± 0.004	0.692 ± 0.005	0.645 ± 0.003	0.645 ± 0.004
	SIS	0.686 ± 0.006	0.736 ± 0.003	0.738 ± 0.003	0.736 ± 0.002	0.733 ± 0.002
	SIR	0.669 ± 0.004	0.706 ± 0.003	0.702 ± 0.003	0.711 ± 0.004	0.706 ± 0.004
Word	Threshold	0.869 ± 0.007	0.952 ± 0.004	0.959 ± 0.003	0.875 ± 0.003	0.874 ± 0.003
	Voter	0.621 ± 0.006	0.674 ± 0.004	0.674 ± 0.003	0.629 ± 0.003	0.628 ± 0.003
	Ising	0.596 ± 0.006	0.647 ± 0.003	0.647 ± 0.003	0.610 ± 0.003	0.610 ± 0.004
	SIS	0.664 ± 0.008	0.733 ± 0.004	0.731 ± 0.004	0.739 ± 0.003	0.736 ± 0.004
	SIR	0.631 ± 0.007	0.715 ± 0.004	0.713 ± 0.004	0.714 ± 0.003	0.714 ± 0.003
Ca-netsci	Threshold	0.755 ± 0.008	0.959 ± 0.003	0.958 ± 0.004	0.912 ± 0.003	0.912 ± 0.003
	Voter	0.618 ± 0.005	0.764 ± 0.003	0.769 ± 0.004	0.729 ± 0.002	0.729 ± 0.002
	Ising	0.551 ± 0.006	0.710 ± 0.003	0.703 ± 0.002	0.666 ± 0.002	0.666 ± 0.003
	SIS	0.836 ± 0.008	0.923 ± 0.003	0.928 ± 0.002	0.924 ± 0.004	0.920 ± 0.003
	SIR	0.795 ± 0.007	0.901 ± 0.002	0.904 ± 0.003	0.907 ± 0.002	0.906 ± 0.003
Email	Threshold	0.680 ± 0.008	0.937 ± 0.002	0.933 ± 0.001	0.858 ± 0.001	0.856 ± 0.001
	Voter	0.543 ± 0.003	0.667 ± 0.002	0.671 ± 0.001	0.611 ± 0.001	0.611 ± 0.001
	Ising	0.536 ± 0.004	0.641 ± 0.002	0.640 ± 0.001	0.592 ± 0.002	0.591 ± 0.002
	SIS	0.871 ± 0.008	0.882 ± 0.002	0.880 ± 0.002	0.883 ± 0.004	0.882 ± 0.003
	SIR	0.878 ± 0.009	0.879 ± 0.002	0.878 ± 0.003	0.878 ± 0.002	0.876 ± 0.002

5. Susceptible-infected-recovered (SIR) model

In SIR dynamics,⁵² any node belongs to one of the three types: susceptible (*S*), infected (*I*), and recovery (*R*). A node in the infected state can infect its susceptible neighbors with the infection probability λ . An infected node can return to the recovery state at the recovery rate μ , and a node in the recovery state will not be infected again. In our study, we set both λ and μ to 0.5, and the number of data resampling steps is 2.

6. Kuramoto model

The Kuramoto model⁵³ describes a system of phase-coupled oscillators, with the following equation governing the phase variable

θ_i of the *i*th oscillator:

$$\frac{d\theta_i}{dt} = \omega_i + k \sum_{j=1}^N A_{ij} \sin(\theta_j - \theta_i), \quad (8)$$

where k is the coupling strength, θ_i is the phase angle of oscillator *i*, and ω_i is the intrinsic frequency of node *i* when uncoupled, which is chosen from the uniform random distribution $[-\pi/2, \pi/2]$. The coupling strength k is set to 0.2 and the number of data resampling steps is 10.

7. Branch model

The Branch model⁵⁴ initiates an avalanche by activating one unit. The unit activates each of its connected k neighbors with the

TABLE III. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_mse metric for three continuous dynamical models (the second column) on synthetic and real networks (the first column). The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Kuramoto	0.155 ± 0.015	0.108 ± 0.009	0.110 ± 0.008	0.109 ± 0.006	0.109 ± 0.006
	Branch	0.075 ± 0.006	0.075 ± 0.004	0.060 ± 0.003	0.051 ± 0.003	0.051 ± 0.004
	CML	0.084 ± 0.020	0.059 ± 0.010	0.059 ± 0.009	0.054 ± 0.006	0.054 ± 0.007
ER	Kuramoto	0.171 ± 0.050	0.122 ± 0.006	0.128 ± 0.009	0.129 ± 0.005	0.129 ± 0.005
	Branch	0.024 ± 0.009	0.024 ± 0.006	0.024 ± 0.006	0.023 ± 0.002	0.021 ± 0.002
	CML	0.076 ± 0.014	0.058 ± 0.010	0.054 ± 0.008	0.056 ± 0.004	0.056 ± 0.004
WS	Kuramoto	0.153 ± 0.059	0.117 ± 0.008	0.119 ± 0.009	0.119 ± 0.006	0.118 ± 0.005
	Branch	0.030 ± 0.005	0.017 ± 0.004	0.018 ± 0.004	0.015 ± 0.001	0.015 ± 0.001
	CML	0.087 ± 0.017	0.060 ± 0.008	0.050 ± 0.009	0.056 ± 0.002	0.056 ± 0.002
Dolphins	Kuramoto	0.127 ± 0.013	0.103 ± 0.002	0.104 ± 0.003	0.106 ± 0.003	0.106 ± 0.003
	Branch	0.039 ± 0.001	0.038 ± 0.003	0.033 ± 0.001	0.032 ± 0.001	0.032 ± 0.001
	CML	0.062 ± 0.002	0.053 ± 0.003	0.056 ± 0.006	0.056 ± 0.003	0.056 ± 0.002
Word	Kuramoto	0.131 ± 0.007	0.113 ± 0.006	0.125 ± 0.007	0.117 ± 0.003	0.118 ± 0.004
	Branch	0.082 ± 0.004	0.067 ± 0.005	0.061 ± 0.003	0.059 ± 0.002	0.059 ± 0.004
	CML	0.037 ± 0.002	0.022 ± 0.002	0.030 ± 0.002	0.026 ± 0.001	0.027 ± 0.001
Ca-netsci	Kuramoto	0.105 ± 0.008	0.096 ± 0.006	0.100 ± 0.006	0.082 ± 0.003	0.083 ± 0.004
	Branch	0.075 ± 0.006	0.047 ± 0.004	0.056 ± 0.004	0.035 ± 0.003	0.036 ± 0.003
	CML	0.070 ± 0.005	0.064 ± 0.007	0.065 ± 0.007	0.065 ± 0.004	0.065 ± 0.004
Email	Kuramoto	0.108 ± 0.004	0.071 ± 0.005	0.071 ± 0.003	0.060 ± 0.004	0.061 ± 0.004
	Branch	0.023 ± 0.003	0.021 ± 0.001	0.019 ± 0.001	0.019 ± 0.001	0.019 ± 0.002
	CML	0.029 ± 0.003	0.022 ± 0.002	0.021 ± 0.005	0.022 ± 0.004	0.022 ± 0.002

probability p at the next time step. The activated units can in turn activate their neighbors and so on. The sequence of activation forms an avalanche that ends when the previously activated set of units no longer activates any unit. The control parameter of the model is $\sigma = p \cdot k$. In our simulations, σ is set to 0.9 and the number of data resampling steps is 10.

8. Coupled map lattices (CML) model

Coupled map lattices⁵⁵ represent a class of spatially discrete, discrete-time spatiotemporal dynamical systems. The map equation for the dynamical variable at the i th site is given by

$$x_i(t+1) = (1-s)f(x_i(t)) + \frac{s}{k_i} \sum_{j=1}^N A_{ij}f(x_j(t)), \quad (9)$$

where s is the coupling parameter and k_i is the degree of node i . In our study, we use the mapping function $f(x) = \lambda x(1-x)$ with the parameter values $\lambda = 3.5$ and $s = 0.2$. The number of data resampling steps is 10.

C. Baseline methods

In this section, we employ a number of baseline methods for network inference and dynamical prediction.

We first design one ablation experiment named CoND₀ to study the impact of the state estimation module on the two tasks, in which this module is removed and random values are assigned to the states of the unobserved nodes. We also exploit the state-of-the-art Gumbel Graph Network (GGN) algorithms,³⁶ which is

a data-driven deep-learning model, which can be used to solve the network completion and dynamical prediction problems.

For the network inference task, we also conduct six types of experiments: four link prediction algorithms and two correlation reconstruction algorithms. In particular, link prediction is implemented on the observed network structure using the following four types of algorithms: common neighbors (CNs),²⁰ Adamic-Adar (AA),²¹ resource allocation (RA),⁵⁶ and local path (LP),⁵⁷ and the two correlation algorithms are the correlation coefficient¹¹ and granger causality inference⁵⁸ methods based solely on the time-series data.

For the dynamical prediction task, we further design two comparative algorithms, named as CoD₁ and CoD₂, where the network inference module is removed. Specifically, CoD₁ uses the observed structure A_o and generates a complete set of time-series data of the nodes from the state estimation module to predict the dynamics, CoD₂ exploits the observed structure A_o and then performs random assignment of unobserved node states to predict the dynamics (details of different baseline methods are given in Appendix B).

D. Evaluation metrics

In this section, the metrics used to evaluate the accuracy of the network inference and the dynamical prediction are introduced.

As for the dynamical prediction task, when considering the discrete dynamical models, the state with the highest probability in the probability vector is viewed as the state of node i , denoted as \hat{y}_i . Therefore, we use the accuracy of state prediction (Dyn_acc) as the

TABLE IV. Network inference performance of CoND and several baseline methods in terms of Net_acc metric for five discrete dynamical models (the second column) on synthetic and real networks (the first column). The best accuracy in each row is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Threshold	0.487 ± 0.048	0.915 ± 0.029	0.815 ± 0.020	0.024 ± 0.008	0.769 ± 0.042				
	Voter	0.475 ± 0.069	0.800 ± 0.034	0.675 ± 0.037	0.017 ± 0.013	0.613 ± 0.025				
	Ising	0.413 ± 0.051	0.785 ± 0.031	0.721 ± 0.045	0.010 ± 0.010	0.667 ± 0.013	0.018 ± 0.011	0.017 ± 0.012	0.017 ± 0.012	0.010 ± 0.013
	SIS	0.431 ± 0.074	0.800 ± 0.032	0.792 ± 0.042	0.012 ± 0.010	0.641 ± 0.034				
	SIR	0.344 ± 0.091	0.677 ± 0.043	0.605 ± 0.052	0.023 ± 0.014	0.590 ± 0.014				
ER	Threshold	0.812 ± 0.056	0.944 ± 0.026	0.815 ± 0.058	0.007 ± 0.008	0.804 ± 0.058				
	Voter	0.696 ± 0.042	0.827 ± 0.030	0.769 ± 0.053	0.003 ± 0.004	0.692 ± 0.039				
	Ising	0.675 ± 0.056	0.785 ± 0.028	0.781 ± 0.037	0.001 ± 0.004	0.806 ± 0.031	0.001 ± 0.002	0.004 ± 0.008	0.003 ± 0.008	0.002 ± 0.006
	SIS	0.767 ± 0.085	0.843 ± 0.059	0.837 ± 0.046	0.003 ± 0.004	0.709 ± 0.020				
	SIR	0.710 ± 0.097	0.798 ± 0.069	0.798 ± 0.066	0.005 ± 0.006	0.661 ± 0.032				
WS	Threshold	0.765 ± 0.064	0.948 ± 0.025	0.85 ± 0.035	0.002 ± 0.004	0.813 ± 0.057				
	Voter	0.725 ± 0.061	0.850 ± 0.031	0.773 ± 0.048	0.001 ± 0.005	0.793 ± 0.043				
	Ising	0.830 ± 0.070	0.850 ± 0.056	0.833 ± 0.058	0.002 ± 0.007	0.805 ± 0.032	0.014 ± 0.002	0.018 ± 0.003	0.018 ± 0.003	0.001 ± 0.001
	SIS	0.745 ± 0.039	0.855 ± 0.044	0.783 ± 0.046	0.001 ± 0.004	0.798 ± 0.047				
	SIR	0.595 ± 0.093	0.780 ± 0.038	0.765 ± 0.050	0.001 ± 0.008	0.590 ± 0.042				
Dolphins	Threshold	0.833 ± 0.052	0.900 ± 0.047	0.820 ± 0.057	0.017 ± 0.027	0.700 ± 0.063				
	Voter	0.473 ± 0.056	0.693 ± 0.040	0.673 ± 0.041	0.013 ± 0.027	0.580 ± 0.055				
	Ising	0.513 ± 0.066	0.667 ± 0.032	0.613 ± 0.041	0.016 ± 0.027	0.747 ± 0.050	0.123 ± 0.025	0.127 ± 0.030	0.113 ± 0.024	0.033 ± 0.013
	SIS	0.767 ± 0.057	0.803 ± 0.041	0.853 ± 0.055	0.010 ± 0.027	0.667 ± 0.042				
	SIR	0.547 ± 0.065	0.867 ± 0.040	0.800 ± 0.040	0.023 ± 0.016	0.533 ± 0.058				
Word	Threshold	0.821 ± 0.051	0.762 ± 0.056	0.786 ± 0.048	0.006 ± 0.005	0.733 ± 0.050				
	Voter	0.302 ± 0.079	0.657 ± 0.061	0.610 ± 0.061	0.004 ± 0.005	0.624 ± 0.042				
	Ising	0.302 ± 0.069	0.617 ± 0.055	0.610 ± 0.062	0.002 ± 0.006	0.578 ± 0.0551	0.073 ± 0.031	0.073 ± 0.029	0.064 ± 0.027	0.105 ± 0.037
	SIS	0.500 ± 0.057	0.571 ± 0.046	0.500 ± 0.042	0.001 ± 0.006	0.509 ± 0.044				
	SIR	0.331 ± 0.058	0.714 ± 0.055	0.621 ± 0.056	0.008 ± 0.005	0.476 ± 0.054				
Ca-netsci	Threshold	0.395 ± 0.058	0.847 ± 0.028	0.807 ± 0.036	0.006 ± 0.006	0.700 ± 0.049				
	Voter	0.497 ± 0.049	0.670 ± 0.050	0.750 ± 0.058	0.011 ± 0.004	0.580 ± 0.031				
	Ising	0.545 ± 0.055	0.774 ± 0.033	0.721 ± 0.036	0.006 ± 0.004	0.747 ± 0.016	0.363 ± 0.042	0.569 ± 0.042	0.543 ± 0.043	0.112 ± 0.034
	SIS	0.407 ± 0.051	0.739 ± 0.057	0.691 ± 0.054	0.001 ± 0.006	0.667 ± 0.044				
	SIR	0.482 ± 0.065	0.793 ± 0.051	0.708 ± 0.050	0.006 ± 0.004	0.533 ± 0.018				
Email	Threshold	0.530 ± 0.048	0.872 ± 0.039	0.853 ± 0.029	0.004 ± 0.002	0.695 ± 0.024				
	Voter	0.301 ± 0.053	0.566 ± 0.067	0.535 ± 0.062	0.003 ± 0.001	0.501 ± 0.024				
	Ising	0.261 ± 0.039	0.472 ± 0.036	0.453 ± 0.029	0.008 ± 0.002	0.523 ± 0.015	0.140 ± 0.009	0.143 ± 0.005	0.153 ± 0.008	0.060 ± 0.009
	SIS	0.258 ± 0.052	0.572 ± 0.047	0.555 ± 0.058	0.006 ± 0.002	0.567 ± 0.011				
	SIR	0.147 ± 0.053	0.553 ± 0.049	0.534 ± 0.046	0.007 ± 0.003	0.494 ± 0.019				

