SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2024, Vol. 67, Iss. 12, 222206:1–222206:16
https://doi.org/10.1007/s11432-023-4216-y

Learning continuous network emerging dynamics from scarce observations via data-adaptive stochastic processes

Jiaxu CUI[1,2], Qipeng WANG[1,2], Bingyi SUN[2,3], Jiming LIU[4] & Bo YANG[1,2*]

[1]*College of Computer Science and Technology, Jilin University, Changchun* 130012*, China;*
[2]*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun* 130012*, China;*
[3]*Public Computer Education and Research Center, Jilin University, Changchun* 130012*, China;*
[4]*Department of Computer Science, Hong Kong Baptist University, Hong Kong* 999077*, China*

Received 24 November 2023/Revised 4 March 2024/Accepted 4 June 2024/Published online 9 December 2024

Abstract Learning network dynamics from the empirical structure and spatio-temporal observation data is crucial to revealing the interaction mechanisms of complex networks in a wide range of domains. However, most existing methods only aim at learning network dynamic behaviors generated by a specific ordinary differential equation instance, resulting in ineffectiveness for new ones, and generally require dense observations. The observed data, especially from network emerging dynamics, are usually difficult to obtain, which brings trouble to model learning. Therefore, learning accurate network dynamics with sparse, irregularly-sampled, partial, and noisy observations remains a fundamental challenge. We introduce a new concept of the stochastic skeleton and its neural implementation, i.e., neural ODE processes for network dynamics (NDP4ND), a new class of stochastic processes governed by stochastic data-adaptive network dynamics, to overcome the challenge and learn continuous network dynamics from scarce observations. Intensive experiments conducted on various network dynamics in ecological population evolution, phototaxis movement, brain activity, epidemic spreading, and real-world empirical systems, demonstrate that the proposed method has excellent data adaptability and computational efficiency, and can adapt to unseen network emerging dynamics, producing accurate interpolation and extrapolation with reducing the ratio of required observation data to only about 6% and improving the learning speed for new dynamics by three orders of magnitude.

Keywords complex networks, network dynamics, emerging spatio-temporal dynamics, neural processes

1 Introduction

Learning the underlying dynamics mechanism of complex systems has been a fundamental research [1], which almost dominates the understanding for the formation and evolution of complex phenomenons with significant impacts in a wide range of domains, such as the global climate anomalies [2], the emergence of public opinion influencing political and economic pattern [3], as well as the epidemic outbreaks endangering the lives and health of billions of people [4]. As a superior modeling tool, the network can be used to depict spatio-temporal interactions among system components, where network dynamics refers to the complex interaction mechanism [5–7]. Therefore, learning network dynamics from empirical observations that capture the topological structure and spatio-temporal behavior is crucial to revealing the interaction mechanism and predicting the system behaviors.

In network dynamics, a tough issue is to learn network emerging dynamics [7,8]. For instance, as a kind of impressive network emerging dynamics in public health [9], the emerging infectious diseases (EID) [8,10] are usually caused by new or newly identified pathogenic microorganisms, e.g., SARS-CoV, MERS-CoV, and SARS-CoV-2 [11], giving rise to the novel exposed dynamics behaviors [4]. The task poses particular challenges, i.e., the novelty and data sparsity. The novelty is reflected in the undiscovered and new formulas or parameters of the dynamics equations, resulting in the inapplicability of the existing models.

* Corresponding author (email: ybo@jlu.edu.cn)

Due to the fact that data acquisition is costly and time-consuming [12], attention to early transmission is inadequate [13], and broken sensors or damaged memory units may occur frequently [14], observed data is usually scarce, irregularly-sampled, partial and noisy, bringing troubles to model learning.

Fortunately, artificial intelligence technologies have gradually led scientific discovery, breaking through protein structure analysis [15], assisting in the design of new drugs [16], and advancing mathematics [17], which also provides a bright road for learning network dynamics. Representative symbolic regression-based network dynamics studies have emerged [18] to actively promote the development of complexity and network sciences. However, the performance of these methods is strictly governed by expert knowledge, such as the selection of basis functions [18, 19].

With the rapid development of deep learning, researchers have developed effective neural simulators for networked systems [20–30]. Modeling ordinary differential equations (ODEs) on networks for characterizing propagation mechanisms through neural networks, these methods are able to automatically learn expressive network dynamics via a data-driven fashion. Recently, few attempts to learn continuous-time network dynamics from irregularly-sampled partial observations have been made by elaborately designing a spatio-temporal transformer to aggregate observations [28] and a graph neural network as the ODE function to model continuous network interactions [29, 30]. Despite extending to promising dynamic graphs [24], across environment learning [27] and brain networks [25], they still struggle with the relatively dense observations required for sufficient training, i.e., at least 40%–80% of all data is required as observations for model learning [28–30]. Moreover, these methods only learn the network dynamics behavior generated by a single network ODE instance, resulting in ineffective processing for new behaviors with few observations.

To address the aforementioned limitations, we introduce a novel concept of the stochastic skeleton to model network dynamics from the non-deterministic perspective and propose its neural implementation, i.e., neural ODE processes for network dynamics (NDP4ND), a new family of stochastic processes governed by stochastic data-adaptive network dynamics, simultaneously considering temporal dynamics and topological interactions. Our analysis on various complex network dynamics scenarios, including mutualistic interaction dynamics in ecosystems, second-order phototaxis dynamics, brain dynamics, compartment models in epidemiology, and real-world global epidemic transmission systems, indicates that the NDP4ND has data adaptability and computational efficiency, and can accurately interpolate and extrapolate the unseen emerging dynamics with sparse, irregularly-sampled, partial, and noisy observations.

2 Background and problem statement

2.1 Network dynamics

A networked system consists of two key components [5, 6]: network dynamics $\pi_\phi \in \Pi$ and network topology $A \in \mathcal{A}$, where Π is a network dynamics space and $\mathcal{A}$ is a topological structure space. Generally, π_ϕ can be considered as an ODE instance, i.e., $\frac{\mathrm{d}X_l(t)}{\mathrm{d}t} = \pi_\phi(\{X_j(t)\}_{j=1:n}, A_{l,1:n})$, where, $X_l(t) \in \mathbb{R}^d$ represents the states on node l at time t, d is the state dimension, and n is the system size. $A_{l,j}$ declares the influence or flow from node j to l. ϕ is an assignment of parameters in an ODE template π, which is a parameterized ODE characterizing continuous state changes. Assigning different values to the parameters in π will result in corresponding network ODE instances that produce various system behaviors. Given π_ϕ and A, system behaviors can be characterized by the spatio-temporal trajectory function $\omega \in \Omega$, where, Ω is a trajectory function space. When the initial state $X_l(0)$ is known, ω can be obtained by calculating the state at any time t by solving an ODE initial-value problem (IVP) using numerical integration [31], i.e., $X_l(t) = X_l(0) + \int_0^t \pi_\phi(\{X_j(\tau)\}_{j=1:n}, A_{l,1:n})\mathrm{d}\tau$. As π_ϕ and A in the trajectory function govern state changes together, the distinguishing feature of network dynamics is that the updating of states on the networked system is influenced by adjacent nodes (see Figure 1(a)). With appropriate choices of the functions f and g, the following skeleton of the dynamical equation can describe a broad range of complex networked systems [6], $\dot{X}_l(t) = f(X_l(t)) + \sum_j A_{l,j} g(X_l(t), X_j(t))$, where, f is self dynamics and g is interaction dynamics. The former quantifies the impact of one's own state on change, while the latter quantifies the impact of neighboring states on change. The skeleton has been empirically applied to guide the reconstruction of governing equations of network dynamics [18].

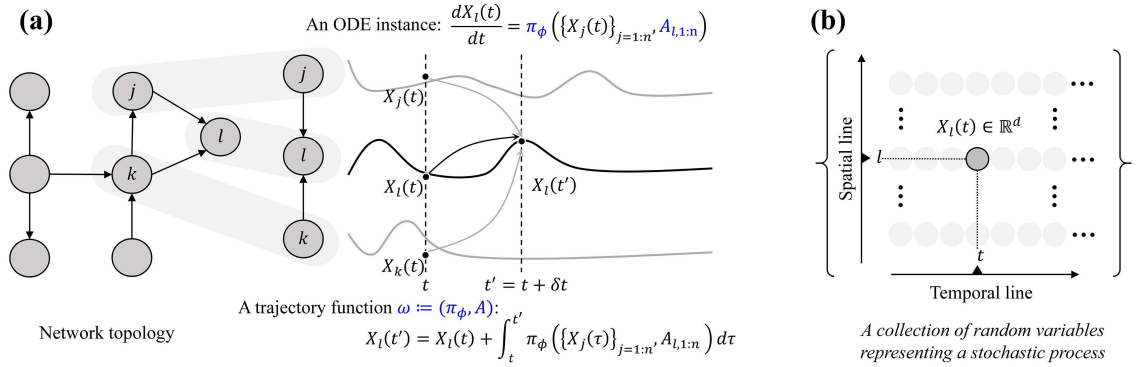

Figure 1 (Color online) (a) The updating of states on a networked system is influenced by neighbors, where network topology A controls the interactions. The state changes on node l can be modeled by an ODE instance, i.e., $\frac{\mathrm{d}X_l(t)}{\mathrm{d}t} = \pi_\phi(\{X_j(t)\}_{j=1:n}, A_{l,1:n})$. A trajectory function ω dominated by π_ϕ and A can be obtained by numerical integration, i.e., $X_l(t+\delta t) = X_l(t) + \int_t^{t+\delta t} \pi_\phi(\{X_j(\tau)\}_{j=1:n}, A_{l,1:n})\mathrm{d}\tau$. (b) We model a collection of $\{X_l(t)\}$ as a stochastic process, where the state $X_l(t)$ on node l at time t is a random variable. In principle, as time and space (network) expand, there are infinitely many such random variables in a networked system.

2.2 Neural ODE

Neural ODEs refer to a technique that parameterizes state differentiation via neural networks [32], i.e., $\dot{X}(t) = \mathrm{NN}(X(t), t)$, and uses the adjoint method [32] to solve the IVP to ensure that the model can effectively perform backpropagation to learn parameters of neural networks. They usually encode the original state into the hidden space, parameterize the differentiation of the hidden state, and then obtain the future state by performing the integration and decoding operations, i.e., $L(0) = \mathrm{enc}(X(0))$, $L(t') = L(0) + \int_{t_0}^{t'} \mathrm{NN}(L(t), t)\mathrm{d}t$, $X(t') = \mathrm{dec}(L(t'))$, where, $X(0)$ is a shorthand for the initial state $X(t_0)$, $L(0)$ is the initial hidden state, and t' is the target time. Note that enc and dec can also be neural networks. It has been observed that the way to learn dynamics after mapping into hidden space has excellent performance [29,33] and can express more complex original dynamics [34]. In network dynamics, effective network encoding can reduce complex spatio-temporal patterns to simple [35] and homogeneous dynamic patterns [36], e.g., the global spread of epidemics and opinions.

To process continuous time series and irregular-sampled observations, using ODEs parameterized by graph neural networks is proposed to model interacting temporal dynamics [37–40]. Recently, neural ODEs on complex networks were explicitly proposed to automatically model network dynamics [23, 26, 29, 30], extended to promising dynamic network setting [24], across environment learning [27] and brain networks [25], However, they still struggle with the relatively dense observations required for sufficient training, i.e., at least 40%–80% of all data is required as observations for model learning [28–30]. More importantly, they only learn network dynamic behaviors generated by a single network ODE instance, resulting in ineffective processing for new system behaviors with few observations.

2.3 Neural processes

Neural processes (NPs) are neural network-parameterized stochastic processes [41], which can model a random function $F : \mathcal{T} \to \mathcal{X} \subseteq \mathbb{R}^d$, i.e., $F(\cdot) = \mathrm{NN}(\cdot, z)$. The neural network (NN) determines the internal statistical behavior of the stochastic process and the global latent random vector z controls the uncertainty, i.e., generating different functions. Note that while the input and output domains of the random function F could be arbitrary, we are more interested in time series on networks, hereby using time as the input and system states as the output. NPs thus can naturally be used to model time series. The global latent random vector z comes from a data-adaptative conditional distribution, i.e., $p(z|\mathcal{D}_\mathbb{C})$, where $\mathcal{D}_\mathbb{C} = \{(t^\mathbb{C}_{1:N_\mathbb{C}}, X(t^\mathbb{C}_{1:N_\mathbb{C}}))\}$ is the context set. Given the target set $t^\mathbb{T}_{1:N_\mathbb{T}}, X(t^\mathbb{T}_{1:N_\mathbb{T}})$, the generative process behind the NPs is $p(X(t^\mathbb{T}_{1:N_\mathbb{T}}), z|t^\mathbb{T}_{1:N_\mathbb{T}}, \mathcal{D}_\mathbb{C}) = p(z|\mathcal{D}_\mathbb{C}) \prod_{i=1}^{N_\mathbb{T}} \mathcal{N}(X(t^\mathbb{T}_i)|\mathrm{NN}(t^\mathbb{T}_i, z), \sigma^2)$, where, $\mathcal{N}$ is the normal distribution and σ is the system noise. By approximating $p(z|\mathcal{D}_\mathbb{C})$ with a neural network-parameterized encoder $q(z|\mathcal{D}_\mathbb{C})$, the parameters in the NPs can be optimized via an amortized variational inference procedure. In principle, NPs can cover broad stochastic processes derived from neural network-based parameterization which provides an interface for learning from data, and retain excellent computability via efficient forward propagation calculations during inference. Due to the strong data adaptability and