TABLE V. Network inference performance of CoND and other baseline methods in terms of Net_acc metric for three continuous dynamical models (the second column) on synthetic and real networks (the first column). The best accuracy in each row is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Kuramoto	0.292 ± 0.050	0.651 ± 0.029	0.523 ± 0.046	0.026 ± 0.014	0.390 ± 0.054				
	Branch	0.418 ± 0.063	0.444 ± 0.070	0.467 ± 0.079	0.017 ± 0.012	0.339 ± 0.044	0.018 ± 0.011	0.017 ± 0.012	0.017 ± 0.012	0.010 ± 0.013
	CML	0.251 ± 0.038	0.410 ± 0.040	0.359 ± 0.050	0.002 ± 0.010	0.280 ± 0.039				
ER	Kuramoto	0.421 ± 0.051	0.723 ± 0.037	0.567 ± 0.046	0.003 ± 0.004	0.377 ± 0.036				
	Branch	0.235 ± 0.092	0.539 ± 0.064	0.308 ± 0.051	0.010 ± 0.006	0.244 ± 0.045	0.001 ± 0.002	0.004 ± 0.008	0.003 ± 0.008	0.002 ± 0.006
	CML	0.2623 ± 0.036	0.515 ± 0.057	0.442 ± 0.052	0.004 ± 0.005	0.231 ± 0.019				
WS	Kuramoto	0.598 ± 0.054	0.775 ± 0.074	0.233 ± 0.067	0.005 ± 0.003	0.350 ± 0.028				
	Branch	0.450 ± 0.049	0.550 ± 0.073	0.425 ± 0.074	0.004 ± 0.005	0.405 ± 0.037	0.014 ± 0.001	0.018 ± 0.003	0.018 ± 0.003	0.001 ± 0.001
	CML	0.303 ± 0.079	0.543 ± 0.070	0.473 ± 0.087	0.002 ± 0.006	0.275 ± 0.027				
Dolphins	Kuramoto	0.320 ± 0.076	0.400 ± 0.064	0.335 ± 0.041	0.067 ± 0.013	0.333 ± 0.027				
	Branch	0.267 ± 0.089	0.427 ± 0.053	0.360 ± 0.053	0.017 ± 0.016	0.507 ± 0.016	0.123 ± 0.025	0.127 ± 0.030	0.113 ± 0.024	0.033 ± 0.013
	CML	0.107 ± 0.034	0.227 ± 0.021	0.347 ± 0.023	0.020 ± 0.025	0.253 ± 0.027				
Word	Kuramoto	0.304 ± 0.074	0.405 ± 0.051	0.333 ± 0.036	0.012 ± 0.006	0.238 ± 0.012				
	Branch	0.113 ± 0.086	0.357 ± 0.063	0.286 ± 0.065	0.012 ± 0.010	0.262 ± 0.046	0.073 ± 0.031	0.073 ± 0.029	0.064 ± 0.027	0.105 ± 0.037
	CML	0.333 ± 0.075	0.476 ± 0.067	0.455 ± 0.027	0.024 ± 0.009	0.202 ± 0.060				
Ca-netsci	Kuramoto	0.330 ± 0.064	0.411 ± 0.037	0.425 ± 0.043	0.006 ± 0.005	0.352 ± 0.025				
	Branch	0.256 ± 0.080	0.539 ± 0.052	0.459 ± 0.076	0.007 ± 0.005	0.306 ± 0.022	0.363 ± 0.042	0.569 ± 0.042	0.543 ± 0.043	0.112 ± 0.034
	CML	0.211 ± 0.027	0.266 ± 0.018	0.322 ± 0.060	0.004 ± 0.005	0.311 ± 0.058				
Email	Kuramoto	0.241 ± 0.019	0.328 ± 0.013	0.264 ± 0.020	0.001 ± 0.002	0.267 ± 0.040				
	Branch	0.212 ± 0.052	0.332 ± 0.009	0.346 ± 0.010	0.003 ± 0.003	0.169 ± 0.044	0.140 ± 0.009	0.143 ± 0.005	0.153 ± 0.008	0.060 ± 0.009
	CML	0.233 ± 0.024	0.290 ± 0.001	0.347 ± 0.022	0.003 ± 0.006	0.114 ± 0.065				

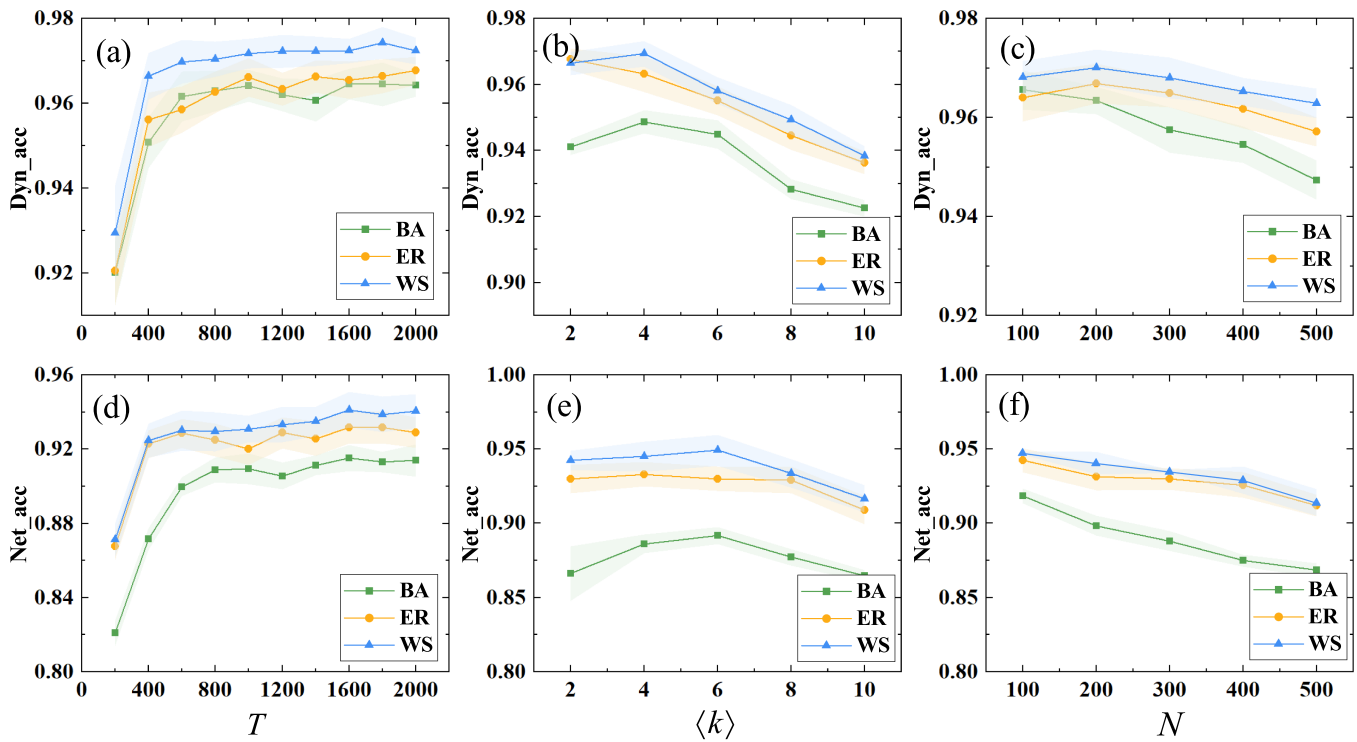


FIG. 3. Impact analysis of parameters regarding the CoND framework to data amount, average degree, and network size. Threshold dynamics on BA, WS, and ER networks are used. (a) and (d) Effects of time-series length on network inference and dynamical prediction, where the length varies from 200 to 2000 at the interval 200. The network size is $N = 200$ and the average degree $\langle k \rangle = 4$. (b) and (e) Effects of $\langle k \rangle$ varying from 2 to 10 (at the interval 2) on performance. The network size is $N = 200$ and the length of time series is $T = 1000$. (c) and (f) Impacts of the network size varying from 100 to 500 (at the interval 100) on prediction/inference performance, for $\langle k \rangle = 4$ and $T = 1000$. Here, the shaded translucent area represents the standard deviation of the results.

evaluation metric,

$$\text{Dyn_acc} = \sum_{i \in V_T} \frac{I(y^i = \hat{y}^i)}{|V_T|}, \quad (10)$$

where $I(\cdot)$ is an indicator function [where $I(y^i = \hat{y}^i) = 1$ if $(y^i = \hat{y}^i)$ is true and $I(y^i = \hat{y}^i) = 0$ otherwise], and $|V_T|$ is the number of nodes in the test set. When considering the continuous dynamical models, we use the mean square error loss (Dyn_mse) as the evaluation metric,

$$\text{Dyn_mse} = \frac{1}{|V_T|} \sum_{i \in V_T} (y^i - \hat{y}^i)^2. \quad (11)$$

As for the network inference task, the metric Net_acc is defined to measure how many top- L predicted edges are the real existing edges, that is,

$$\text{Net_acc} = \frac{c}{L}, \quad (12)$$

where c represents the number of real existing edges.

E. Experimental setup

To deal with incomplete structural information and incomplete dynamics, we consider these scenarios: some information about the network structure is lost during data collection and some time-series data of some nodes are lost as well. The experiments are conducted by removing a fixed fraction of the edges and the time-series data of a fraction of nodes. In our simulations, all time series have 1000 steps. Each time series is divided into a training set and a test set with the ratio 9:1. To reduce the statistical uncertainties, the values of these measures are averaged over ten independent realizations.

We test two settings: (a) 90% structure and 90% dynamics (i.e., data from a random 10% of the nodes are removed and (b) 80% structure and 80% dynamics (corresponding to the removal of the data from a random 20% of the nodes). To present our results concisely and clearly, we include only the results with the GCN model for the case of 90% structure and 90% dynamics in the main text.

F. Simulation results

Tables II and III show the comparison among different methods for discrete and continuous network dynamics, respectively, for the task of dynamical prediction. The first observation is that the accuracy of our CoND framework is typically higher than that of

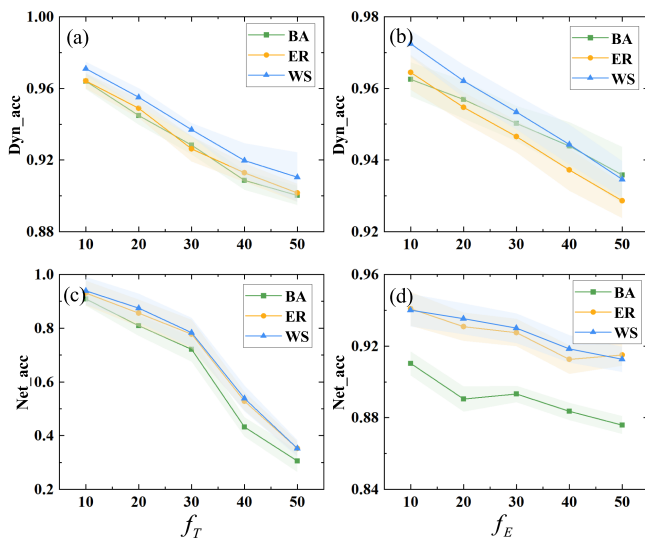


FIG. 4. Further support for the mutually beneficial roles of the two prediction tasks. Threshold dynamics on BA, WS, and ER networks are studied. (a) and (c) Effects of the fraction f_T of removed time-series data, ranging from 10% to 50% at the interval of 10%, on network inference and dynamical prediction. The fraction f_E of missing edges is fixed at 10%. (b) and (d) Effects of f_E , ranging from 10% to 50% at an interval of 10%, on the performance of the two prediction tasks, for fixed $f_T = 10\%$. In all cases, the simulation parameters are $\langle k \rangle = 4$, $N = 200$, and $T = 1000$. Here, the shaded translucent area represents the standard deviation of the results.

CoND₀. The reason is that CoND₀ does not include a state evaluation module, so it is unable to recover the states of the missing nodes, while the CoND framework not only can recover the states of unobserved nodes but also better learn the network structure. Note that CoND₀ outperforms GGN. The second observation is that CoD₁ generally outperforms CoD₂ for both discrete and continuous dynamics, again because of the inclusion of the state evaluation module in CoD₁. Overall, if exact state information is known, the network dynamics can be better predicted.

Another observation is that the accuracies of CoND₀ and CoND are generally higher than those of CoD₁ and CoD₂, especially for discrete dynamics, suggesting that iteratively updating the network structure in the network inference module is beneficial to improving the accuracy of dynamical prediction. However, CoD₁ and CoD₂ can slightly outperform CoND and CoND₀ in some cases, e.g., continuous dynamics that are somewhat more difficult to be learned than discrete dynamics. In fact, because CoD₁ and CoD₂ focus on the task of dynamical prediction without performing network inference, they are more adaptable to this prediction task. On the contrary, CoND and CoND₀ need to balance the two tasks simultaneously, causing a reduction in the prediction accuracy.

We now compare the accuracies of different methods for the network inference task. As listed in Tables IV and V, CoND and CoND₀ outperform the GGN method for both continuous and discrete dynamics. A further comparison of CoND and CoND₀ indicates that, in most cases, the former generally outperforms the latter in inferring the network structure. Considering that CoND₀ also has

lower performance in predicting the dynamics, it is inferior to the CoND framework, suggesting the necessity to incorporate a state estimation module into the machine-learning framework for general tasks requiring predicting both the network structure and the network dynamics.

Tables IV and V also summarize the results of comparing our CoND framework with two classical network structure inference methods: the correlation and Granger causality methods (note that the classical methods cannot be used to predict the dynamics). Because the available information (data) is assumed to be sparse, the correlation method is inferior, as caused by the low similarity between the time-series data of the nodal dynamics. The Granger method performs relatively well because it uses certain functions to reconstruct the network but the associated computational complexity is high, especially for large networks. In most cases, our CoND framework greatly outperforms both classical methods. Tables IV and V also present the performance of four existing link prediction algorithms, which is generally low because they only use the observed network structure without considering the observed dynamics data.

Taken together, the results in Tables II–V demonstrate that our CoND framework is capable of not only accurately predicting the dynamics but also inferring the precise network structures and is, thus, far superior to any existing method. A key feature of our method is that the tasks of dynamical prediction and network inference are mutually reinforcing each other: network inference improves dynamical prediction and vice versa.

Figure 3 demonstrates the robustness of our CoND framework by illustrating the impacts of the network size N , the average degree $\langle k \rangle$, and the length T of time-series data for the prediction performance of the two tasks on BA, WS, and ER networks. For concreteness, we consider the threshold dynamics model on three synthetic networks. Figures 3(a) and 3(d) reveal that increasing T can improve the accuracy of network inference as well as the dynamical prediction. Figures 3(b) and 3(e) display the impacts of $\langle k \rangle$ on the prediction performance, and the effects of N are shown in Figs. 3(c) and 3(f), revealing that the prediction performance decreases with $\langle k \rangle$ and N . The reason is that predicting more densely connected and larger networks requires more data. If the data amount is fixed, the prediction accuracy will be sacrificed (the details of the other dynamical processes are provided in Appendix C).

To strengthen the correlation between network inference and dynamical prediction, we conduct an impact analysis of parameters with respect to incomplete time-series data and missing edges on BA, WS, and ER networks. We first assume that 10% of the edges are not observed and study the effects of incomplete data. Figures 4(a) and 4(c) show that incomplete dynamical data have a dramatic influence on network inference. As the available time series become shorter, the accuracy of both tasks decreases significantly. We then assume that 10% of the nodal time series cannot be observed. Figures 4(b) and 4(d) show that the performance of our CoND framework decreases but slowly as the fraction of removed edges increases. Since only a small fraction of the time-series data are removed, most information about the network structure and dynamics is still contained in the available time-series data. As a result, the network structure can be recovered to a great extent even if more edges are deleted, meaning that the dynamics can

TABLE VI. Performance comparison of CoND and other baseline methods for network inference and dynamical prediction as applied to the influenza data in US States and the PM_{2.5} data in Chinese cities.

Dynamical prediction									
Dataset	GGN	CoND	CoND ₀	CoD ₁	CoD ₂	\	\	\	\
Influenza	0.009 ± 0.001	0.007 ± 0.000	0.008 ± 0.000	0.008 ± 0.000	0.008 ± 0.000	\	\	\	\
PM _{2.5}	0.005 ± 0.000	0.003 ± 0.001	0.002 ± 0.001	0.003 ± 0.000	0.003 ± 0.000	\	\	\	\
Network inference									
Dataset	GGN	CoND	CoND ₀	Correlation	Granger	CN	RA	AA	LP
Influenza	0.241 ± 0.035	0.355 ± 0.038	0.296 ± 0.054	0.130 ± 0.045	0.259 ± 0.057	0.291 ± 0.038	0.292 ± 0.044	0.273 ± 0.044	0.301 ± 0.042
PM _{2.5}	0.035 ± 0.028	0.257 ± 0.039	0.324 ± 0.043	0.007 ± 0.002	0.128 ± 0.047	0.130 ± 0.036	0.142 ± 0.031	0.151 ± 0.045	0.082 ± 0.031

be predicted even if many edges are missing. The results from these two experiments provide further support for the mutually beneficial roles of the two prediction tasks: a better-estimated network structure improves the accuracy of dynamical prediction and better-estimated dynamical rules improve the accuracy of network inference.

Finally, we validate the effectiveness of our CoND framework using two real-world datasets: an influenza⁵⁹ and a PM_{2.5}⁶⁰ dataset. The influenza dataset is the weekly influenza data reported by US public health and clinical laboratories for the period from week 40 of 2011 to week 39 of 2016. We use influenza-like illness (ILI) to measure the influenza outbreaks and normalize weekly ILI-related visits in 37 US States to calculate the ILI ratio. (The average weekly ILI ratio of the other 14 States is less than 1%, so they are removed from the analysis to ensure a sufficient sample size in the data) by recording the weekly ILI ratio so that we can obtain time-series data for each state. Since small values of the ILI ratio can lead to large biases in the prediction process, we multiply the ILI ratio by a factor of 100 and use the resulting values as dynamic data. The commuting network between the US States is generated using the commuting traffic data from different States in 2015. The PM_{2.5} dataset is a complete 4-year (January 1, 2015 to December 31, 2018) PM_{2.5} dataset, which selects a wide range of areas (longitude 103°E–122°E and latitude 28°E–42°E), covering a total of 184 cities (nodes) in China. We construct the city network using the latitude, longitude, and altitude information, and the PM_{2.5} data are the dynamical time-series data (a detailed description of the two real datasets is provided in Appendix D).