computational efficiency, its advanced versions have been developed, incorporating application-specific inductive biases of attention mechanism [42] and translation equivariance [43]. A small number of studies used graphs to model dependencies among inputs [44] and incorporated graph convolutional networks into the NP architecture to improve the performance of traditional graph learning tasks, including node classification [45], edge imputation [46], and link predictions [47]. But they cannot model and deal with temporal dynamics. Although a class of neural ODE processes (NDPs) [33] generalized NPs defined over time by combining latent neural ODEs with NPs, it only aims at temporal dynamics and ignores spatial interactions between temporal dynamics on networks, resulting in insufficient behavior tracking for network dynamics learning.

2.4 Problem statement

We consider learning the network dynamics based on empirical observations capturing the structure and spatio-temporal behavior generated from complex networked systems. Denote an observation on spatio-temporal series produced by a networked system as a triple $(t, l, X_l(t))$, where $t \in \mathbb{R}$, $l \in \{1, 2, \ldots, n\}$, and n is the number of nodes on the interested system. Formally, given a network topology A of a complex networked system and a set of observations of the spatio-temporal series produced by the system, i.e., $\mathcal{D}_{\mathbb{C}} = \{(t_1^{\mathbb{C}}, l_1^{\mathbb{C}}, X_{l_1^{\mathbb{C}}}(t_1^{\mathbb{C}})), \ldots, (t_{N_{\mathbb{C}}}^{\mathbb{C}}, l_{N_{\mathbb{C}}}^{\mathbb{C}}, X_{l_{N_{\mathbb{C}}}^{\mathbb{C}}}(t_{N_{\mathbb{C}}}^{\mathbb{C}}))\}$, where $N_{\mathbb{C}}$ is the number of observations, learning network dynamics refers to forecasting the state on any node l at any time t, i.e., $X_l(t)$, based on empirical observations capturing the structure and spatio-temporal behavior. Let T_o denote the maximum value of all observed times in $\mathcal{D}_{\mathbb{C}}$. When all t in the prediction task is less than or equal to T_o, it becomes an interpolation task, and while all t in the prediction task is greater than T_o, then it is an extrapolation task. When $\mathcal{D}_{\mathbb{C}}$ comes from a network dynamics with a new ODE instance and its involving observations are sparse, irregularly-sampled, partial, and noisy, it can be called a network emerging dynamics learning problem that we are concerned about.

3 NDP4ND: neural ODE processes for network dynamics

Abundant networked systems jointly produce a spatio-temporal trajectory function space of network dynamics, i.e., Ω. Let $\omega := (\pi_\phi, A) \in \Omega$ denote a trajectory function on a networked system, describing the spatio-temporal series of system behavior. Since π_ϕ and A in trajectory function govern state changes together, we can obtain the state at any time t' through numerical integration, as shown in Figure 1(a). As noted, most of the existing methods only learn a specific trajectory sample in trajectory function space and are tricky to generalize to other trajectory samples from new network ODE or emerging empirical systems. On the contrary, we expect our method to represent the trajectory function space and adapt to unseen tasks with few observations. To achieve this goal, our core idea is to use stochastic processes to model the trajectory function space Ω. Specifically, we propose NDP4ND, a new family of stochastic processes parameterized by neural networks, simultaneously considering temporal dynamics and node interactions on topological structures, to approximate random spatio-temporal trajectory functions on networks. By properly combining the universal skeleton of network dynamics [6], neural ODE [32], and NPs [41] to build the family of distributions suitable for network dynamics modeling, the NDP4ND maintains their respective strengths, including broad network dynamics representation, continuous temporal changes, and data-adaptation.

3.1 Constructing stochastic processes to represent trajectory function space

Although one can independently model a stochastic process for the trajectory per node, due to the fact that nodes in the networked system interact with others, there are often important correlations between nodes. We, thus, establish a general stochastic process ($\mathcal{F}$) to model the complex trajectory function space, while considering the correlation between the states of adjacent nodes. Specifically, we model a collection $\{X_l(t)\}$ as a stochastic process, where the node state $X_l(t) \in \mathbb{R}^d$ is a random variable. In principle, as time and space (network) expand, there are infinitely many such random variables in the networked system, as shown in Figure 1(b). To build the stochastic processes, the critical issue we face is how to construct a family of finite-dimensional distributions to completely characterize the statistical relationship among any finite set of these random variables.

3.1.1 *Family of finite-dimensional distributions*

Since the NDP4ND should model the dynamic changes of node states on the networked systems, we properly combine the universal skeleton of network dynamics [6], neural ODE [32], and NPs [41] to build its family of distributions suitable for network dynamics modeling, retaining advantages including broad network interactions, temporal changes, and data-adaptation. Please see the probabilistic graphical model of the generative process behind NDP4ND in Figure A1 in Appendix A of Supporting information.

Given a network A and a context set $\mathcal{D}_{\mathbb{C}}$, the generative model can be formulated as

$$p_{X,z,L(0)} = p(z|\mathcal{D}_{\mathbb{C}}, A)p(L(0)|\mathcal{D}_{\mathbb{C}}) \prod_{i=1}^{N_{\mathbb{T}}} p(X_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}})|t_i^{\mathbb{T}}, l_i^{\mathbb{T}}, z, L(0), A), \tag{1}$$

where $p_{X,z,L(0)}$ is a shorthand for the generative model $p(X_{l^{\mathbb{T}}_{1:N_{\mathbb{T}}}}(t^{\mathbb{T}}_{1:N_{\mathbb{T}}}), z, L(0)|t^{\mathbb{T}}_{1:N_{\mathbb{T}}}, l^{\mathbb{T}}_{1:N_{\mathbb{T}}}, \mathcal{D}_{\mathbb{C}}, A)$, $t^{\mathbb{T}}_{1:N_{\mathbb{T}}}$, $l^{\mathbb{T}}_{1:N_{\mathbb{T}}}$, and $X_{l^{\mathbb{T}}_{1:N_{\mathbb{T}}}}(t^{\mathbb{T}}_{1:N_{\mathbb{T}}})$ are the target set, $L(0)$ and z denote the initial states in latent space and the global random vector that can control the network ODE, respectively.

Modeling distributions $p(z|\mathcal{D}_{\mathbb{C}}, A)$ **and** $p(L(0)|\mathcal{D}_{\mathbb{C}})$. We encode network topology and context set into two latent variables, i.e., $z \sim q(z|\mathcal{D}_{\mathbb{C}}, A)$ and $L(0) \sim q(L(0)|\mathcal{D}_{\mathbb{C}})$. To parameterize the distribution of z, we first integrate observations with the same time in the contexts and combine them with the topology to form a graph, i.e., $G(t) = (X(t), M(t), A)$, where $X(t) \in \mathbb{R}^{n\times d}$ stores the all observed states at time t and the mask $M(t)$ has the same shape as $X(t)$, indicating which nodes have been observed (its value on the corresponding position is set to 1, otherwise it is 0). The context set $\mathcal{D}_{\mathbb{C}}$ is then transformed into a graph set $\{(t_1, G(t_1)), \ldots, (t_K, G(t_K))\}$, where K is consistent with the number of t in contexts that eliminate duplicates. We use a neural network $\boldsymbol{\varphi}$ to produce a representation for each pair in the graph set, i.e., $r_k = \boldsymbol{\varphi}(t_k, G(t_k))$, and then use another neural network $\boldsymbol{\rho}$ to give the distribution of z by $q(z|\mathcal{D}_{\mathbb{C}}, A) = \mathcal{N}(\mu_z, \mathrm{diag}(\sigma_z^2))$, where $[\mu_z; \sigma_z^2] = \boldsymbol{\rho}(r)$, $r = \mathrm{agg}(\{r_k\}_{k=1}^K)$, and agg is an aggregation operation with permutation invariance, such as element-wise mean. Although the distribution of $L(0)$ can also be obtained through the above fashion, we handle it in a simpler way when the initial states are always known, i.e., $q(L(0)|\mathcal{D}_{\mathbb{C}}) = q(L(0)|X(0)) = \prod_{l=1}^n \mathcal{N}(\mu_{L_l(0)}, \mathrm{diag}(\sigma^2_{L_l(0)}))$, where $[\mu_{L_l(0)}; \sigma^2_{L_l(0)}] = \boldsymbol{e}(X_l(0))$ and function $\boldsymbol{e}$ is as a neural network for encoding each initial state at any node l.

Modeling distribution $p(X_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}})|t_i^{\mathbb{T}}, l_i^{\mathbb{T}}, z, L(0), A)$. After obtaining the distributions of z and $L(0)$, we perform dynamic propagation in the hidden space with initial hidden state $L(0)$ based on the universal skeleton of network dynamics as follows:

$$L_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}}) = L_{l_i^{\mathbb{T}}}(0) + \int_{t_0}^{t_i^{\mathbb{T}}} \left(\boldsymbol{S}(L_{l_i^{\mathbb{T}}}(t), \tilde{z}) + \sum_{j=1}^{n} A_{l_i^{\mathbb{T}}, l_j^{\mathbb{T}}} \boldsymbol{I}(L_{l_i^{\mathbb{T}}}(t), L_{l_j^{\mathbb{T}}}(t), \tilde{z}) \right) \mathrm{d}t, \tag{2}$$

where $\boldsymbol{S}$ and $\boldsymbol{I}$ are neural networks, representing self dynamics and interaction dynamics, respectively. Note that the dynamic process is mainly controlled by $\tilde{z} = [z; r]$, where r transmits the context information to the propagation process from a deterministic path, and random variables z and $L(0)$ jointly characterize the uncertainty of underlying network dynamics. Given $\tilde{z}$ and $L_{l_i^{\mathbb{T}}}(0)$, the evolved latent state $L_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}})$ can be seen as a deterministic function dominated by $t_i^{\mathbb{T}}$ and $l_i^{\mathbb{T}}$, involving the interaction between nodes. Assuming that output states are noisy, for a given $L_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}})$, we can decode it into the predictive state by $X_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}}) \sim p(X_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}})|t_i^{\mathbb{T}}, l_i^{\mathbb{T}}, z, L(0), A) = \mathcal{N}(\mu_{X_{l_i^{\mathbb{T}}}}(t_i^{\mathbb{T}}), \sigma^2_{X_{l_i^{\mathbb{T}}}}(t_i^{\mathbb{T}}))$, where $[\mu_{X_{l_i^{\mathbb{T}}}}(t_i^{\mathbb{T}}); \sigma^2_{X_{l_i^{\mathbb{T}}}}(t_i^{\mathbb{T}})] = \boldsymbol{d}(t_i^{\mathbb{T}}, L_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}}), \tilde{z})$ and $\boldsymbol{d}$ is a neural network.

3.1.2 *Theoretical existence*

We state that the constructed family of finite-dimensional distributions satisfies the exchangeability and consistency conditions (see Proposition 1). The Kolmogorov extension theorem guarantees that these conditions are sufficient to define a stochastic process [48]. In other words, the stochastic process we established exists, and the family of distributions is just it.

Proposition 1. NDP4ND satisfies the exchangeability and consistency conditions.

The detailed proof can be found in Appendix A.2 of Supporting information.