Table VI compares the accuracy of prediction/inference tasks for 90% of the network structure and 90% of the dynamics data. There are several findings. First, for both real-world datasets, the performance of CoND and CoND₀ for the two tasks is better than that of the GGN and the traditional approaches. CoND outperforms CoND₀ for the influenza dataset, but the opposite is observed for the PM_{2.5} dataset, with insignificant differences. For network inference, the value of Net_acc is low for CoND and CoND₀: about 0.25–0.35. The main reason is that the true structures of the influenza transmission and city networks are not known: they are reconstructed based on limited empirical knowledge (e.g., commuting data or geographic location information). Nonetheless, the networks inferred by the CoND or CoND₀ method may help us discover some potential relationships among the nodes in the respective applications (more results from the setting of 80%structure and 80%

dynamics based on the GCN model, as well as the results from the GIN and SAGE models are given in Appendix E).

V. CONCLUSION AND DISCUSSIONS

In spite of the large literature, the problem of data-driven prediction of complex networks subject to severely missing data and edges has been longstanding. This work develops a data-driven adaptive collaborative learning framework (CoND) to address this challenge. Our CoND assumption only allows for the observation of a small group of observed edges (existing edges) and the missing time-series data of nodes can be independent of the data of network structure, thus representing a more general framework for solving the structural and dynamic inverse problems of networks with limited available information/data. Our machine-learning framework is capable of accurately capturing the complex interplay between the network structure and dynamics, providing a bridge between network inference and dynamical prediction. The CoND framework has been validated for various discrete and continuous dynamics on three synthetic and four empirical networks and has been shown to be superior to the currently available baseline methods. The CoND framework has also been tested on two real-world datasets: an influenza⁵⁹ and a PM_{2.5}⁶⁰ dataset.

To further assess the advantages and limitations of the collaborative learning task and understand its boundary of applicability, we have conducted experiments with varying data length, average degree, and network size. The results show that, while these factors can affect the performance of CoND to some extent, the framework is effective. Simulations have also been carried out on how the degrees of missing data and edges affect the performance, revealing that information about the network dynamics is more important to the success of the prediction/inference tasks than that about the network structure. The CoND framework is compatible with different GNNs and is applicable to a wide range of scenarios. In fact, good knowledge of the network structure can enhance the performance of dynamical prediction and vice versa.

Although our model has achieved good performance in network inference and dynamical prediction, there is much room for improvement. First, as the complexity of real systems continues to increase, network structures also become more complex and diverse, including multilayer networks,⁵¹ temporal networks,⁶² and higher-order networks.⁶³ Then, the network size N may be an unknown variable, making the inference and prediction tasks for the size of

the unknown network a challenging problem. Finally, our deep-learning-based framework general performs well in many scenarios, but its interpretability is not well solved. All these mentioned issues are worthy of in-depth research.

ACKNOWLEDGMENTS

X.D. and H.-F.Z. are supported by the National Natural Science Foundation of China (NNSFC) (No. 61973001) and the University Synergy Innovation Program of Anhui Province (No. GXXT-2021-032). L.-W.K. and Y.-C.L. are supported by AFOSR under Grant No. FA9550-21-1-0438.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Xiao Ding: Formal analysis (equal); Investigation (equal); Methodology (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Ling-Wei Kong:** Formal analysis (equal); Investigation (equal); Methodology (equal); Writing – original draft (equal). **Hai-Feng Zhang:** Conceptualization (equal); Formal analysis (equal); Funding acquisition (equal); Methodology (equal); Writing – original draft (equal); Writing – review & editing (equal). **Ying-Cheng Lai:** Formal analysis (equal); Investigation (equal); Methodology (equal); Writing – original draft (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: DESCRIPTION OF THREE TYPES OF GNN MODELS IN THIS WORK

In this section, we briefly introduce the three types of GNN models used in this paper.

1. GCN model

Given the graph signal (i.e., feature) X and the graph adjacency matrix A , Kipf and Welling⁴⁰ proposed the classical graph convolutional network (GCN) as follows:

$$H^{(k+1)} = \sigma(\tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2} H^{(k)} \theta^{(k)}), \quad (\text{A1})$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the undirect graph G with self-loop, I_N is the identity matrix, \tilde{D} is the degree matrix of \tilde{A} , and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $\theta^{(k)}$ is the learnable parameter, and $\sigma(\cdot)$ denotes an activation function. $H^{(k)}$ is the k th layer activation matrix, and $H^{(0)} = X$.

2. SAGE model

In Graph SAmple and aggreGatE (SAGE) model,⁴¹ $AGGREGATE(\cdot)$ function and $CONCAT(\cdot)$ function are defined,

TABLE VII. The architectures of GCN, SAGE, and GIN models. S represents the number of states for a considered discrete dynamical model.

Model	Discrete dynamics	Continuous dynamics
Input	Linear(1,32) ReLU	Linear(1,32) ReLU
Aggregation	GCN(32,32)/ SAGE (32,32) / GIN(MLP(32,64,32)) ReLU	GCN(32,32)/ SAGE (32,32) / GIN(MLP(32,64,32)) ReLU
Output	Linear(32,S) Softmax	Linear(32,1) ReLU
Loss function	Cross-entropy loss	Mean square loss

and their roles are to aggregate neighbor's information and to concatenate the feature of the node itself and the aggregation features from its neighbors, respectively. For a given node v , its feature at k th iteration is defined as

$$h_{N(v)}^{(k)} = AGGREGATE^{(k)}(\{h_u^{(k-1)}\}, \forall u \in N(v)), \quad (\text{A2})$$

$$h_v^{(k)} = \sigma(\theta^{(k)} \cdot CONCAT^{(k)}(h_v^{(k-1)}, h_{N(v)}^{(k)})), \quad (\text{A3})$$

where $N(v)$ denotes the sampled neighbors of node v , $h_v^{(k)}$ represents the feature of node v at k th iteration (note that $h_v^{(0)} = x_v$), and $h_{N(v)}^{(k)}$ represents the aggregation feature from neighbors at k th iteration. Specifically, $AGGREGATE(\cdot)$ function is the mean aggregation and $CONCAT(\cdot)$ function is the vector concatenation.

3. GIN model

In the Graph Isomorphism Network (GIN) model,⁴² the sum aggregation is defined as Eq. (A2), and the MultiLayer Perceptron (MLP) is used to simulate the $CONCAT(\cdot)$ function. Then, the GIN layer for learning node representation is mathematically defined as

$$h_v^{(k)} = \sigma(MLP^{(k)}(1 + \theta^{(k)} h_v^{(k-1)} + \sum_{N(v)} h_u^{(k-1)})). \quad (\text{A4})$$

In this work, the detailed architectures of GCN, SAGE, and GIN models are summarized in Table VII. The input layer is the linear layer, the aggregation layer is the GCN, SAGE, or GIN, respectively, and the output layer is given by the linear layer too. For discrete dynamical models, the activation function employed is the Softmax function, and the loss function is the cross-entropy loss.⁶⁴ In contrast, for continuous dynamical models, the activation function used is the ReLU function, and the loss function is the mean squared error.⁶⁵

APPENDIX B: INTRODUCTION OF SOME BASELINE METHODS IN THIS WORK

In this section, we introduce baseline methods for network inference and dynamical prediction, which are used to compare with our proposed method.

1. GGN method

Gumbel Graph Network (GGN) is a data-driven deep-learning model, which can be used to solve the network completion and dynamical prediction problems.³⁶

2. CoND₀ method

The algorithm consists of a network inference module and a dynamical prediction module but without a state estimation module. The states of the unobserved nodes are initialized randomly.

3. CoD₁ method

The method uses the dynamical learning module and the state estimation module to predict the dynamics, where the input graph has a partial network structure characterized by the adjacency matrix A_o .

4. CoD₂ method

This method contains a dynamical learning module for dynamical prediction, where the input graph has a partial network structure characterized by the adjacency matrix A_o . The states of the unobserved nodes are randomly initialized.

5. Correlation coefficient method

The correlation coefficient method¹¹ measures the connection possibility of node pairs by calculating the time-series correlation of them. For the time series x_i and x_j of two nodes, the element \hat{a}_{ij} of correlation matrix is expressed as

$$\hat{a}_{ij} = \left| \frac{(x_i - 1\mu_i)^T(x_j - 1\mu_j)}{\sigma_i\sigma_j} \right|, \quad (\text{B1})$$

where μ_i and σ_i are the mean value and the standard deviation of x_i , respectively.

6. Granger causality method

In the Granger causality method,⁵⁸ the element \hat{a}_{ij} of the adjacency matrix is estimated as the ratio of the standard deviation of the prediction error of linear models $F(\cdot)$ trained solely with x_i , to the standard deviation of the prediction error of the linear models $G(\cdot)$ trained with x_i and x_j ,

$$\hat{a}_{ij} = \frac{\text{std}[x_i - F(x_i)]}{\text{std}[x_i - G(x_i, x_j)]}. \quad (\text{B2})$$

7. Common neighbor (CN) index

CN index²⁰ is one of the simplest similarity indexes based on the local structure of networks. This index denotes that the connection probability of a pair of nodes (i, j) is higher if they share more common neighbors, thus

$$\hat{a}_{ij} = |\Gamma(i) \cap \Gamma(j)|, \quad (\text{B3})$$

where $\Gamma(i)$ is the neighborhood set of node i .

8. Adamic-Adar (AA) index

The AA index²¹ suggests that the fewer connected neighbors should contribute more weights to the similarity, which is expressed as

$$\hat{a}_{ij} = \sum_{z \in |\Gamma(i) \cap \Gamma(j)|} \frac{1}{\log k_z}, \quad (\text{B4})$$

where k_z is the degree of z .

9. Resource allocation (RA) index

This index is proposed by Zhou *et al.*,⁵⁶ which assumes that each common neighbor z has one unit resource and evenly allocates the resource to its neighbors, and it is defined as

$$\hat{a}_{ij} = \sum_{z \in |\Gamma(i) \cap \Gamma(j)|} \frac{1}{k_z}. \quad (\text{B5})$$

10. Local path (LP) index

LP index⁵⁷ provides a good trade-off of accuracy and computational complexity by taking account of the different contributions of nodes in local paths. For a node pair (i, j) , the LP index is defined as

$$\hat{a}_{ij} = (A^2)_{ij} + \alpha(A^3)_{ij}, \quad (\text{B6})$$

where the parameter α is used to adjust the contribution of neighbors with three-length. $(A^l)_{ij}$ ($l = 2$ or 3) gives the number of paths with l -length between nodes i and j . Here, α is set as 1.

Since the values of \hat{a}_{ij} calculated by Eqs. (B1)–(B6) may be greater than 1, as a result, each element \hat{a}_{ij} in \hat{A} is normalized by dividing by the maximum value in \hat{A} .

APPENDIX C: MORE IMPACT ANALYSIS OF PARAMETERS FOR DIFFERENT SCENARIOS

In this section, the impact analysis of parameters for different scenarios on three synthetic networks, i.e., BA, WS, and ER networks, is discussed, including the length T of time-series data, the average degree $\langle k \rangle$ and the network size N , the fraction f_T of removed time-series data, and the fraction f_E of missing edges.

1. The length of time-series data

The effects of the length of the time-series data on the dynamical prediction as well as network inference are demonstrated in Fig. 5. One can find that, with an increase in length T of time-series data, the accuracies of dynamical prediction and network inference are both improved significantly, and the accuracy of the model tends to be stable when the amount of data is sufficient.

2. The average degree

The effects of the average degree on the dynamical prediction as well as network inference are shown in Fig. 6. On the one hand, since the network structure becomes complicate when the average degree of networks is increased, giving rise to that the accuracy of dynamical prediction decreases with the average degree $\langle k \rangle$. On the other hand, the accuracy of network inference is decreased if the network is too sparse or too dense.

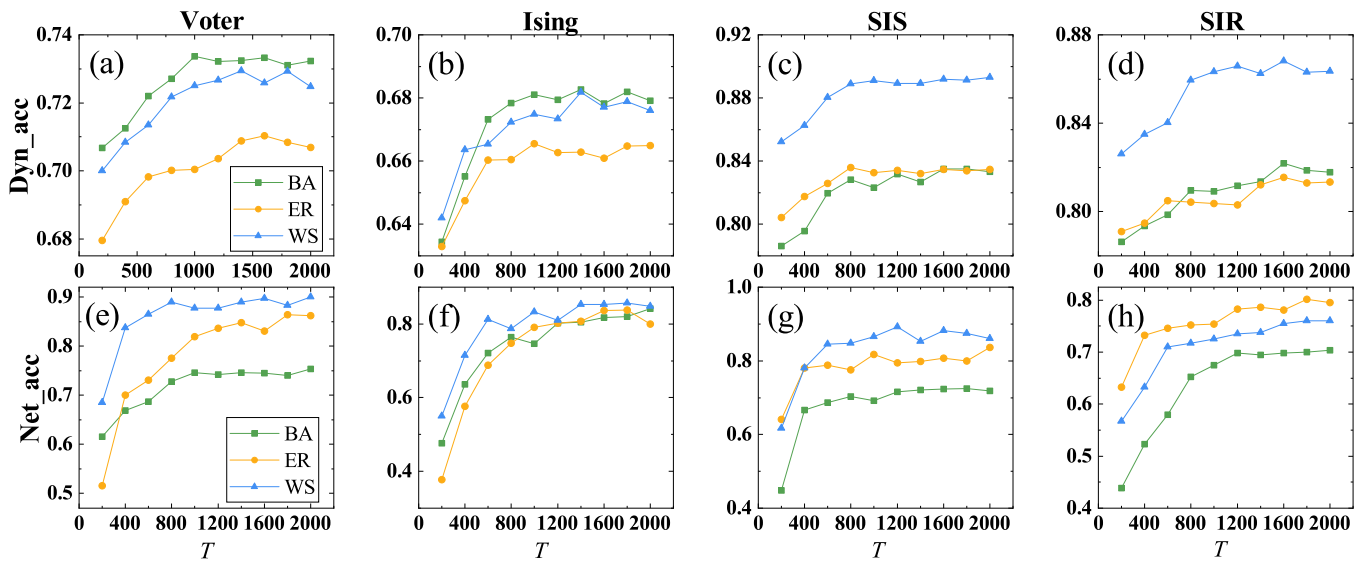


FIG. 5. Effects of the length of time-series data on dynamical prediction and network inference on BA, WS, and ER networks. (a) and (e) Voter model. (b) and (f) Ising model. (c) and (g) SIS model. (d) and (h) SIR model. Here, $N = 200$, $\langle k \rangle = 4$, $f_T = 10\%$, and $f_E = 10\%$.

3. The network size

The effects of the network size on the dynamical prediction as well as network inference are illustrated in Fig. 7. Evidently, when the length of time-series data is fixed, the performance of the model decreases with the network size.

4. The fraction of removed time-series data

The effects of the fraction of removed time-series data on the dynamical prediction as well as network inference are given in Fig. 8. In general, the performance of the two tasks decreases with the fraction of removed time-series data.

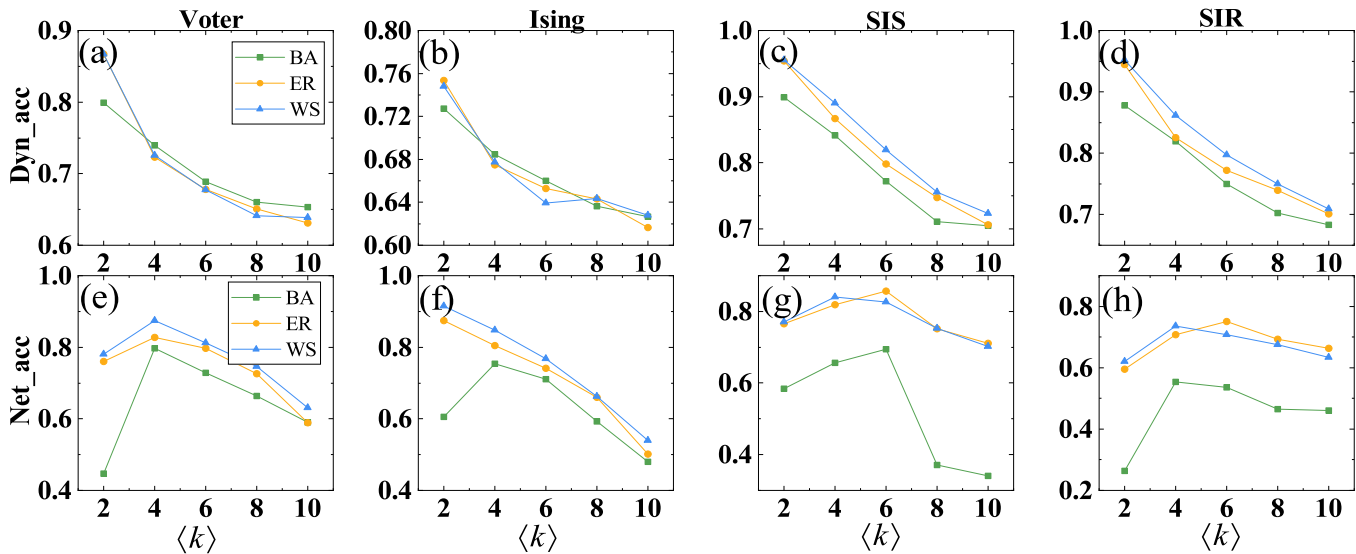


FIG. 6. Effects of the average degree on dynamical prediction and network inference on BA, WS, and ER networks. (a) and (e) Voter model. (b) and (f) Ising model. (c) and (g) SIS model. (d) and (h) SIR model. Here, $N = 200$, $T = 1000$, $f_T = 10\%$, and $f_E = 10\%$.