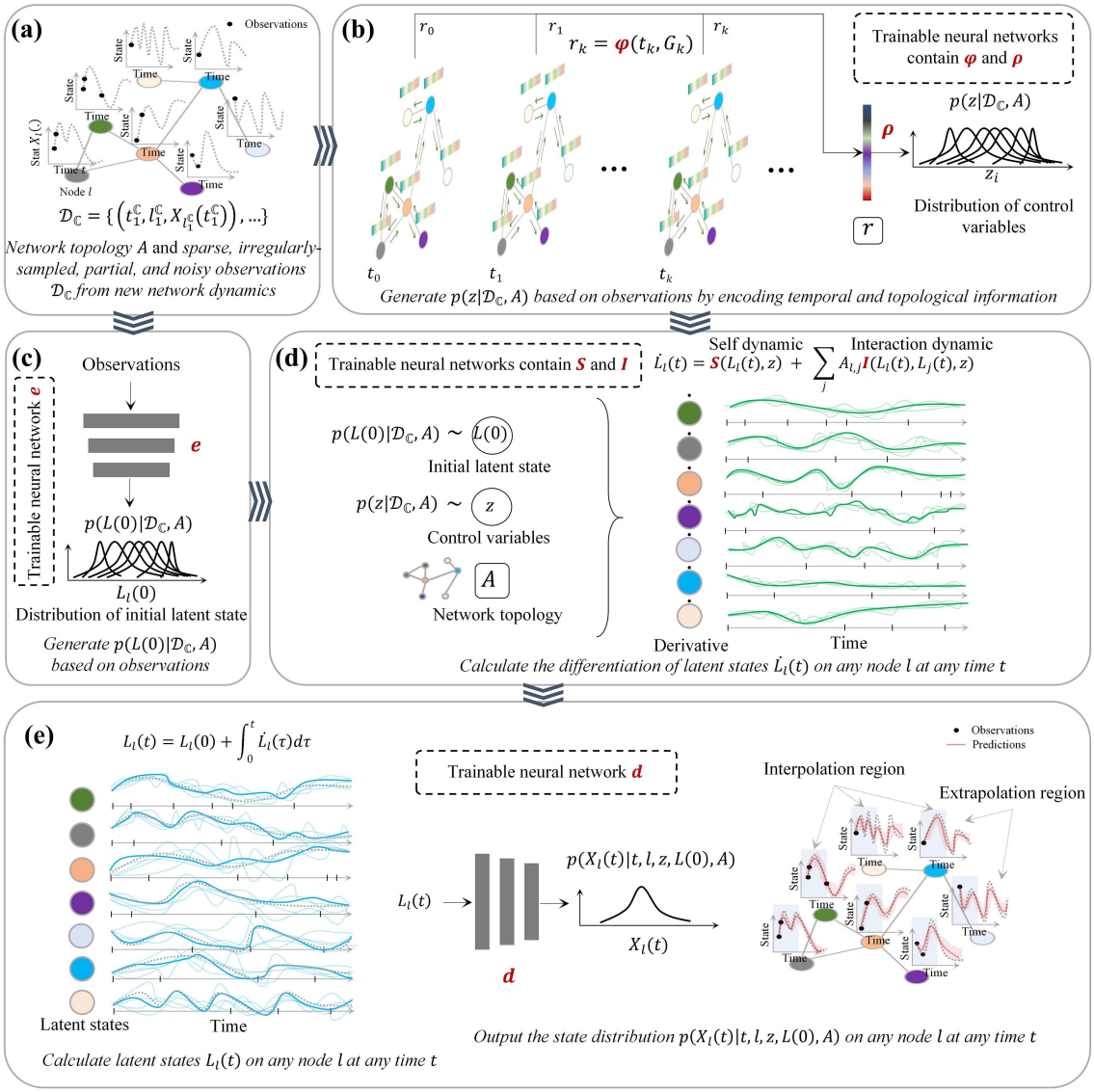

Figure 2 (Color online) Overall workflow of the NDP4ND. (a) Empirical data from network dynamics, including network topology A and state observations $\mathcal{D}_{\mathbb{C}}$. (b) Generating the distribution over the global random vector z, i.e., $p(z|\mathcal{D}_{\mathbb{C}}, A)$ by encoding temporal and topological information. (c) Generating the distribution over the initial states in latent space, i.e., $p(L(0)|\mathcal{D}_{\mathbb{C}}, A)$. (d) After obtaining the distributions of z and $L(0)$, we perform dynamic propagation in the hidden space based on the universal skeleton of network dynamics and give the differentiation of hidden states on node l at time t, i.e., $\dot{L}_l(t)$. (e) Calculating the hidden states $L_l(t)$ and then output the state distribution on any node l at any time t, i.e., $p(X_l(t)|t, l, z, L(0), A)$. Note that neural networks ($\boldsymbol{\varphi}$, $\boldsymbol{\rho}$, $\boldsymbol{e}$, $\boldsymbol{S}$, $\boldsymbol{I}$, $\boldsymbol{d}$) are to parameterize the stochastic processes and achieve adaptive learning from data.

3.2 The workflow of the NDP4ND

Figure 2 shows the overall workflow of the NDP4ND. After obtaining the empirical data from network emerging dynamics (Figure 2(a)), including network topology A and sparse, irregularly-sampled, partial, and noisy observations $\mathcal{D}_{\mathbb{C}}$, we can infer a distribution over the global random vector z by encoding temporal and topological information (Figure 2(b)), i.e., $p(z|\mathcal{D}_{\mathbb{C}}, A)$. By integrating observations with the same time and combining them with the topology to form a graph $G(t)$, the observation set $\mathcal{D}_{\mathbb{C}}$ can be then transformed into a graph set $\{(t_1, G(t_1)), \ldots, (t_K, G(t_K))\}$. We use $\boldsymbol{\varphi}$ to produce a representation for each pair in the graph set by $r_k = \boldsymbol{\varphi}(t_k, G(t_k))$, and then use $\boldsymbol{\rho}$ to give the distribution of z based on an aggregation of all representations, i.e., $r = \text{agg}(\{r_k\}_{k=1}^K)$, where, agg is an aggregation operation with permutation invariance. Simultaneously, the distribution over the initial states in latent space, i.e., $p(L(0)|\mathcal{D}_{\mathbb{C}}, A)$, can be generated by a neural network $\boldsymbol{e}$ based on empirical data (Figure 2(c)). After inferring the distributions of z and $L(0)$, we perform dynamic propagation in the hidden space based

on the universal skeleton of network dynamics and give the differentiation of hidden states on node l at time t (Figure 2(d)), i.e., $\dot{L}_l(t) = \boldsymbol{S}(L_l(t), z) + \sum_{j=1}^{n} A_{l,j}\boldsymbol{I}(L_l(t), L_j(t), z)$, where $\boldsymbol{S}$ and $\boldsymbol{I}$ are neural networks, representing self dynamics and interaction dynamics respectively, and global random vector z can stochastically control the derivative function of network dynamics. Then, we can calculate the hidden states $L_l(t)$ by $L_l(t) = L_l(0) + \int_0^t \dot{L}_l(\tau)\mathrm{d}\tau$ and use neural network $\boldsymbol{d}$ to output the state distribution on any node l at any time t, i.e., $p(X_l(t)|t, l, z, L(0), A)$ (Figure 2(e)). These neural networks ($\boldsymbol{\varphi}$, $\boldsymbol{\rho}$, $\boldsymbol{e}$, $\boldsymbol{S}$, $\boldsymbol{I}$, $\boldsymbol{d}$) are to parameterize the stochastic processes and achieve adaptive learning from data. Their specific architectures in the NDP4ND can be found in Appendix A.1.1 of Supporting information.