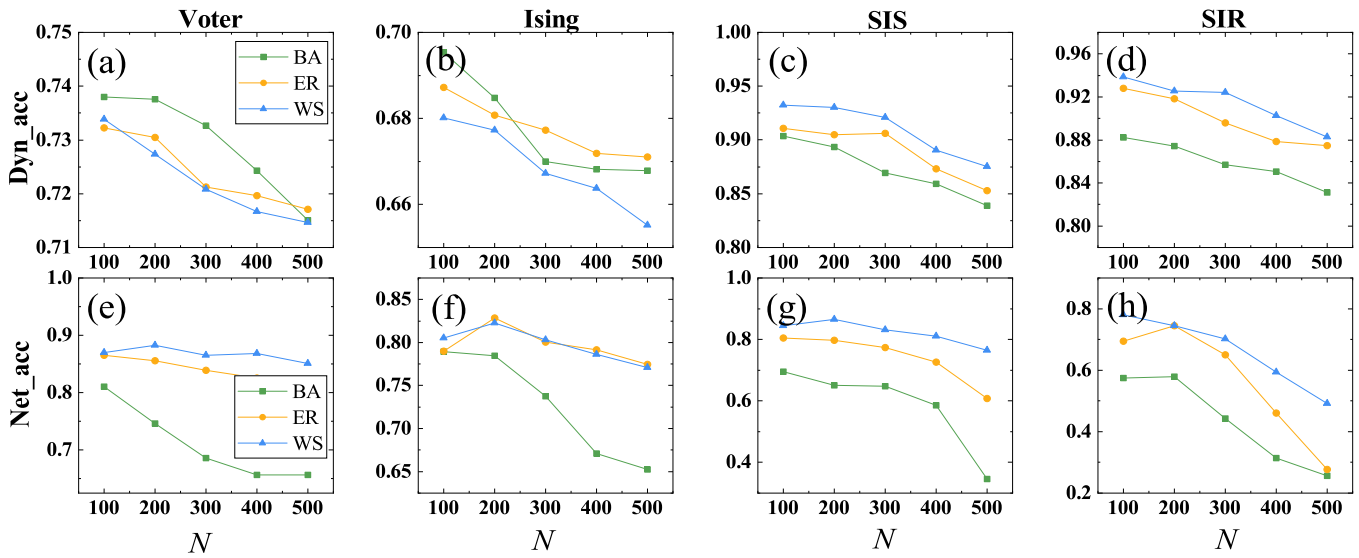


FIG. 7. Effects of the network size on dynamical prediction and network inference on BA, WS, and ER networks. (a) and (e) Voter model. (b) and (f) Ising model. (c) and (g) SIS model. (d) and (h) SIR model. Here, $\langle k \rangle = 4$, $T = 1000$, $f_T = 10\%$, and $f_E = 10\%$.

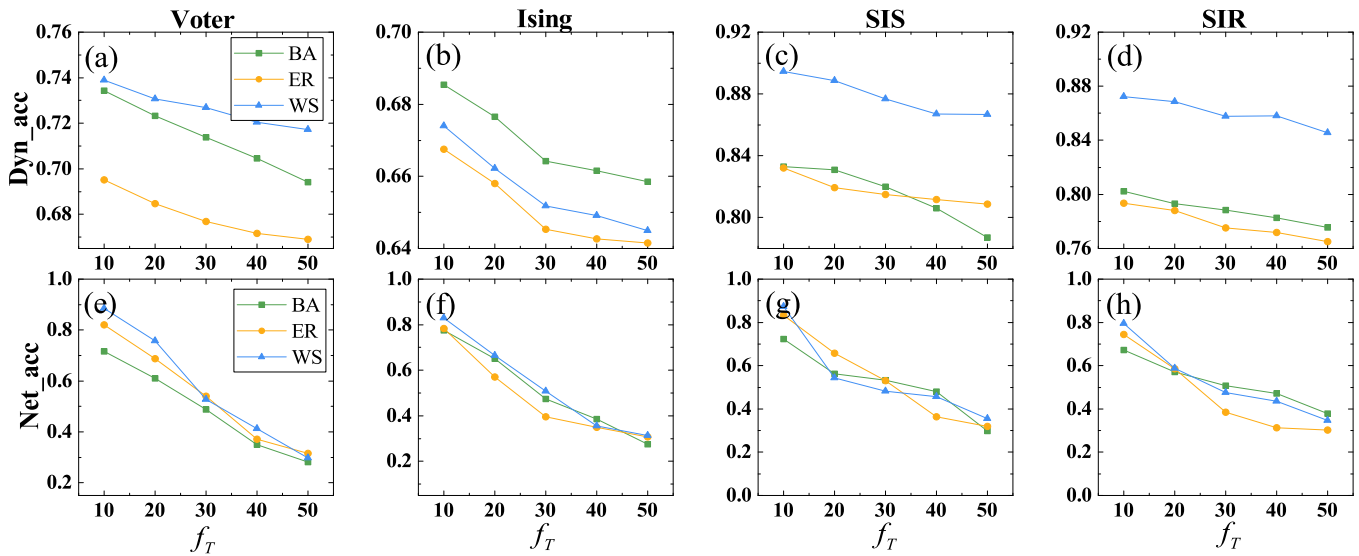


FIG. 8. Effects of the fraction of removed time-series data on dynamical prediction and network inference on BA, WS, and ER networks. (a) and (e) Voter model. (b) and (f) Ising model. (c) and (g) SIS model. (d) and (h) SIR model. Here, $N = 200$, $\langle k \rangle = 4$, $T = 1000$, and $f_E = 10\%$.

05 April 2024 00:35:44

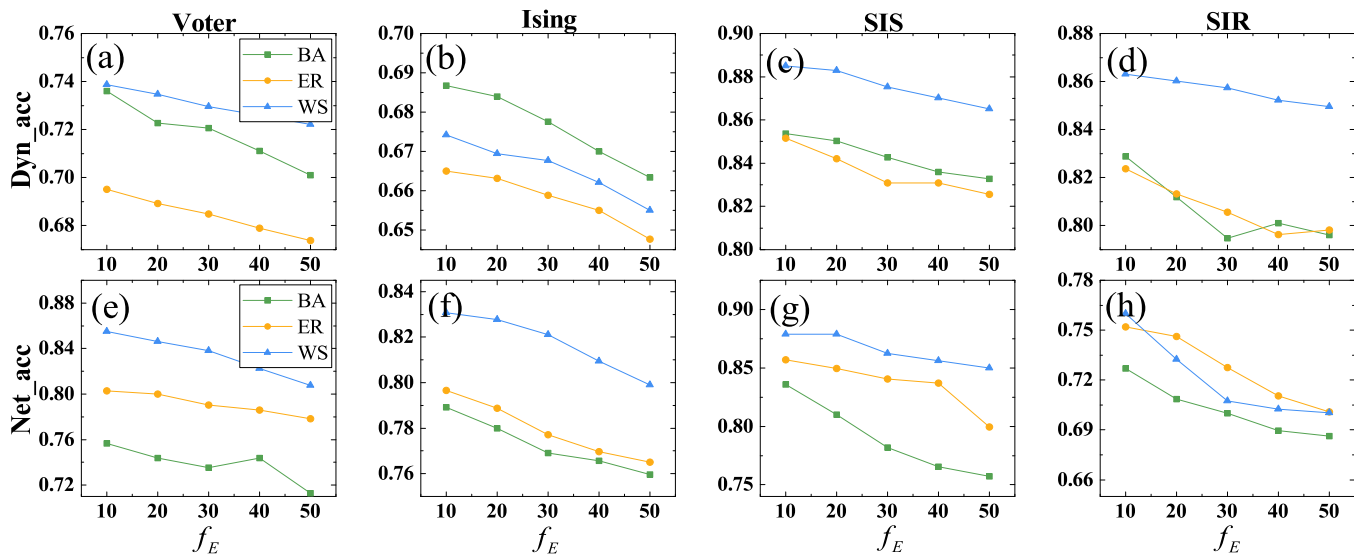


FIG. 9. Effects of the fraction of missing edges on dynamical prediction and network inference on BA, WS, and ER networks. (a) and (e) Voter model. (b) and (f) Ising model. (c) and (g) SIS model. (d) and (h) SIR model. Here, $N = 200$, $\langle k \rangle = 4$, $T = 1000$, and $f_T = 10\%$.

5. The fraction of missing edges in networks

The effects of the fraction of missing edges on the dynamical prediction as well as network inference are given in Fig. 9. Also, the performance of the two tasks decreases with the fraction of missing edges in networks.

APPENDIX D: DATA DESCRIPTION OF THE INFLUENZA DATA AND PM_{2.5} DATA

In this section, we introduce the influenza data and PM_{2.5} data.

1. Influenza data

The influenza-like illness (ILI) data are obtained from the US public health and clinical laboratory⁶⁶ for the period from week from week 40 of 2011 to week 39 of 2016. We standardized the ILI data using the number of patients seen in each US State to calculate ILI rates.⁵⁹ To ensure that there is an adequate sample size in the data, we assume that States with an average ILI ratio of less than 1% are not representative (14 States were removed) and then only remaining 37 States as nodes. Since small values of the ILI ratio (about 2%) may lead to errors, we multiply the ILI ratio by the factor of 100 and use the resulting value for the dynamical data.³⁰ In addition, we use commuting data between US cities (2015 Census Report⁶⁷) to construct the commuting network. The network is undirected, where nodes represent the US States and edges are determined by the data for commuting between the States. To prevent the network from being too dense, we use the criterion that if the number of commuters between two States is less than 100, the corresponding edges are removed, and, finally, the commuting network is obtained.

2. PM_{2.5} data

We use a complete 4-year (January 1, 2015 to December 31, 2018) data for the fine particulate matter smaller than 2.5 μm in diameter (PM_{2.5}) as dynamical data, which are selected from a wide range of regions (the longitude 103°E-122°E and the latitude 28°E-42°E) covering a total of 184 cities in China.⁶⁸ To construct the city network for the diffusion of PM_{2.5} pollutants, we need to calculate the correlation of nodes (i.e., cities). To do so, we define the geographic distance between each node determined by the latitude and the longitude. Meanwhile, we consider the mountain ranges in two cities that can impede the diffusion of the PM_{2.5} pollutants. Based on these metrics, we construct the adjacency matrix according to the following method.⁶⁰

$$a_{ij} = I(d_{ij} - d_{\theta}) \cdot I(l_{ij} - l_{\theta}), \quad (\text{D1})$$

$$d_{ij} = \|\rho_i - \rho_j\|, \quad (\text{D2})$$

$$l_{ij} = \sup_{\lambda \in (0,1)} \{h(\lambda\rho_i + (1-\lambda)\rho_j) - \max\{h(\rho_i), h(\rho_j)\}\}, \quad (\text{D3})$$

where ρ_i denotes the position determined by the longitude and latitude of node i , and $h(\rho_i)$ is the altitude of node i . $I(t)$ is a Heaviside step function, $I(t) = 1$ when $t > 0$, otherwise its value is 0. d_{θ} and l_{θ} are the distance and altitude thresholds, respectively. In this paper, we set $d_{\theta} = 170$ km as the distance threshold and $l_{\theta} = 1200$ m as the altitude threshold. In particular, PM_{2.5} pollutants can be transmitted from one city to the other cities only when the distance between two cities is less than 170 km and the altitude between them is less than 1200 m.

APPENDIX E: MORE RESULTS UNDER DIFFERENT SETTINGS AND GNN MODELS

In this section, more results under different settings are given to validate the universality of our CoND framework. These results aim to emphasize the focus on the good scalability of our model.

First, more results for the case of 80% dynamics (i.e., 20% time-series data are randomly removed) and 80% structure (i.e., 20% edges are randomly removed) are systematically studied, where the GCN model is considered in the dynamical learning module. Tables VIII and IX show the accuracy of dynamical prediction. Tables X and XI show the accuracy of network inference. It can be found that our method is still superior to the baseline methods for this case.

Then, we perform network inference and dynamical prediction tasks by considering the SAGE (GIN) model in the dynamical

learning module containing 90% dynamics and 90% structure. Tables XII and XIII (Tables XVI and XVII) summarize the accuracy of dynamical prediction, and Tables XIV and XV (Tables XVIII and XIX) show the accuracy of network inference. All results indicate that the performances of the CoND framework in terms of network inference and dynamical prediction are superior to the baseline methods. These results imply that our CoND framework is compatible with different GNN models.

Finally, we conduct additional experiments on the PM_{2.5} and influenza datasets by replacing the GCN model with GIN and SAGE models with 90% dynamics and 90% structure is considered. Tables XX and XXI show the results for the GIN model and the SAGE model, respectively. The results in both tables again highlight the superior performance of our CoND framework in term of dynamical prediction and network inference even for real cases.

TABLE VIII. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_acc metric for five discrete dynamical models, with the GCN model containing 80% dynamics and 80% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Threshold	0.717 ± 0.004	0.934 ± 0.003	0.932 ± 0.004	0.885 ± 0.005	0.875 ± 0.005
	Voter	0.563 ± 0.006	0.699 ± 0.005	0.682 ± 0.004	0.647 ± 0.007	0.647 ± 0.006
	Ising	0.572 ± 0.005	0.651 ± 0.005	0.660 ± 0.002	0.634 ± 0.007	0.634 ± 0.007
	SIS	0.800 ± 0.002	0.841 ± 0.002	0.831 ± 0.003	0.830 ± 0.002	0.829 ± 0.002
	SIR	0.775 ± 0.010	0.792 ± 0.003	0.783 ± 0.002	0.785 ± 0.006	0.780 ± 0.005
ER	Threshold	0.717 ± 0.004	0.944 ± 0.003	0.932 ± 0.002	0.877 ± 0.003	0.875 ± 0.002
	Voter	0.563 ± 0.003	0.679 ± 0.004	0.682 ± 0.005	0.647 ± 0.002	0.647 ± 0.002
	Ising	0.572 ± 0.005	0.671 ± 0.002	0.660 ± 0.002	0.634 ± 0.004	0.634 ± 0.004
	SIS	0.800 ± 0.001	0.841 ± 0.007	0.831 ± 0.002	0.828 ± 0.002	0.829 ± 0.002
	SIR	0.775 ± 0.003	0.792 ± 0.001	0.783 ± 0.002	0.775 ± 0.004	0.770 ± 0.003
WS	Threshold	0.719 ± 0.005	0.945 ± 0.003	0.942 ± 0.003	0.886 ± 0.003	0.886 ± 0.004
	Voter	0.600 ± 0.002	0.727 ± 0.002	0.732 ± 0.001	0.714 ± 0.005	0.704 ± 0.005
	Ising	0.532 ± 0.002	0.655 ± 0.005	0.667 ± 0.003	0.639 ± 0.003	0.639 ± 0.002
	SIS	0.797 ± 0.004	0.901 ± 0.003	0.893 ± 0.002	0.880 ± 0.001	0.873 ± 0.002
	SIR	0.802 ± 0.001	0.879 ± 0.005	0.868 ± 0.001	0.836 ± 0.007	0.825 ± 0.004
Dolphins	Threshold	0.810 ± 0.002	0.935 ± 0.006	0.929 ± 0.005	0.864 ± 0.008	0.855 ± 0.008
	Voter	0.615 ± 0.003	0.725 ± 0.002	0.732 ± 0.011	0.671 ± 0.006	0.670 ± 0.006
	Ising	0.573 ± 0.003	0.687 ± 0.001	0.684 ± 0.005	0.647 ± 0.007	0.637 ± 0.006
	SIS	0.659 ± 0.004	0.724 ± 0.001	0.732 ± 0.002	0.721 ± 0.006	0.723 ± 0.006
	SIR	0.633 ± 0.004	0.788 ± 0.006	0.776 ± 0.003	0.762 ± 0.004	0.759 ± 0.006
Word	Threshold	0.8138 ± 0.004	0.938 ± 0.003	0.928 ± 0.004	0.859 ± 0.006	0.859 ± 0.006
	Voter	0.576 ± 0.002	0.664 ± 0.002	0.660 ± 0.004	0.621 ± 0.004	0.621 ± 0.005
	Ising	0.578 ± 0.006	0.633 ± 0.006	0.634 ± 0.003	0.601 ± 0.003	0.600 ± 0.003
	SIS	0.614 ± 0.003	0.760 ± 0.002	0.755 ± 0.002	0.707 ± 0.009	0.722 ± 0.001
	SIR	0.574 ± 0.003	0.699 ± 0.004	0.708 ± 0.002	0.697 ± 0.007	0.675 ± 0.007
Ca-netsci	Threshold	0.673 ± 0.004	0.931 ± 0.007	0.929 ± 0.003	0.902 ± 0.003	0.902 ± 0.003
	Voter	0.587 ± 0.003	0.748 ± 0.003	0.736 ± 0.002	0.718 ± 0.002	0.718 ± 0.003
	Ising	0.540 ± 0.002	0.683 ± 0.004	0.691 ± 0.003	0.657 ± 0.001	0.657 ± 0.002
	SIS	0.756 ± 0.010	0.901 ± 0.003	0.913 ± 0.002	0.921 ± 0.003	0.919 ± 0.003
	SIR	0.795 ± 0.003	0.894 ± 0.002	0.884 ± 0.002	0.857 ± 0.002	0.846 ± 0.002
Email	Threshold	0.668 ± 0.001	0.942 ± 0.006	0.930 ± 0.003	0.903 ± 0.003	0.903 ± 0.002
	Voter	0.592 ± 0.003	0.748 ± 0.002	0.736 ± 0.002	0.718 ± 0.002	0.719 ± 0.002
	Ising	0.537 ± 0.005	0.683 ± 0.004	0.691 ± 0.003	0.656 ± 0.002	0.656 ± 0.002
	SIS	0.783 ± 0.001	0.881 ± 0.003	0.855 ± 0.002	0.831 ± 0.002	0.828 ± 0.003
	SIR	0.764 ± 0.003	0.879 ± 0.002	0.874 ± 0.001	0.856 ± 0.002	0.856 ± 0.003