3.3 Training and prediction

To learn the distribution over trajectory functions, which is equivalent to learning the parameters of the neural networks in the NDP4ND, we train the model on a set of state observations generated from B different network dynamics with $A^{(1)}, \ldots, A^{(B)}$ topological structures. Also, we split the observations into B context sets $\mathcal{D}_{\mathbb{C}}^{1:B}$ and B target sets $\mathcal{D}_{\mathbb{T}}^{1:B}$. $\mathcal{D}_{\mathbb{C}}^{b}$ is usually a subset of $\mathcal{D}_{\mathbb{T}}^{b}$, where $b \in \{1, 2, \ldots, B\}$. Since the generative process (1) of the NDP4ND contains multiple highly non-linear neural networks, the true posterior is intractable. We, thus, follow [33, 41, 49] and use an amortized variational inference to train the parameters by reparametrization trick and stochastic gradient descent. The derived total training loss under all observations is as follows:

$$\mathcal{L} = \frac{1}{B}\sum_{b=1}^{B}\left[\mathbb{E}_{q_{z,L(0)}}\left(\sum_{i=1}^{N_{\mathbb{T}}^{b}} -\log p\left(X_{l_i^{\mathbb{T}_b}}(t_i^{\mathbb{T}_b})|t_i^{\mathbb{T}_b}, l_i^{\mathbb{T}_b}, z, L(0), A^{(b)}\right)\right) + \beta\mathrm{KL}(q(z|\mathcal{D}_{\mathbb{T}}^{b})\|q(z|\mathcal{D}_{\mathbb{C}}^{b}))\right], \quad (3)$$

where $q_{z,L(0)} = q(z|\mathcal{D}_{\mathbb{T}}^{b}, A^{(b)})q(L(0)|\mathcal{D}_{\mathbb{T}}^{b})$ and β is a tunable parameter to relieve the KL vanishing.

Given the learned NDP4ND, a network topology A, and a set of state observations $\mathcal{D}_{\mathbb{C}}$ from the task at hand, we obtain the predictive distribution of the states on any node $l^{\mathbb{T}}$ at any time $t^{\mathbb{T}}$ as

$$p(X_{l^{\mathbb{T}}}(t^{\mathbb{T}})|t^{\mathbb{T}}, l^{\mathbb{T}}, \mathcal{D}_{\mathbb{C}}, A) = \int q(z|\mathcal{D}_{\mathbb{C}}, A)q(L(0)|\mathcal{D}_{\mathbb{C}})p(X_{l^{\mathbb{T}}}(t^{\mathbb{T}})|t^{\mathbb{T}}, l^{\mathbb{T}}, z, L(0), A)\mathrm{d}z\mathrm{d}L(0). \quad (4)$$

We can use the Monte Carlo method to approximate the integral and moment matching to construct a Gaussian posterior approximation for the distribution as $p(X_{l^{\mathbb{T}}}(t^{\mathbb{T}})|t^{\mathbb{T}}, l^{\mathbb{T}}, \mathcal{D}_{\mathbb{C}}, A) \approx \mathcal{N}(\mathcal{Z}_1, \mathcal{Z}_2)$, where $\mathcal{Z}_1 = \frac{1}{J}\sum_{j=1}^{J}\mu_{X_{l^{\mathbb{T}}}}^{(j)}(t^{\mathbb{T}})$, $\mathcal{Z}_2 = [\frac{1}{J}\sum_{j=1}^{J}(\sigma_{X_{l^{\mathbb{T}}}}^{(j)}(t^{\mathbb{T}}))^2 + (\mu_{X_{l^{\mathbb{T}}}}^{(j)}(t^{\mathbb{T}}))^2] - [\frac{1}{J}\sum_{j=1}^{J}\mu_{X_{l^{\mathbb{T}}}}^{(j)}(t^{\mathbb{T}})]^2$, J is the sampling number for z and $L(0)$, and $\mu_{X_{l^{\mathbb{T}}}}^{(j)}(t^{\mathbb{T}})$ and $\sigma_{X_{l^{\mathbb{T}}}}^{(j)}(t^{\mathbb{T}})$ are outputs of decoder network $\boldsymbol{d}$. The full derivation of the training loss and predictions can be found in Appendixes A.1.2 and A.1.3 of Supporting information.

4 Deep insights into the NDP4ND

4.1 Stochastic skeleton vs. deterministic skeletons

In network dynamics, both the node itself and its neighbors can affect state updates, making direct learning of network dynamics π_ϕ extremely difficult due to the combinatorial explosion problem caused by too many free variables [50]. The current approaches usually separate the original dynamics into two main parts [6, 18, 30, 51]: self dynamics and interaction dynamics, to reduce the number of free variables. In principle, there are two extreme situations as shown in Figure 3. By assuming that all nodes share self dynamics f and interaction dynamics g, one situation is to extremely simplify network dynamics to only two functions, but also reduces the capacity of the model to represent complex networked systems. The other situation is that each node has its self dynamics f_l and each pair of neighbors has their interaction dynamics $g_{l,j}$. The modeling flexibility leads to the need to learn n^2 functions, where n represents the system size. Although it is possible to manually screen for levels of modeling, how to balance complexity and flexibility is still an open question. More importantly, these situations are deterministic. That is, the learned dynamics functions remain unchanged, thus lacking generalization for new system behaviors.

Herein, we introduce a novel concept, i.e., stochastic skeleton, to produce diverse data-adaptive templates, which have the potential to overcome the above issues. It models network dynamics from the non-deterministic perspective as $\frac{\mathrm{d}X_l(t)}{\mathrm{d}t} = f_l(X_l(t)) + \sum_j A_{l,j}g_{l,j}(X_l(t), X_j(t))$, s.t. $f_l \sim p(f|\text{Observations})$,

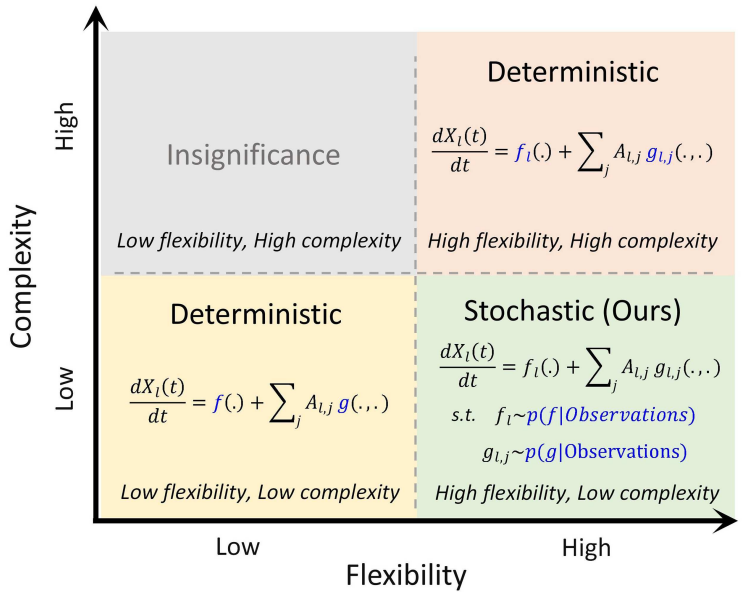

Figure 3 (Color online) Principle behind the NDP4ND is stochastic skeleton, which can model network dynamics from a non-deterministic perspective, achieving the balance between flexibility and complexity.