TABLE IX. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_mse metric for three continuous dynamical models, with the GCN model containing 80% dynamics and 80% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Kuramoto	0.051 ± 0.004	0.042 ± 0.001	0.044 ± 0.003	0.052 ± 0.002	0.053 ± 0.004
	Branch	0.091 ± 0.004	0.040 ± 0.003	0.031 ± 0.002	0.044 ± 0.005	0.044 ± 0.003
	CML	0.067 ± 0.005	0.044 ± 0.005	0.046 ± 0.004	0.041 ± 0.003	0.046 ± 0.004
ER	Kuramoto	0.073 ± 0.003	0.055 ± 0.003	0.058 ± 0.003	0.065 ± 0.003	0.066 ± 0.002
	Branch	0.051 ± 0.002	0.024 ± 0.004	0.025 ± 0.005	0.026 ± 0.006	0.027 ± 0.003
	CML	0.091 ± 0.004	0.066 ± 0.003	0.075 ± 0.002	0.063 ± 0.003	0.069 ± 0.003
WS	Kuramoto	0.060 ± 0.002	0.051 ± 0.0030	0.053 ± 0.002	0.053 ± 0.004	0.054 ± 0.002
	Branch	0.0480 ± 0.004	0.033 ± 0.002	0.040 ± 0.005	0.034 ± 0.003	0.036 ± 0.004
	CML	0.067 ± 0.005	0.061 ± 0.004	0.064 ± 0.002	0.065 ± 0.005	0.067 ± 0.004
Dolphins	Kuramoto	0.049 ± 0.002	0.029 ± 0.007	0.029 ± 0.007	0.030 ± 0.004	0.031 ± 0.002
	Branch	0.049 ± 0.002	0.045 ± 0.007	0.037 ± 0.007	0.044 ± 0.005	0.043 ± 0.002
	CML	0.075 ± 0.005	0.044 ± 0.004	0.045 ± 0.003	0.056 ± 0.004	0.059 ± 0.002
Word	Kuramoto	0.071 ± 0.004	0.064 ± 0.003	0.066 ± 0.004	0.070 ± 0.003	0.071 ± 0.003
	Branch	0.096 ± 0.004	0.064 ± 0.008	0.068 ± 0.005	0.061 ± 0.005	0.061 ± 0.004
	CML	0.081 ± 0.007	0.076 ± 0.003	0.068 ± 0.002	0.074 ± 0.002	0.079 ± 0.006
Ca-netsci	Kuramoto	0.074 ± 0.003	0.061 ± 0.004	0.063 ± 0.005	0.065 ± 0.006	0.067 ± 0.006
	Branch	0.059 ± 0.003	0.033 ± 0.004	0.037 ± 0.002	0.036 ± 0.003	0.036 ± 0.004
	CML	0.060 ± 0.006	0.048 ± 0.003	0.049 ± 0.002	0.052 ± 0.004	0.054 ± 0.003
Email	Kuramoto	0.069 ± 0.004	0.058 ± 0.009	0.041 ± 0.003	0.046 ± 0.002	0.046 ± 0.003
	Branch	0.042 ± 0.004	0.036 ± 0.004	0.039 ± 0.005	0.042 ± 0.003	0.043 ± 0.002
	CML	0.068 ± 0.002	0.032 ± 0.003	0.038 ± 0.006	0.032 ± 0.004	0.033 ± 0.004

TABLE X. Network inference performance of CoND and other baseline methods in terms of the Net_acc metric for five discrete dynamical models, with the GCN model containing 80% dynamics and 80% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Threshold	0.635 ± 0.036	0.859 ± 0.026	0.631 ± 0.034	0.022 ± 0.002	0.655 ± 0.041				
	Voter	0.548 ± 0.027	0.653 ± 0.051	0.629 ± 0.024	0.021 ± 0.004	0.599 ± 0.016				
	Ising	0.576 ± 0.011	0.626 ± 0.017	0.633 ± 0.040	0.018 ± 0.004	0.524 ± 0.020	0.012 ± 0.009	0.037 ± 0.010	0.050 ± 0.018	0.024 ± 0.014
	SIS	0.589 ± 0.067	0.652 ± 0.055	0.643 ± 0.056	0.023 ± 0.004	0.617 ± 0.009				
ER	SIR	0.356 ± 0.021	0.500 ± 0.054	0.640 ± 0.032	0.033 ± 0.001	0.492 ± 0.018				
	Threshold	0.635 ± 0.014	0.859 ± 0.015	0.631 ± 0.022	0.005 ± 0.004	0.646 ± 0.032				
	Voter	0.548 ± 0.028	0.653 ± 0.025	0.629 ± 0.01	0.0019 ± 0.0015	0.592 ± 0.021				
	Ising	0.576 ± 0.011	0.596 ± 0.016	0.633 ± 0.024	0.002 ± 0.001	0.629 ± 0.013	0.002 ± 0.002	0.003 ± 0.004	0.003 ± 0.003	0.002 ± 0.002
WS	SIS	0.589 ± 0.026	0.672 ± 0.013	0.643 ± 0.022	0.005 ± 0.001	0.615 ± 0.002				
	SIR	0.356 ± 0.047	0.710 ± 0.056	0.640 ± 0.051	0.006 ± 0.003	0.502 ± 0.013				
	Threshold	0.638 ± 0.021	0.840 ± 0.027	0.638 ± 0.024	0.001 ± 0.001	0.673 ± 0.009				
	Voter	0.629 ± 0.021	0.726 ± 0.023	0.645 ± 0.022	0.001 ± 0.001	0.650 ± 0.022				
Dolphins	Ising	0.631 ± 0.021	0.635 ± 0.019	0.634 ± 0.021	0.003 ± 0.001	0.700 ± 0.025	0.026 ± 0.003	0.016 ± 0.002	0.021 ± 0.003	0.033 ± 0.003
	SIS	0.609 ± 0.027	0.670 ± 0.041	0.636 ± 0.021	0.002 ± 0.001	0.600 ± 0.034				
	SIR	0.474 ± 0.071	0.613 ± 0.052	0.654 ± 0.032	0.004 ± 0.001	0.463 ± 0.026				
	Threshold	0.636 ± 0.076	0.729 ± 0.072	0.619 ± 0.050	0.022 ± 0.004	0.550 ± 0.015				
Word	Voter	0.200 ± 0.058	0.684 ± 0.029	0.568 ± 0.044	0.046 ± 0.004	0.503 ± 0.034				
	Ising	0.197 ± 0.069	0.548 ± 0.059	0.507 ± 0.044	0.013 ± 0.005	0.617 ± 0.038	0.111 ± 0.015	0.089 ± 0.008	0.093 ± 0.007	0.030 ± 0.016
	SIS	0.442 ± 0.022	0.484 ± 0.035	0.613 ± 0.047	0.017 ± 0.003	0.505 ± 0.026				
	SIR	0.442 ± 0.067	0.532 ± 0.027	0.645 ± 0.041	0.023 ± 0.001	0.383 ± 0.050				
Ca-netsci	Threshold	0.604 ± 0.040	0.678 ± 0.025	0.604 ± 0.049	0.018 ± 0.001	0.660 ± 0.035				
	Voter	0.120 ± 0.025	0.468 ± 0.021	0.429 ± 0.027	0.012 ± 0.001	0.445 ± 0.026				
	Ising	0.144 ± 0.066	0.406 ± 0.023	0.369 ± 0.027	0.024 ± 0.003	0.541 ± 0.030	0.066 ± 0.039	0.061 ± 0.022	0.070 ± 0.021	0.103 ± 0.030
	SIS	0.329 ± 0.024	0.598 ± 0.014	0.424 ± 0.050	0.012 ± 0.003	0.421 ± 0.027				
Email	SIR	0.252 ± 0.060	0.472 ± 0.055	0.545 ± 0.025	0.035 ± 0.004	0.421 ± 0.023				
	Threshold	0.604 ± 0.014	0.678 ± 0.022	0.604 ± 0.025	0.001 ± 0.002	0.542 ± 0.030				
	Voter	0.120 ± 0.019	0.538 ± 0.020	0.429 ± 0.029	0.003 ± 0.001	0.487 ± 0.024				
	Ising	0.144 ± 0.014	0.406 ± 0.026	0.369 ± 0.018	0.002 ± 0.002	0.597 ± 0.002	0.310 ± 0.027	0.464 ± 0.010	0.442 ± 0.018	0.128 ± 0.005
Email	SIS	0.329 ± 0.022	0.608 ± 0.057	0.424 ± 0.038	0.002 ± 0.001	0.542 ± 0.020				
	SIR	0.252 ± 0.009	0.472 ± 0.049	0.545 ± 0.056	0.001 ± 0.001	0.356 ± 0.041				
	Threshold	0.604 ± 0.064	0.678 ± 0.018	0.604 ± 0.023	0.003 ± 0.004	0.502 ± 0.021				
	Voter	0.120 ± 0.015	0.468 ± 0.012	0.429 ± 0.017	0.006 ± 0.001	0.354 ± 0.037				
Email	Ising	0.144 ± 0.016	0.406 ± 0.031	0.369 ± 0.019	0.003 ± 0.002	0.366 ± 0.022	0.138 ± 0.012	0.138 ± 0.008	0.153 ± 0.011	0.058 ± 0.011
	SIS	0.329 ± 0.051	0.488 ± 0.047	0.424 ± 0.064	0.005 ± 0.002	0.433 ± 0.019				
	SIR	0.252 ± 0.068	0.472 ± 0.059	0.545 ± 0.040	0.007 ± 0.001	0.337 ± 0.041				

TABLE XI. Network inference performance of CoND and other baseline methods in terms of the Net_acc metric for three continuous dynamical models, with the GCN model containing 80% dynamics and 80% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Kuramoto	0.214 ± 0.019	0.414 ± 0.028	0.299 ± 0.053	0.021 ± 0.003	0.283 ± 0.039				
	Branch	0.210 ± 0.021	0.409 ± 0.038	0.352 ± 0.050	0.020 ± 0.013	0.232 ± 0.019	0.012 ± 0.009	0.037 ± 0.010	0.050 ± 0.018	0.024 ± 0.014
	CML	0.151 ± 0.027	0.306 ± 0.014	0.344 ± 0.024	0.001 ± 0.014	0.173 ± 0.014				
ER	Kuramoto	0.284 ± 0.022	0.558 ± 0.031	0.462 ± 0.020	0.002 ± 0.001	0.261 ± 0.022				
	Branch	0.230 ± 0.041	0.250 ± 0.023	0.280 ± 0.033	0.002 ± 0.001	0.129 ± 0.030	0.002 ± 0.002	0.003 ± 0.004	0.003 ± 0.003	0.002 ± 0.001
	CML	0.177 ± 0.059	0.282 ± 0.023	0.256 ± 0.028	0.003 ± 0.002	0.109 ± 0.015				
WS	Kuramoto	0.305 ± 0.028	0.6399 ± 0.041	0.608 ± 0.057	0.003 ± 0.003	0.233 ± 0.013				
	Branch	0.206 ± 0.075	0.431 ± 0.025	0.440 ± 0.052	0.002 ± 0.001	0.289 ± 0.021	0.026 ± 0.003	0.016 ± 0.002	0.021 ± 0.003	0.033 ± 0.003
	CML	0.111 ± 0.014	0.279 ± 0.019	0.225 ± 0.031	0.001 ± 0.001	0.268 ± 0.013				
Dolphins	Kuramoto	0.219 ± 0.031	0.294 ± 0.032	0.292 ± 0.035	0.012 ± 0.032	0.215 ± 0.012				
	Branch	0.190 ± 0.093	0.304 ± 0.039	0.355 ± 0.042	0.024 ± 0.012	0.290 ± 0.022	0.111 ± 0.015	0.089 ± 0.008	0.093 ± 0.007	0.030 ± 0.016
	CML	0.197 ± 0.065	0.342 ± 0.030	0.261 ± 0.030	0.031 ± 0.020	0.235 ± 0.012				
Word	Kuramoto	0.295 ± 0.055	0.345 ± 0.023	0.275 ± 0.031	0.011 ± 0.001	0.221 ± 0.017				
	Branch	0.133 ± 0.039	0.331 ± 0.036	0.233 ± 0.022	0.020 ± 0.001	0.243 ± 0.032	0.066 ± 0.039	0.061 ± 0.022	0.070 ± 0.021	0.103 ± 0.030
	CML	0.138 ± 0.015	0.294 ± 0.031	0.233 ± 0.027	0.014 ± 0.002	0.286 ± 0.025				
Ca-netsci	Kuramoto	0.254 ± 0.066	0.297 ± 0.038	0.378 ± 0.034	0.004 ± 0.001	0.195 ± 0.021				
	Branch	0.138 ± 0.060	0.260 ± 0.020	0.215 ± 0.041	0.005 ± 0.002	0.189 ± 0.017	0.310 ± 0.027	0.464 ± 0.010	0.442 ± 0.018	0.128 ± 0.005
	CML	0.123 ± 0.006	0.233 ± 0.026	0.292 ± 0.023	0.003 ± 0.002	0.194 ± 0.044				
Email	Kuramoto	0.179 ± 0.039	0.221 ± 0.012	0.277 ± 0.022	0.001 ± 0.001	0.160 ± 0.022				
	Branch	0.135 ± 0.027	0.240 ± 0.028	0.164 ± 0.026	0.002 ± 0.001	0.182 ± 0.030	0.138 ± 0.012	0.138 ± 0.008	0.153 ± 0.011	0.058 ± 0.011
	CML	0.202 ± 0.024	0.348 ± 0.021	0.266 ± 0.022	0.001 ± 0.001	0.201 ± 0.020				

TABLE XII. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_acc metric for five discrete dynamical models, with the SAGE model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Threshold	0.812 ± 0.004	0.982 ± 0.004	0.978 ± 0.003	0.954 ± 0.005	0.947 ± 0.006
	Voter	0.624 ± 0.006	0.758 ± 0.001	0.757 ± 0.002	0.753 ± 0.003	0.752 ± 0.003
	Ising	0.585 ± 0.002	0.705 ± 0.002	0.707 ± 0.001	0.702 ± 0.002	0.702 ± 0.002
	SIS	0.783 ± 0.009	0.835 ± 0.004	0.834 ± 0.002	0.833 ± 0.002	0.830 ± 0.001
	SIR	0.774 ± 0.003	0.835 ± 0.001	0.836 ± 0.002	0.834 ± 0.004	0.831 ± 0.004
ER	Threshold	0.833 ± 0.003	0.985 ± 0.002	0.980 ± 0.003	0.950 ± 0.002	0.950 ± 0.003
	Voter	0.594 ± 0.005	0.711 ± 0.001	0.710 ± 0.001	0.702 ± 0.003	0.702 ± 0.002
	Ising	0.577 ± 0.003	0.680 ± 0.001	0.684 ± 0.002	0.680 ± 0.002	0.680 ± 0.002
	SIS	0.804 ± 0.001	0.835 ± 0.004	0.833 ± 0.002	0.833 ± 0.001	0.832 ± 0.001
	SIR	0.809 ± 0.001	0.829 ± 0.002	0.827 ± 0.003	0.813 ± 0.001	0.810 ± 0.008
WS	Threshold	0.823 ± 0.002	0.987 ± 0.002	0.983 ± 0.001	0.952 ± 0.003	0.952 ± 0.002
	Voter	0.627 ± 0.006	0.746 ± 0.001	0.750 ± 0.001	0.743 ± 0.002	0.743 ± 0.002
	Ising	0.561 ± 0.006	0.697 ± 0.002	0.694 ± 0.002	0.684 ± 0.002	0.686 ± 0.002
	SIS	0.879 ± 0.004	0.895 ± 0.001	0.893 ± 0.001	0.895 ± 0.003	0.894 ± 0.003
	SIR	0.860 ± 0.002	0.898 ± 0.001	0.896 ± 0.001	0.889 ± 0.002	0.885 ± 0.002
Dolphins	Threshold	0.900 ± 0.009	0.995 ± 0.001	0.992 ± 0.002	0.958 ± 0.003	0.957 ± 0.002
	Voter	0.691 ± 0.005	0.776 ± 0.006	0.774 ± 0.003	0.762 ± 0.007	0.762 ± 0.007
	Ising	0.640 ± 0.006	0.725 ± 0.002	0.725 ± 0.001	0.714 ± 0.008	0.712 ± 0.008
	SIS	0.689 ± 0.009	0.738 ± 0.006	0.739 ± 0.003	0.735 ± 0.004	0.729 ± 0.003
	SIR	0.676 ± 0.008	0.713 ± 0.002	0.712 ± 0.005	0.704 ± 0.003	0.700 ± 0.008
Word	Threshold	0.876 ± 0.007	0.992 ± 0.002	0.986 ± 0.002	0.957 ± 0.008	0.954 ± 0.001
	Voter	0.628 ± 0.005	0.717 ± 0.002	0.713 ± 0.003	0.700 ± 0.004	0.703 ± 0.006
	Ising	0.595 ± 0.002	0.673 ± 0.004	0.676 ± 0.004	0.671 ± 0.002	0.671 ± 0.002
	SIS	0.660 ± 0.007	0.739 ± 0.004	0.739 ± 0.007	0.723 ± 0.006	0.726 ± 0.001
	SIR	0.614 ± 0.007	0.713 ± 0.006	0.718 ± 0.006	0.713 ± 0.005	0.705 ± 0.003
Ca-netsci	Threshold	0.751 ± 0.006	0.989 ± 0.001	0.984 ± 0.002	0.965 ± 0.002	0.961 ± 0.003
	Voter	0.608 ± 0.002	0.786 ± 0.001	0.785 ± 0.001	0.781 ± 0.001	0.780 ± 0.002
	Ising	0.553 ± 0.006	0.728 ± 0.001	0.718 ± 0.001	0.716 ± 0.001	0.714 ± 0.002
	SIS	0.845 ± 0.009	0.822 ± 0.002	0.829 ± 0.001	0.823 ± 0.004	0.820 ± 0.004
	SIR	0.876 ± 0.003	0.821 ± 0.001	0.829 ± 0.002	0.811 ± 0.001	0.806 ± 0.001
Email	Threshold	0.731 ± 0.006	0.973 ± 0.001	0.969 ± 0.002	0.923 ± 0.002	0.919 ± 0.003
	Voter	0.554 ± 0.008	0.711 ± 0.001	0.707 ± 0.001	0.705 ± 0.001	0.704 ± 0.002
	Ising	0.546 ± 0.009	0.670 ± 0.002	0.670 ± 0.001	0.669 ± 0.002	0.669 ± 0.002
	SIS	0.861 ± 0.008	0.878 ± 0.002	0.880 ± 0.001	0.889 ± 0.003	0.883 ± 0.0010
	SIR	0.862 ± 0.009	0.871 ± 0.002	0.883 ± 0.002	0.876 ± 0.002	0.862 ± 0.005

TABLE XIII. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_mse metric for three continuous dynamical models, with the SAGE model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Kuramoto	0.084 ± 0.005	0.019 ± 0.003	0.029 ± 0.009	0.031 ± 0.004	0.032 ± 0.007
	Branch	0.075 ± 0.003	0.060 ± 0.005	0.064 ± 0.002	0.061 ± 0.006	0.061 ± 0.003
	CML	0.084 ± 0.009	0.058 ± 0.002	0.059 ± 0.004	0.054 ± 0.004	0.054 ± 0.005
ER	Kuramoto	0.071 ± 0.008	0.042 ± 0.001	0.038 ± 0.004	0.039 ± 0.003	0.039 ± 0.002
	Branch	0.024 ± 0.004	0.020 ± 0.002	0.024 ± 0.005	0.022 ± 0.005	0.021 ± 0.007
	CML	0.076 ± 0.003	0.068 ± 0.002	0.064 ± 0.003	0.056 ± 0.002	0.056 ± 0.006
WS	Kuramoto	0.153 ± 0.007	0.117 ± 0.004	0.119 ± 0.005	0.119 ± 0.003	0.118 ± 0.004
	Branch	0.030 ± 0.005	0.018 ± 0.003	0.018 ± 0.005	0.015 ± 0.005	0.015 ± 0.006
	CML	0.087 ± 0.006	0.050 ± 0.003	0.050 ± 0.004	0.056 ± 0.010	0.056 ± 0.001
Dolphins	Kuramoto	0.157 ± 0.003	0.124 ± 0.004	0.133 ± 0.001	0.136 ± 0.004	0.136 ± 0.002
	Branch	0.039 ± 0.006	0.038 ± 0.005	0.033 ± 0.001	0.036 ± 0.005	0.036 ± 0.005
	CML	0.062 ± 0.002	0.053 ± 0.004	0.056 ± 0.008	0.056 ± 0.005	0.056 ± 0.003
Word	Kuramoto	0.082 ± 0.003	0.067 ± 0.002	0.061 ± 0.005	0.058 ± 0.002	0.059 ± 0.002
	Branch	0.091 ± 0.007	0.083 ± 0.003	0.085 ± 0.006	0.089 ± 0.005	0.088 ± 0.007
	CML	0.037 ± 0.004	0.022 ± 0.002	0.030 ± 0.004	0.027 ± 0.004	0.027 ± 0.006
Ca-netsci	Kuramoto	0.105 ± 0.007	0.092 ± 0.002	0.093 ± 0.002	0.093 ± 0.003	0.093 ± 0.003
	Branch	0.075 ± 0.004	0.047 ± 0.006	0.056 ± 0.001	0.035 ± 0.003	0.035 ± 0.003
	CML	0.070 ± 0.010	0.058 ± 0.006	0.057 ± 0.001	0.059 ± 0.001	0.059 ± 0.003
Email	Kuramoto	0.083 ± 0.005	0.071 ± 0.004	0.076 ± 0.002	0.077 ± 0.004	0.079 ± 0.008
	Branch	0.029 ± 0.003	0.024 ± 0.005	0.023 ± 0.002	0.025 ± 0.004	0.024 ± 0.002
	CML	0.108 ± 0.009	0.074 ± 0.005	0.070 ± 0.002	0.061 ± 0.002	0.061 ± 0.005

TABLE XIV. Network inference performance of CoND and other baseline methods in terms of the Net_acc metric for five discrete dynamical models, with the SAGE model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Threshold	0.490 ± 0.056	0.923 ± 0.053	0.803 ± 0.038	0.044 ± 0.005	0.693 ± 0.057				
	Voter	0.326 ± 0.039	0.805 ± 0.023	0.795 ± 0.065	0.038 ± 0.007	0.641 ± 0.031				
	Ising	0.477 ± 0.046	0.815 ± 0.077	0.782 ± 0.025	0.008 ± 0.009	0.667 ± 0.022	0.013 ± 0.010	0.039 ± 0.012	0.051 ± 0.019	0.026 ± 0.015
	SIS	0.497 ± 0.041	0.787 ± 0.027	0.764 ± 0.076	0.038 ± 0.010	0.635 ± 0.024				
ER	SIR	0.300 ± 0.084	0.692 ± 0.059	0.567 ± 0.040	0.044 ± 0.014	0.567 ± 0.012				
	Threshold	0.819 ± 0.036	0.987 ± 0.011	0.819 ± 0.036	0.002 ± 0.007	0.759 ± 0.048				
	Voter	0.744 ± 0.060	0.819 ± 0.044	0.808 ± 0.036	0.001 ± 0.004	0.697 ± 0.037				
	Ising	0.677 ± 0.044	0.777 ± 0.035	0.821 ± 0.039	0.001 ± 0.005	0.630 ± 0.028	0.002 ± 0.002	0.003 ± 0.003	0.002 ± 0.003	0.001 ± 0.001
WS	SIS	0.750 ± 0.048	0.835 ± 0.051	0.808 ± 0.025	0.004 ± 0.004	0.702 ± 0.016				
	SIR	0.696 ± 0.083	0.765 ± 0.051	0.762 ± 0.022	0.004 ± 0.006	0.599 ± 0.018				
	Threshold	0.763 ± 0.067	0.985 ± 0.014	0.825 ± 0.025	0.006 ± 0.004	0.778 ± 0.024				
	Voter	0.818 ± 0.063	0.893 ± 0.057	0.838 ± 0.074	0.001 ± 0.005	0.751 ± 0.013				
Dolphins	Ising	0.813 ± 0.068	0.830 ± 0.060	0.815 ± 0.065	0.004 ± 0.003	0.676 ± 0.020	0.025 ± 0.001	0.015 ± 0.001	0.020 ± 0.001	0.032 ± 0.001
	SIS	0.750 ± 0.078	0.803 ± 0.048	0.783 ± 0.047	0.003 ± 0.004	0.689 ± 0.059				
	SIR	0.608 ± 0.053	0.768 ± 0.027	0.730 ± 0.064	0.004 ± 0.007	0.529 ± 0.041				
	Threshold	0.887 ± 0.056	0.967 ± 0.037	0.873 ± 0.087	0.033 ± 0.017	0.613 ± 0.030				
Word	Voter	0.367 ± 0.046	0.747 ± 0.049	0.747 ± 0.028	0.023 ± 0.016	0.473 ± 0.070				
	Ising	0.513 ± 0.056	0.727 ± 0.030	0.680 ± 0.089	0.027 ± 0.019	0.686 ± 0.073	0.117 ± 0.021	0.095 ± 0.014	0.098 ± 0.012	0.035 ± 0.021
	SIS	0.753 ± 0.054	0.773 ± 0.045	0.647 ± 0.037	0.033 ± 0.025	0.569 ± 0.022				
	SIR	0.673 ± 0.054	0.647 ± 0.052	0.647 ± 0.029	0.040 ± 0.023	0.449 ± 0.065				
Ca-netsci	Threshold	0.817 ± 0.066	0.943 ± 0.024	0.836 ± 0.055	0.007 ± 0.004	0.643 ± 0.050				
	Voter	0.310 ± 0.052	0.643 ± 0.048	0.595 ± 0.053	0.006 ± 0.004	0.519 ± 0.021				
	Ising	0.307 ± 0.055	0.538 ± 0.085	0.555 ± 0.055	0.007 ± 0.006	0.545 ± 0.045	0.068 ± 0.041	0.063 ± 0.023	0.071 ± 0.022	0.104 ± 0.032
	SIS	0.367 ± 0.052	0.676 ± 0.038	0.745 ± 0.073	0.008 ± 0.006	0.440 ± 0.032				
Email	SIR	0.248 ± 0.078	0.571 ± 0.061	0.639 ± 0.052	0.012 ± 0.007	0.419 ± 0.038				
	Threshold	0.463 ± 0.080	0.937 ± 0.010	0.819 ± 0.037	0.002 ± 0.004	0.623 ± 0.045				
	Voter	0.501 ± 0.016	0.782 ± 0.034	0.785 ± 0.010	0.002 ± 0.005	0.548 ± 0.039				
	Ising	0.542 ± 0.052	0.784 ± 0.033	0.757 ± 0.062	0.003 ± 0.002	0.678 ± 0.015	0.354 ± 0.032	0.568 ± 0.014	0.547 ± 0.023	0.132 ± 0.009
Email	SIS	0.468 ± 0.014	0.688 ± 0.040	0.592 ± 0.041	0.002 ± 0.005	0.601 ± 0.036				
	SIR	0.284 ± 0.074	0.678 ± 0.031	0.607 ± 0.094	0.002 ± 0.004	0.469 ± 0.014				
	Threshold	0.573 ± 0.028	0.985 ± 0.012	0.982 ± 0.041	0.002 ± 0.002	0.617 ± 0.036				
	Voter	0.306 ± 0.035	0.589 ± 0.039	0.563 ± 0.041	0.003 ± 0.001	0.474 ± 0.052				
Email	Ising	0.255 ± 0.044	0.486 ± 0.042	0.459 ± 0.063	0.004 ± 0.001	0.455 ± 0.025	0.136 ± 0.009	0.135 ± 0.005	0.151 ± 0.008	0.056 ± 0.009
	SIS	0.108 ± 0.015	0.512 ± 0.041	0.644 ± 0.020	0.002 ± 0.002	0.509 ± 0.025				
	SIR	0.046 ± 0.060	0.545 ± 0.036	0.579 ± 0.054	0.002 ± 0.002	0.429 ± 0.056				

TABLE XV. Network inference performance of CoND and other baseline methods in terms of the Net_acc metric for three continuous dynamical models, with the SAGE model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Kuramoto	0.292 ± 0.018	0.651 ± 0.034	0.523 ± 0.041	0.022 ± 0.004	0.336 ± 0.042				
	Branch	0.418 ± 0.066	0.444 ± 0.037	0.467 ± 0.049	0.021 ± 0.015	0.285 ± 0.022	0.013 ± 0.010	0.039 ± 0.012	0.051 ± 0.019	0.026 ± 0.015
	CML	0.251 ± 0.029	0.410 ± 0.100	0.359 ± 0.070	0.002 ± 0.016	0.226 ± 0.016				
ER	Kuramoto	0.421 ± 0.024	0.723 ± 0.057	0.567 ± 0.058	0.003 ± 0.002	0.315 ± 0.024				
	Branch	0.235 ± 0.040	0.539 ± 0.065	0.308 ± 0.059	0.003 ± 0.002	0.182 ± 0.032	0.002 ± 0.002	0.003 ± 0.003	0.002 ± 0.003	0.001 ± 0.001
	CML	0.262 ± 0.028	0.515 ± 0.040	0.442 ± 0.091	0.004 ± 0.001	0.162 ± 0.017				
WS	Kuramoto	0.598 ± 0.011	0.775 ± 0.022	0.233 ± 0.030	0.004 ± 0.005	0.287 ± 0.015				
	Branch	0.450 ± 0.048	0.550 ± 0.052	0.425 ± 0.045	0.003 ± 0.002	0.343 ± 0.023	0.025 ± 0.001	0.015 ± 0.001	0.020 ± 0.001	0.032 ± 0.001
	CML	0.3025 ± 0.031	0.543 ± 0.046	0.473 ± 0.035	0.002 ± 0.001	0.321 ± 0.015				
Dolphins	Kuramoto	0.320 ± 0.021	0.410 ± 0.039	0.405 ± 0.035	0.013 ± 0.033	0.268 ± 0.014				
	Branch	0.267 ± 0.052	0.427 ± 0.073	0.356 ± 0.056	0.026 ± 0.013	0.343 ± 0.024	0.117 ± 0.021	0.095 ± 0.014	0.098 ± 0.012	0.035 ± 0.021
	CML	0.107 ± 0.031	0.227 ± 0.063	0.345 ± 0.015	0.032 ± 0.021	0.288 ± 0.014				
Word	Kuramoto	0.333 ± 0.017	0.476 ± 0.048	0.226 ± 0.011	0.012 ± 0.002	0.275 ± 0.019				
	Branch	0.355 ± 0.056	0.405 ± 0.042	0.333 ± 0.073	0.021 ± 0.002	0.297 ± 0.034	0.068 ± 0.041	0.063 ± 0.023	0.071 ± 0.022	0.104 ± 0.032
	CML	0.113 ± 0.045	0.357 ± 0.044	0.286 ± 0.044	0.015 ± 0.001	0.339 ± 0.027				
Ca-netsci	Kuramoto	0.211 ± 0.014	0.266 ± 0.052	0.322 ± 0.043	0.005 ± 0.002	0.248 ± 0.023				
	Branch	0.330 ± 0.038	0.451 ± 0.052	0.425 ± 0.053	0.006 ± 0.003	0.242 ± 0.020	0.354 ± 0.032	0.568 ± 0.014	0.547 ± 0.023	0.132 ± 0.009
	CML	0.256 ± 0.069	0.359 ± 0.040	0.539 ± 0.039	0.004 ± 0.003	0.248 ± 0.047				
Email	Kuramoto	0.233 ± 0.015	0.347 ± 0.073	0.210 ± 0.051	0.002 ± 0.002	0.214 ± 0.024				
	Branch	0.241 ± 0.033	0.328 ± 0.048	0.264 ± 0.066	0.003 ± 0.003	0.135 ± 0.033	0.136 ± 0.009	0.135 ± 0.005	0.151 ± 0.008	0.056 ± 0.009
	CML	0.212 ± 0.036	0.192 ± 0.026	0.346 ± 0.046	0.001 ± 0.001	0.050 ± 0.023				