$g_{l,j} \sim p(g|\text{Observations})$, where $p(f|\text{Observations})$ and $p(g|\text{Observations})$ are data-conditional distributions of self and interaction dynamics respectively. It can ensure the modeling flexibility by sampling diverse dynamics functions and only needs to learn two distributions, thereby reducing the complexity of learning, and achieving an adaptive balance between flexibility and complexity. Moreover, it can generalize new system behaviors, which is brought by data-adaptive modeling.

4.2 NDP4ND as a neural implementation of stochastic skeleton

Actually, the essence of NDP4ND is a neural implementation of stochastic skeletons. We build two conditional distributions of dynamics functions via an encoding-processing-decoding process. $\boldsymbol{S}(\cdot, z)$ and $\boldsymbol{I}(\cdot, \cdot, z)$ in the NDP4ND are to implement $p(f|\text{Observations})$ and $p(g|\text{Observations})$ separately, where $\boldsymbol{S}$ and $\boldsymbol{I}$ are parameterized by neural networks, which brings non-linear modeling capabilities. z is a random vector that follows a distribution conditioned on observation information, i.e., $p(z|\mathcal{D}_{\mathbb{C}}, A)$. Since z serves as the input for $\boldsymbol{S}$ and $\boldsymbol{I}$, uncertainty can be introduced. By sampling different z, $\boldsymbol{S}(\cdot, z)$ and $\boldsymbol{I}(\cdot, \cdot, z)$ can represent diverse functions, even if the parameters in $\boldsymbol{S}$ and $\boldsymbol{I}$ are frozen. This way enables the NDP4ND to achieve data-adaptive modeling for conditional distributions of network dynamics functions.

5 Experiments

To verify the effectiveness of the NDP4ND, we carried out extensive experiments on widely networked systems, including mutualistic interaction dynamics in ecosystems, second-order phototaxis dynamics, brain dynamics, compartment models in epidemiology, and empirical systems[1)].

Construction for training and testing sets. We assign randomly sampled values to the tunable parameters in the ODE skeletons of networked systems to obtain massive network ODE instances for each networked system. Based on different network ODE instances, topologies, and initial states, we randomly generate 300–1500 diverse training tasks and 20–100 testing tasks per system. Note that the sampled ODE instances to produce testing tasks have not appeared in the training set. Therefore, for the model, the testing tasks to be handled come from the emerging dynamics. We irregularly and sparsely sample spatio-temporal observations from each testing task to trigger predictions. Please refer to Appendix B of Supporting information for details of network dynamics and experimental settings.

1) The data and source code are freely available at GitHub (https://github.com/csjtx1021/neural_ode_processes_for_network_dynamics-master) to ensure the reproduction of our experimental results.

Table 1 Interpolation and extrapolation results across all testing network ODEs on mutualistic interaction dynamics in terms of quantitative predictive error and similarity[a)]

Methods	Metrics	Interpolation ($T_o \leqslant 50$)	Extrapolation ($T_o > 50$)
LG-ODE	MAE	4.49E+02 (2.24E+03)	1.33E+03 (6.66E+03)
	DTW	2.24E+04 (1.12E+05)	6.67E+04 (3.33E+05)
NDCN	MAE	5.08E+00 (1.34E+01)	1.98E+02 (1.50E+03)
	DTW	2.34E+02 (6.69E+02)	9.89E+03 (7.50E+04)
DNND	MAE	7.40E−01 (8.96E−01)	1.57E+03 (1.57E+04)
	DTW	2.69E+01 (4.59E+01)	7.87E+04 (7.83E+05)
NP	MAE	5.77E+00 (1.37E+00)	5.79E+00 (1.45E+00)
	DTW	2.86E+02 (6.90E+01)	2.89E+02 (7.24E+01)
NDP	MAE	5.56E+00 (1.42E+00)	5.75E+00 (1.46E+00)
	DTW	2.76E+02 (7.04E+01)	2.87E+02 (7.30E+01)
NDP4ND (ours)	MAE	**3.62E−01 (2.16E−01)**	**3.67E−01 (2.73E−01)**
	DTW	**1.43E+01 (1.01E+01)**	**1.82E+01 (1.37E+01)**
NDP4ND w/o ode	MAE	7.95E−01 (4.23E−01)	7.75E−01 (4.85E−01)
	DTW	3.17E+01 (1.95E+01)	3.87E+01 (2.43E+01)
NDP4ND w/o z	MAE	1.09E+00 (8.86E−01)	1.06E+00 (8.81E−01)
	DTW	4.96E+01 (4.45E+01)	5.31E+01 (4.41E+01)

a) The reported values are the mean (standard deviation) of the mean absolute error (MAE) and dynamic time warping (DTW) between prediction and ground truth. The best results are bolded.

Baselines. We compared against the state-of-the-art neural dynamics methods that can handle irregularly-sampled observations on networks, including latent graph ODE (LG-ODE) [28], neural dynamics on complex network (NDCN) [29], and Dy-Net neural dynamics (DNND) [30], and several NPs methods including NPs [41], NDPs [33], and variants of our model, i.e., NDP4ND w/o ode and NDP4ND w/o z. We note that NPs and NDPs are not specifically designed for network dynamics learning. Using them as comparison methods in the experiments is to verify the necessity of introducing interactions of time dynamics on networks. Since the neural dynamics methods focus on learning a single network dynamics task, they should start from scratch training for each new testing task. On the contrary, our NDP4ND model is actually trained only once per networked system.