TABLE XVI. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_acc metric for five discrete dynamical models, with the GIN model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Threshold	0.835 ± 0.004	0.934 ± 0.002	0.913 ± 0.002	0.909 ± 0.002	0.910 ± 0.002
	Voter	0.622 ± 0.003	0.722 ± 0.004	0.731 ± 0.003	0.729 ± 0.003	0.727 ± 0.003
	Ising	0.585 ± 0.004	0.636 ± 0.002	0.686 ± 0.002	0.673 ± 0.002	0.674 ± 0.001
	SIS	0.801 ± 0.003	0.842 ± 0.002	0.843 ± 0.002	0.842 ± 0.002	0.841 ± 0.002
	SIR	0.827 ± 0.005	0.849 ± 0.004	0.841 ± 0.004	0.839 ± 0.005	0.840 ± 0.005
ER	Threshold	0.829 ± 0.003	0.964 ± 0.002	0.963 ± 0.002	0.901 ± 0.002	0.881 ± 0.003
	Voter	0.625 ± 0.003	0.731 ± 0.003	0.735 ± 0.004	0.722 ± 0.002	0.724 ± 0.004
	Ising	0.562 ± 0.004	0.688 ± 0.004	0.682 ± 0.003	0.669 ± 0.005	0.673 ± 0.006
	SIS	0.880 ± 0.003	0.896 ± 0.001	0.895 ± 0.001	0.895 ± 0.001	0.893 ± 0.001
	SIR	0.829 ± 0.004	0.899 ± 0.002	0.897 ± 0.002	0.897 ± 0.005	0.895 ± 0.007
WS	Threshold	0.832 ± 0.005	0.954 ± 0.004	0.939 ± 0.003	0.877 ± 0.002	0.881 ± 0.002
	Voter	0.591 ± 0.004	0.691 ± 0.003	0.684 ± 0.004	0.675 ± 0.002	0.675 ± 0.002
	Ising	0.581 ± 0.004	0.677 ± 0.004	0.670 ± 0.003	0.649 ± 0.005	0.653 ± 0.004
	SIS	0.803 ± 0.005	0.835 ± 0.002	0.835 ± 0.001	0.833 ± 0.001	0.832 ± 0.001
	SIR	0.810 ± 0.004	0.834 ± 0.002	0.831 ± 0.002	0.832 ± 0.002	0.828 ± 0.003
Dolphins	Threshold	0.889 ± 0.004	0.934 ± 0.002	0.929 ± 0.002	0.870 ± 0.002	0.880 ± 0.003
	Voter	0.698 ± 0.004	0.719 ± 0.003	0.720 ± 0.003	0.704 ± 0.002	0.713 ± 0.003
	Ising	0.642 ± 0.006	0.681 ± 0.002	0.676 ± 0.002	0.642 ± 0.002	0.635 ± 0.002
	SIS	0.684 ± 0.003	0.737 ± 0.001	0.737 ± 0.001	0.730 ± 0.006	0.724 ± 0.004
	SIR	0.668 ± 0.002	0.729 ± 0.003	0.728 ± 0.003	0.722 ± 0.002	0.721 ± 0.003
Word	Threshold	0.872 ± 0.006	0.947 ± 0.003	0.945 ± 0.003	0.898 ± 0.003	0.903 ± 0.003
	Voter	0.623 ± 0.005	0.679 ± 0.003	0.675 ± 0.003	0.677 ± 0.001	0.675 ± 0.009
	Ising	0.593 ± 0.006	0.636 ± 0.006	0.636 ± 0.007	0.633 ± 0.002	0.637 ± 0.003
	SIS	0.665 ± 0.005	0.740 ± 0.004	0.742 ± 0.004	0.736 ± 0.005	0.737 ± 0.003
	SIR	0.616 ± 0.003	0.733 ± 0.003	0.735 ± 0.003	0.731 ± 0.005	0.725 ± 0.005
Ca-netsci	Threshold	0.756 ± 0.007	0.946 ± 0.002	0.941 ± 0.003	0.927 ± 0.004	0.902 ± 0.003
	Voter	0.613 ± 0.007	0.764 ± 0.004	0.768 ± 0.004	0.753 ± 0.005	0.752 ± 0.002
	Ising	0.552 ± 0.004	0.700 ± 0.004	0.697 ± 0.004	0.699 ± 0.003	0.693 ± 0.002
	SIS	0.813 ± 0.003	0.823 ± 0.001	0.822 ± 0.002	0.822 ± 0.001	0.822 ± 0.003
	SIR	0.760 ± 0.005	0.816 ± 0.003	0.820 ± 0.005	0.814 ± 0.002	0.816 ± 0.004
Email	Threshold	0.783 ± 0.005	0.947 ± 0.003	0.937 ± 0.002	0.887 ± 0.004	0.887 ± 0.003
	Voter	0.544 ± 0.002	0.673 ± 0.002	0.667 ± 0.002	0.658 ± 0.003	0.656 ± 0.003
	Ising	0.530 ± 0.004	0.667 ± 0.004	0.660 ± 0.003	0.659 ± 0.003	0.667 ± 0.003
	SIS	0.767 ± 0.003	0.813 ± 0.002	0.802 ± 0.002	0.804 ± 0.003	0.800 ± 0.003
	SIR	0.766 ± 0.004	0.812 ± 0.002	0.804 ± 0.002	0.804 ± 0.002	0.798 ± 0.003

TABLE XVII. Dynamical prediction performance of CoND and other baseline methods in terms of the Dyn_mse metric for three continuous dynamical models, with the GIN model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	CoD ₁	CoD ₂
BA	Kuramoto	0.081 ± 0.004	0.059 ± 0.003	0.080 ± 0.003	0.068 ± 0.002	0.068 ± 0.003
	Branch	0.074 ± 0.003	0.063 ± 0.002	0.065 ± 0.001	0.069 ± 0.005	0.070 ± 0.003
	CML	0.092 ± 0.002	0.057 ± 0.003	0.052 ± 0.003	0.053 ± 0.003	0.054 ± 0.002
ER	Kuramoto	0.095 ± 0.003	0.062 ± 0.003	0.076 ± 0.002	0.065 ± 0.001	0.066 ± 0.001
	Branch	0.036 ± 0.009	0.036 ± 0.002	0.030 ± 0.002	0.022 ± 0.004	0.021 ± 0.003
	CML	0.079 ± 0.003	0.084 ± 0.003	0.071 ± 0.004	0.061 ± 0.002	0.062 ± 0.002
WS	Kuramoto	0.157 ± 0.005	0.065 ± 0.002	0.087 ± 0.004	0.069 ± 0.005	0.071 ± 0.001
	Branch	0.031 ± 0.004	0.019 ± 0.001	0.020 ± 0.003	0.015 ± 0.003	0.015 ± 0.002
	CML	0.034 ± 0.002	0.022 ± 0.004	0.022 ± 0.005	0.023 ± 0.002	0.027 ± 0.003
Dolphins	Kuramoto	0.024 ± 0.003	0.018 ± 0.003	0.013 ± 0.005	0.016 ± 0.006	0.017 ± 0.001
	Branch	0.038 ± 0.002	0.036 ± 0.003	0.040 ± 0.002	0.033 ± 0.001	0.033 ± 0.003
	CML	0.026 ± 0.002	0.020 ± 0.003	0.021 ± 0.004	0.023 ± 0.004	0.023 ± 0.003
Word	Kuramoto	0.020 ± 0.003	0.014 ± 0.001	0.014 ± 0.001	0.015 ± 0.004	0.014 ± 0.003
	Branch	0.035 ± 0.004	0.027 ± 0.006	0.028 ± 0.006	0.031 ± 0.003	0.031 ± 0.002
	CML	0.052 ± 0.002	0.044 ± 0.003	0.045 ± 0.002	0.035 ± 0.003	0.037 ± 0.004
Ca-netsci	Kuramoto	0.037 ± 0.002	0.032 ± 0.002	0.034 ± 0.003	0.036 ± 0.003	0.037 ± 0.003
	Branch	0.057 ± 0.004	0.045 ± 0.002	0.047 ± 0.002	0.048 ± 0.003	0.048 ± 0.002
	CML	0.098 ± 0.005	0.064 ± 0.002	0.069 ± 0.005	0.068 ± 0.003	0.069 ± 0.003
Email	Kuramoto	0.046 ± 0.002	0.050 ± 0.004	0.040 ± 0.003	0.048 ± 0.003	0.049 ± 0.002
	Branch	0.074 ± 0.001	0.043 ± 0.002	0.049 ± 0.001	0.041 ± 0.002	0.041 ± 0.001
	CML	0.092 ± 0.003	0.063 ± 0.002	0.064 ± 0.004	0.067 ± 0.003	0.068 ± 0.004

TABLE XVIII. Network inference performance of CoND and other baseline methods in terms of the Net_acc metric for five discrete dynamical models, with the GIN model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Threshold	0.787 ± 0.036	0.933 ± 0.041	0.801 ± 0.036	0.022 ± 0.003	0.752 ± 0.044				
	Voter	0.328 ± 0.041	0.797 ± 0.021	0.736 ± 0.033	0.021 ± 0.005	0.696 ± 0.019				
	Ising	0.480 ± 0.044	0.615 ± 0.035	0.631 ± 0.029	0.018 ± 0.005	0.621 ± 0.012	0.026 ± 0.009	0.014 ± 0.011	0.015 ± 0.012	0.025 ± 0.012
	SIS	0.480 ± 0.055	0.846 ± 0.045	0.862 ± 0.018	0.023 ± 0.005	0.714 ± 0.012				
ER	SIR	0.374 ± 0.062	0.795 ± 0.024	0.746 ± 0.048	0.033 ± 0.001	0.589 ± 0.021				
	Threshold	0.703 ± 0.042	0.915 ± 0.011	0.773 ± 0.025	0.005 ± 0.005	0.743 ± 0.035				
	Voter	0.790 ± 0.024	0.825 ± 0.024	0.800 ± 0.020	0.002 ± 0.001	0.689 ± 0.024				
	Ising	0.820 ± 0.038	0.780 ± 0.030	0.815 ± 0.017	0.002 ± 0.002	0.726 ± 0.015	0.002 ± 0.001	0.002 ± 0.002	0.003 ± 0.002	0.002 ± 0.001
WS	SIS	0.778 ± 0.025	0.863 ± 0.017	0.818 ± 0.064	0.005 ± 0.002	0.712 ± 0.014				
	SIR	0.618 ± 0.029	0.885 ± 0.029	0.833 ± 0.034	0.006 ± 0.004	0.599 ± 0.016				
	Threshold	0.698 ± 0.054	0.879 ± 0.019	0.767 ± 0.035	0.001 ± 0.001	0.770 ± 0.012				
	Voter	0.685 ± 0.052	0.750 ± 0.031	0.702 ± 0.020	0.001 ± 0.001	0.647 ± 0.024				
Dolphins	Ising	0.742 ± 0.024	0.639 ± 0.016	0.596 ± 0.040	0.003 ± 0.001	0.597 ± 0.028	0.015 ± 0.001	0.015 ± 0.001	0.013 ± 0.001	0.014 ± 0.001
	SIS	0.764 ± 0.060	0.837 ± 0.014	0.827 ± 0.033	0.002 ± 0.001	0.717 ± 0.047				
	SIR	0.660 ± 0.038	0.764 ± 0.026	0.754 ± 0.028	0.004 ± 0.001	0.560 ± 0.029				
	Threshold	0.493 ± 0.052	0.873 ± 0.029	0.833 ± 0.052	0.022 ± 0.005	0.647 ± 0.018				
Word	Voter	0.560 ± 0.050	0.680 ± 0.020	0.687 ± 0.034	0.046 ± 0.004	0.600 ± 0.057				
	Ising	0.453 ± 0.035	0.660 ± 0.039	0.560 ± 0.014	0.013 ± 0.006	0.554 ± 0.061	0.121 ± 0.014	0.095 ± 0.012	0.078 ± 0.012	0.092 ± 0.022
	SIS	0.740 ± 0.029	0.653 ± 0.013	0.813 ± 0.042	0.017 ± 0.004	0.602 ± 0.009				
	SIR	0.513 ± 0.021	0.800 ± 0.036	0.600 ± 0.027	0.023 ± 0.002	0.480 ± 0.052				
Ca-netsci	Threshold	0.600 ± 0.053	0.899 ± 0.028	0.858 ± 0.029	0.018 ± 0.002	0.757 ± 0.038				
	Voter	0.310 ± 0.019	0.671 ± 0.022	0.600 ± 0.032	0.012 ± 0.002	0.542 ± 0.009				
	Ising	0.302 ± 0.029	0.560 ± 0.019	0.504 ± 0.011	0.024 ± 0.003	0.438 ± 0.033	0.065 ± 0.020	0.056 ± 0.021	0.062 ± 0.031	0.104 ± 0.021
	SIS	0.402 ± 0.033	0.787 ± 0.026	0.606 ± 0.030	0.012 ± 0.004	0.518 ± 0.020				
Email	SIR	0.329 ± 0.030	0.701 ± 0.021	0.741 ± 0.018	0.035 ± 0.004	0.518 ± 0.026				
	Threshold	0.586 ± 0.075	0.807 ± 0.032	0.719 ± 0.023	0.002 ± 0.001	0.639 ± 0.033				
	Voter	0.476 ± 0.027	0.691 ± 0.022	0.696 ± 0.027	0.003 ± 0.001	0.584 ± 0.027				
	Ising	0.551 ± 0.027	0.647 ± 0.038	0.620 ± 0.017	0.002 ± 0.002	0.562 ± 0.013	0.335 ± 0.022	0.374 ± 0.012	0.440 ± 0.023	0.143 ± 0.011
Email	SIS	0.408 ± 0.023	0.776 ± 0.025	0.729 ± 0.035	0.002 ± 0.001	0.639 ± 0.023				
	SIR	0.260 ± 0.026	0.706 ± 0.024	0.663 ± 0.014	0.001 ± 0.001	0.453 ± 0.032				
	Threshold	0.613 ± 0.038	0.804 ± 0.015	0.927 ± 0.029	0.003 ± 0.001	0.599 ± 0.024				
	Voter	0.341 ± 0.026	0.590 ± 0.050	0.556 ± 0.015	0.006 ± 0.002	0.451 ± 0.040				
Email	Ising	0.275 ± 0.038	0.491 ± 0.035	0.540 ± 0.022	0.003 ± 0.001	0.463 ± 0.013	0.144 ± 0.008	0.144 ± 0.004	0.158 ± 0.005	0.056 ± 0.007
	SIS	0.404 ± 0.015	0.546 ± 0.055	0.609 ± 0.019	0.005 ± 0.003	0.530 ± 0.012				
	SIR	0.356 ± 0.051	0.551 ± 0.024	0.637 ± 0.025	0.007 ± 0.001	0.434 ± 0.044				

TABLE XIX. Network inference performance of CoND and other baseline methods in terms of the Net_acc metric for three continuous dynamical models, with the GIN model containing 90% dynamics and 90% structure. The best accuracy in each row of the results is highlighted in bold.

Network	Dynamics	GGN	CoND	CoND ₀	Correlation	Granger	CN	AA	RA	LP
BA	Kuramoto	0.274 ± 0.025	0.277 ± 0.069	0.339 ± 0.047	0.024 ± 0.008	0.310 ± 0.012				
	Branch	0.418 ± 0.073	0.454 ± 0.054	0.410 ± 0.033	0.022 ± 0.004	0.259 ± 0.011	0.026 ± 0.009	0.014 ± 0.011	0.015 ± 0.012	0.025 ± 0.012
	CML	0.053 ± 0.049	0.282 ± 0.086	0.180 ± 0.070	0.003 ± 0.004	0.200 ± 0.004				
ER	Kuramoto	0.517 ± 0.020	0.635 ± 0.011	0.664 ± 0.049	0.003 ± 0.003	0.299 ± 0.012				
	Branch	0.227 ± 0.050	0.390 ± 0.036	0.375 ± 0.035	0.005 ± 0.005	0.157 ± 0.010	0.002 ± 0.001	0.002 ± 0.002	0.003 ± 0.002	0.002 ± 0.001
	CML	0.075 ± 0.056	0.237 ± 0.083	0.369 ± 0.069	0.005 ± 0.003	0.151 ± 0.010				
WS	Kuramoto	0.590 ± 0.011	0.603 ± 0.056	0.723 ± 0.067	0.004 ± 0.003	0.276 ± 0.013				
	Branch	0.360 ± 0.035	0.475 ± 0.052	0.440 ± 0.054	0.005 ± 0.003	0.332 ± 0.011	0.015 ± 0.001	0.015 ± 0.001	0.013 ± 0.001	0.014 ± 0.001
	CML	0.087 ± 0.061	0.400 ± 0.041	0.350 ± 0.058	0.003 ± 0.003	0.310 ± 0.026				
Dolphins	Kuramoto	0.380 ± 0.089	0.487 ± 0.059	0.447 ± 0.021	0.025 ± 0.003	0.257 ± 0.025				
	Branch	0.300 ± 0.056	0.380 ± 0.076	0.320 ± 0.038	0.027 ± 0.002	0.332 ± 0.035	0.121 ± 0.014	0.095 ± 0.012	0.078 ± 0.012	0.092 ± 0.022
	CML	0.020 ± 0.087	0.247 ± 0.082	0.287 ± 0.067	0.023 ± 0.002	0.277 ± 0.026				
Word	Kuramoto	0.324 ± 0.011	0.293 ± 0.088	0.336 ± 0.039	0.013 ± 0.005	0.264 ± 0.030				
	Branch	0.2308 ± 0.085	0.442 ± 0.072	0.289 ± 0.062	0.023 ± 0.001	0.286 ± 0.025	0.065 ± 0.020	0.056 ± 0.021	0.062 ± 0.031	0.104 ± 0.021
	CML	0.007 ± 0.078	0.133 ± 0.064	0.200 ± 0.061	0.014 ± 0.001	0.151 ± 0.038				
Ca-netsci	Kuramoto	0.158 ± 0.053	0.242 ± 0.032	0.200 ± 0.043	0.003 ± 0.001	0.160 ± 0.028				
	Branch	0.143 ± 0.021	0.231 ± 0.033	0.209 ± 0.059	0.003 ± 0.001	0.154 ± 0.025	0.335 ± 0.022	0.374 ± 0.012	0.440 ± 0.023	0.143 ± 0.011
	CML	0.014 ± 0.075	0.328 ± 0.067	0.319 ± 0.066	0.005 ± 0.002	0.260 ± 0.022				
Email	Kuramoto	0.1991 ± 0.038	0.334 ± 0.050	0.229 ± 0.060	0.005 ± 0.001	0.226 ± 0.029				
	Branch	0.128 ± 0.050	0.268 ± 0.029	0.222 ± 0.046	0.004 ± 0.002	0.147 ± 0.038	0.144 ± 0.008	0.144 ± 0.004	0.158 ± 0.005	0.056 ± 0.007
	CML	0.054 ± 0.063	0.140 ± 0.070	0.191 ± 0.049	0.003 ± 0.001	0.162 ± 0.028				

TABLE XX. Comparison of the performance of CoND and other baseline methods with 90% dynamics and 90% structure based on the SAGE model by using the influenza data and the PM_{2.5} data.

Dynamical prediction									
Dataset	GGN	CoND	CoND ₀	CoD ₁	CoD ₂	\	\	\	\
Influenza	0.008 ± 0.000	0.006 ± 0.000	0.007 ± 0.001	0.007 ± 0.000	0.008 ± 0.000	\	\	\	\
PM _{2.5}	0.006 ± 0.001	0.003 ± 0.000	0.003 ± 0.000	0.003 ± 0.000	0.003 ± 0.000	\	\	\	\
Network inference									
Dataset	GGN	CoND	CoND ₀	Correlation	Granger	CN	RA	AA	LP
Influenza	0.232 ± 0.047	0.315 ± 0.078	0.259 ± 0.055	0.117 ± 0.032	0.235 ± 0.043	0.283 ± 0.029	0.273 ± 0.035	0.273 ± 0.035	0.250 ± 0.031
PM _{2.5}	0.121 ± 0.022	0.250 ± 0.033	0.311 ± 0.037	0.007 ± 0.001	0.094 ± 0.020	0.115 ± 0.032	0.162 ± 0.025	0.155 ± 0.022	0.074 ± 0.014

TABLE XXI. Comparison of the performance of CoND and other baseline methods with 90% dynamics and 90% structure based on the GIN model by using the influenza data and the PM_{2.5} data.

Dynamical prediction									
Dataset	GGN	CoND	CoND ₀	CoD ₁	CoD ₂	\	\	\	\
Influenza	0.004 ± 0.000	0.001 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	0.002 ± 0.000	\	\	\	\
PM _{2.5}	0.004 ± 0.001	0.002 ± 0.000	0.003 ± 0.0002	0.003 ± 0.000	0.003 ± 0.001	\	\	\	\
Network inference									
Dataset	GGN	CoND	CoND ₀	Correlation	Granger	CN	RA	AA	LP
Influenza	0.235 ± 0.063	0.319 ± 0.045	0.2981 ± 0.051	0.114 ± 0.035	0.229 ± 0.027	0.252 ± 0.018	0.284 ± 0.024	0.273 ± 0.034	0.271 ± 0.023
PM _{2.5}	0.134 ± 0.039	0.284 ± 0.025	0.270 ± 0.020	0.003 ± 0.002	0.095 ± 0.015	0.118 ± 0.026	0.148 ± 0.025	0.137 ± 0.022	0.086 ± 0.019

REFERENCES

- ¹O. Descalzi, S. Curilef, L. Velazquez, and V. Muñoz, “Complex systems and inter/transdisciplinary research: A review,” *Chaos* **34**, 010401 (2024).
- ²C. W. Lynn, L. Papadopoulos, A. E. Kahn, and D. S. Bassett, “Human information processing in complex networks,” *Nat. Phys.* **16**, 965–973 (2020).
- ³M. Nitzan, J. Casadiego, and M. Timme, “Revealing physical interaction networks from statistics of collective dynamics,” *Sci. Adv.* **3**, e1600396 (2017).
- ⁴J. Sun, L. Feng, J. Xie, X. Ma, D. Wang, and Y. Hu, “Revealing the predictability of intrinsic structure in complex networks,” *Nat. Commun.* **11**, 574 (2020).
- ⁵T.-T. Gao and G. Yan, “Autonomous inference of complex network dynamics from incomplete and noisy data,” *Nat. Comput. Sci.* **2**, 160–168 (2022).
- ⁶J. Koch, Z. Chen, A. Tuor, J. Drgona, and D. Vrabie, “Structural inference of networked dynamical systems with universal differential equations,” *Chaos* **33**, 023103 (2023).
- ⁷W.-X. Wang, Y.-C. Lai, and C. Grebogi, “Data based identification and prediction of nonlinear and complex dynamical systems,” *Phys. Rep.* **644**, 1–76 (2016).
- ⁸J. U. Song, K. Choi, S. M. Oh, and B. Kahng, “Exploring nonlinear dynamics and network structures in Kuramoto systems using machine learning approaches,” *Chaos* **33**, 073148 (2023).
- ⁹N. Horsevad, D. Mateo, R. E. Kooij, A. Barrat, and R. Bouffanais, “Transition from simple to complex contagion in collective decision-making,” *Nat. Commun.* **13**, 1442 (2022).
- ¹⁰C. Ma, H. Wang, and H.-F. Zhang, “Reconstructing networks via discrete state dynamical data: A mini-review,” *Europhys. Lett.* **144**, 21002 (2023).
- ¹¹Z.-K. Gao, M. Small, and J. Kurths, “Complex network analysis of time series,” *Europhys. Lett.* **116**, 50001 (2017).
- ¹²A. Sheikhattar, S. Miran, J. Liu, J. B. Fritz, S. A. Shamma, P. O. Kanold, and B. Babadi, “Extracting neuronal functional network dynamics via adaptive Granger causality analysis,” *Proc. Natl. Acad. Sci. U.S.A.* **115**, E3869–E3878 (2018).
- ¹³X. Han, Z. Shen, W.-X. Wang, and Z. Di, “Robust reconstruction of complex networks from sparse data,” *Phys. Rev. Lett.* **114**, 028701 (2015).
- ¹⁴H. Wang, C. Ma, H.-S. Chen, Y.-C. Lai, and H.-F. Zhang, “Full reconstruction of simplicial complexes from binary contagion and Ising data,” *Nat. Commun.* **13**, 3043 (2022).
- ¹⁵C. Ma, Y.-C. Lai, X. Li, and H.-F. Zhang, “General optimization framework for accurate and efficient reconstruction of symmetric complex networks from dynamical data,” *Phys. Rev. E* **108**, 034304 (2023).
- ¹⁶I. Topal and D. Eroglu, “Reconstructing network dynamics of coupled discrete chaotic units from data,” *Phys. Rev. Lett.* **130**, 117401 (2023).
- ¹⁷E. Tan, D. Correa, T. Stemler, and M. Small, “A backpropagation algorithm for inferring disentangled nodal dynamics and connectivity structure of dynamical networks,” *IEEE Trans. Netw. Sci. Eng.* **11**, 613–624 (2024).
- ¹⁸L. Lü and T. Zhou, “Link prediction in complex networks: A survey,” *Physica A* **390**, 1150–1170 (2011).
- ¹⁹A. Ghasemian, H. Hosseinmardi, A. Galstyan, E. M. Airolidi, and A. Clauset, “Stacking models for nearly optimal link prediction in complex networks,” *Proc. Natl. Acad. Sci. U.S.A.* **117**, 23393–23400 (2020).
- ²⁰M. E. Newman, “Clustering and preferential attachment in growing networks,” *Phys. Rev. E* **64**, 025102 (2001).
- ²¹L. A. Adamic and E. Adar, “Friends and neighbors on the web,” *Soc. Netw.* **25**, 211–230 (2003).
- ²²A. Clauset, C. Moore, and M. E. Newman, “Hierarchical structure and the prediction of missing links in networks,” *Nature* **453**, 98–101 (2008).
- ²³Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, “Link prediction in complex networks: A local naive Bayes model,” *Europhys. Lett.* **96**, 48007 (2011).
- ²⁴K. Li, L. Tu, and L. Chai, “Ensemble-model-based link prediction of complex networks,” *Comput. Netw.* **166**, 106978 (2020).
- ²⁵L. Cai, J. Li, J. Wang, and S. Ji, “Line graph neural networks for link prediction,” *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 5103–5113 (2022).
- ²⁶E. Baake, M. Baake, H. Bock, and K. Briggs, “Fitting ordinary differential equations to chaotic data,” *Phys. Rev. A* **45**, 5524 (1992).
- ²⁷C. Yao and E. M. Bollt, “Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems,” *Physica D* **227**, 78–99 (2007).
- ²⁸E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory* **52**, 489–509 (2006).
- ²⁹Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H. T. Zhang, H. Ding, and J. Goncalves, “Data driven discovery of cyber physical systems,” *Nat. Commun.* **10**, 4894 (2019).
- ³⁰X. Xia, Y. Su, L. Lü, X. Zhang, Y. C. Lai, and H. F. Zhang, “Machine learning prediction of network dynamics with privacy protection,” *Phys. Rev. Res.* **4**, 043076 (2022).
- ³¹C. Murphy, E. Laurence, and A. Allard, “Deep learning of contagion dynamics on complex networks,” *Nat. Commun.* **12**, 4720 (2021).
- ³²H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang, and S. Y. Philip, “Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting,” *Inf. Sci.* **521**, 277–290 (2020).
- ³³B. Jhun, H. Choi, Y. Lee, J. Lee, C. H. Kim, and B. Kahng, “Prediction and mitigation of nonlocal cascading failures using graph neural networks,” *Chaos* **33**, 013115 (2023).
- ³⁴J. Han, H. Liu, H. Zhu, H. Xiong, and D. Dou, “Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, 2021), Vol. 35, pp. 4081–4089.
- ³⁵M. Chen, Y. Zhang, Z. Zhang, L. Du, S. Wang, and J. Zhang, “Inferring network structure with unobservable nodes from time series data,” *Chaos* **32**, 013126 (2022).
- ³⁶M. Chen, J. Zhang, Z. Zhang, L. Du, Q. Hu, S. Wang, and J. Zhu, “Inference for network structure and dynamics from time series data via graph neural network,” *arXiv:2001.06576* (2020).
- ³⁷Y. Zhang, Y. Guo, Z. Zhang, M. Chen, S. Wang, and J. Zhang, “Universal framework for reconstructing complex networks and node dynamics from discrete or continuous dynamics data,” *Phys. Rev. E* **106**, 034315 (2022).
- ³⁸T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, “Neural relational inference for interacting systems,” in *Proceedings of the International Conference on Machine Learning* (PMLR, 2018), pp. 2688–2697.
- ³⁹R. Li, S. Wang, F. Zhu, and J. Huang, “Adaptive graph convolutional neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, 2018), Vol. 32.
- ⁴⁰T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proceedings of the International Conference on Learning Representations* (OpenReview, 2017).
- ⁴¹W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS’17)* (Curran Associates Inc., Red Hook, NY, 2017), pp. 1025–1035.
- ⁴²K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?,” in *Proceedings of the International Conference on Learning Representations* (OpenReview, 2019).
- ⁴³A.-L. Barabási, “Scale-free networks: A decade and beyond,” *Science* **325**, 412–413 (2009).
- ⁴⁴M. E. Newman and D. J. Watts, “Renormalization group analysis of the small-world network model,” *Phys. Lett. A* **263**, 341–346 (1999).
- ⁴⁵J. Gómez-Gardeñes and Y. Moreno, “From scale-free to Erdos-Rényi networks,” *Phys. Rev. E* **73**, 056124 (2006).
- ⁴⁶M. E. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Phys. Rev. E* **74**, 036104 (2006).
- ⁴⁷R. Rossi and N. Ahmed, “The network data repository with interactive graph analytics and visualization,” in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, 2015), Vol. 29.
- ⁴⁸R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas, “Self-similar community structure in a network of human interactions,” *Phys. Rev. E* **68**, 065103 (2003).
- ⁴⁹M. R. Sanatkar, W. N. White, B. Natarajan, C. M. Scoglio, and K. A. Garrett, “Epidemic threshold of an SIS model in dynamic switching networks,” *IEEE Trans. Syst. Man Cybern. Syst.* **46**, 345–355 (2015).
- ⁵⁰F. S. N. Karan and S. Chakraborty, “Dynamics of a repulsive voter model,” *IEEE Trans. Comput. Soc. Syst.* **3**, 13–22 (2016).

- ⁵¹A. Y. Lohkov, M. Vuffray, S. Misra, and M. Chertkov, "Optimal structure and parameter learning of Ising models," *Sci. Adv.* **4**, e1700791 (2018).
- ⁵²B. Shulgin, L. Stone, and Z. Agur, "Pulse vaccination strategy in the SIR epidemic model," *Bull. Math. Biol.* **60**, 1123–1148 (1998).
- ⁵³F. A. Rodrigues, T. K. D. Peron, P. Ji, and J. Kurths, "The Kuramoto model in complex networks," *Phys. Rep.* **610**, 1–98 (2016).
- ⁵⁴A. Levina and V. Priesemann, "Subsampling scaling," *Nat. Commun.* **8**, 1–9 (2017).
- ⁵⁵K. Kaneko, "Overview of coupled map lattices," *Chaos* **2**, 279–282 (1992).
- ⁵⁶T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *Eur. Phys. J. B* **71**, 623–630 (2009).
- ⁵⁷L. Lü, C.-H. Jin, and T. Zhou, "Similarity index based on local paths for link prediction of complex networks," *Phys. Rev. E* **80**, 046122 (2009).
- ⁵⁸J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz-Marí, and E. H. Van Nes, "Inferring causation from time series in earth system sciences," *Nat. Commun.* **10**, 1–13 (2019).
- ⁵⁹S. Pei, S. Kandula, W. Yang, and J. Shaman, "Forecasting the spatial transmission of influenza in the United States," *Proc. Natl. Acad. Sci. U.S.A.* **115**, 2752–2757 (2018).
- ⁶⁰S. Wang, Y. Li, J. Zhang, Q. Meng, L. Meng, and F. Gao, "PM2.5-GNN: A domain knowledge enhanced graph neural network for PM2.5 forecasting," in *Proceedings of the International Conference on Advances in Geographic Information Systems* (ACM, 2020), pp. 163–166.
- ⁶¹M. M. Danziger and A. -L. Barabási, "Recovery coupling in multilayer networks," *Nat. Commun.* **13**, 955 (2022).
- ⁶²P. Holme and J. Saramäki, "Temporal networks," *Phys. Rep.* **519**, 97–125 (2012).
- ⁶³S. Boccaletti, P. De Lellis, C. I. Del Genio, K. Alfaro-Bittner, R. Criado, S. Jalan, and M. Romance, "The structure and dynamics of networks with higher order interactions," *Phys. Rep.* **1018**, 1–64 (2023).
- ⁶⁴P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Ann. Oper. Res.* **134**, 19–67 (2005).
- ⁶⁵W. James and C. Stein, "Estimation with quadratic loss," in *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics Probability* (Springer, 1960).
- ⁶⁶See <https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html/> for information about "The ILI data of USA."
- ⁶⁷See <https://www.census.gov/data/tables/2015/demo/metro-micro/commuting-flows-2015.html/> for information about "The county-to-county commuting data of USA."
- ⁶⁸Y. Qi, Q. Li, H. Karimian, and D. Liu, "A hybrid model for spatiotemporal forecasting of PM_{2.5} based on graph convolutional neural network and long short-term memory," *Sci. Total Environ.* **664**, 1–10 (2019).