5.1 Learning various network dynamics

5.1.1 *Mutualistic interaction dynamics in ecosystems*

Mutualistic interaction dynamics among species in ecology capture the abundance $X_i(t)$ of species i by integrating the mutualistic interactions from neighboring ecosystems, such as the incoming migration and negative growth [51]. We consider five network topologies in this scenario, i.e., grid, random, power-law, small world, and community networks. By assigning randomly sampled values from a certain range to the tunable parameters ϕ in the network ODE of mutualistic interaction dynamics, we can obtain various ODE instances. Then, based on different ODE instances, topologies, and initial states sampled from the distribution $\mathcal{I}$, we randomly produce N_{Tr} network dynamics trajectories as the training set and N_{Te} trajectories as the testing set. Please note that the sampling ODE instances used to generate the testing set have not yet appeared in the training set. That is to say, all testing ODEs can be considered as network emerging dynamics. The irregular-and-sparse-sampled partial observations from each testing task are to trigger predictions. We compared with the state-of-the-art neural dynamics methods on networks for interpolation and extrapolation tasks in terms of prediction error and similarity, i.e., the LG-ODE [28], NDCN [29], and DNND [30], who can deal with the irregular-sampled partial observations and learn continuous network dynamics. The power comes from using ODE to model the continuous change in hidden space [28,29] or state space [30]. The average ratio of observations in the testing network ODEs is 5.81%. We see that the compared neural dynamics methods unsurprisingly underestimate the testing ODEs and are extremely unstable because of the low sampling density, while our NDP4ND reaches the best results for both interpolation and extrapolation tasks, showing its effectiveness in learning on sparse and irregular-sampled observations from new ODEs (Table 1). The LG-ODE has not learned any effective network dynamics from a few observations (Figure 4(a)) and the NDCN only fits well at the observed locations, while the result in the extrapolation region has a significant gap from the ground truth (Figure 4(b)). Although the DNND can learn relatively stable network dynamics, it still

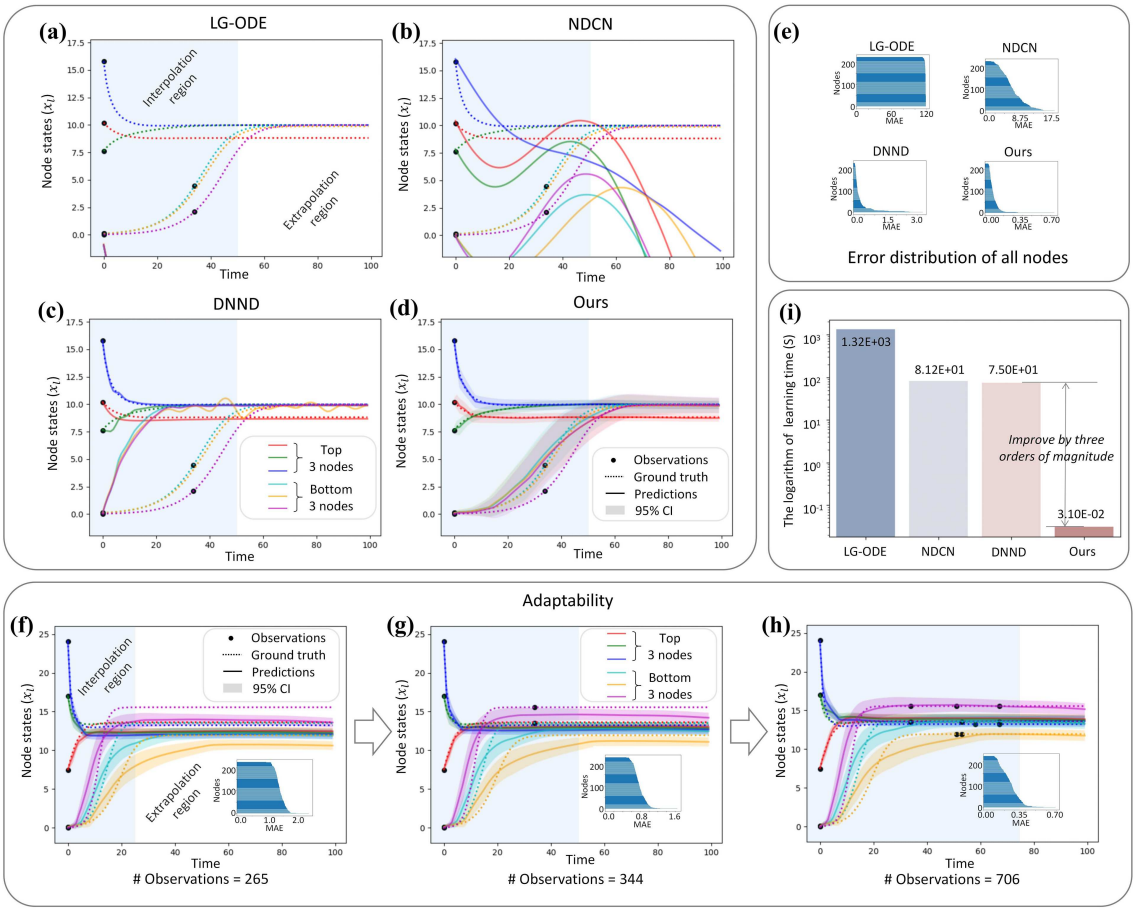

Figure 4 (Color online) Interpolation and extrapolation results on mutualistic interaction dynamics. The testing results of the best-performing and worst-performing three nodes of (a) LG-ODE, (b) NDCN, (c) DNND, and (d) our NDP4ND. The ratio of observations in this testing network ODE is 3.06%. The number of nodes and the maximum value of all observed times (T_o) are 225 and 50, respectively. Our NDP4ND has the best performance in both interpolation and extrapolation, and ultimately stabilizes at two population values. (e) The distributions of MAE between predictions and ground truth for all nodes, demonstrating the high-precision predictions from our method. (f)–(h) Our NDP4ND can directly utilize newly arrived data and does not require any retraining, with adaptively rectifying predictions toward ground truth as observations increase. Note that the subfigures only show the observation data on six nodes instead of all nodes. (i) The average learning time for all testing network ODEs, showing that our NDP4ND improves the learning speed for new dynamics by three orders of magnitude.

underestimates the task (Figure 4(c)). Our NDP4ND has high-precision predictions for interpolation and extrapolation, and ultimately stabilizes at two population values (Figures 4(d) and (e)). Since the LG-ODE, NDCN, and DNND are aimed at learning a single network dynamics instance, it is necessary to retrain for each testing task on sampled observations. Especially, when new observation data from testing ODEs arrives, they need to be retrained to use newly arrived observations. On the contrary, ours does not require any retraining and can adaptively rectify predictions toward ground truth (Figures 4(f)–(h)), benefiting from modeling stochastic processes. By analyzing the computational complexity of our NDP4ND, the cost of dealing with unseen dynamics increases linearly with the system scale (please see Appendix A.3). From the reported average time for processing network ODEs (Figure 4(i)), we see that our NDP4ND significantly improves learning speed by three orders of magnitude compared to the latest method, thanking data adaptability.

5.1.2 *Second-order phototaxis dynamics*

Phototaxis dynamics is a second-order system simulating the dynamics of phototactic bacteria towards a fixed light source [52], which can be applied to the movement of bacteria towards food sources [53] and emergency evacuation modeling [54]. The system performs bacteria's movements through the excitation level of bacteria to the light and their interactions with each other [53]. The complete graph is used to describe all interactions between any two bacteria. The average ratio of observations in the testing

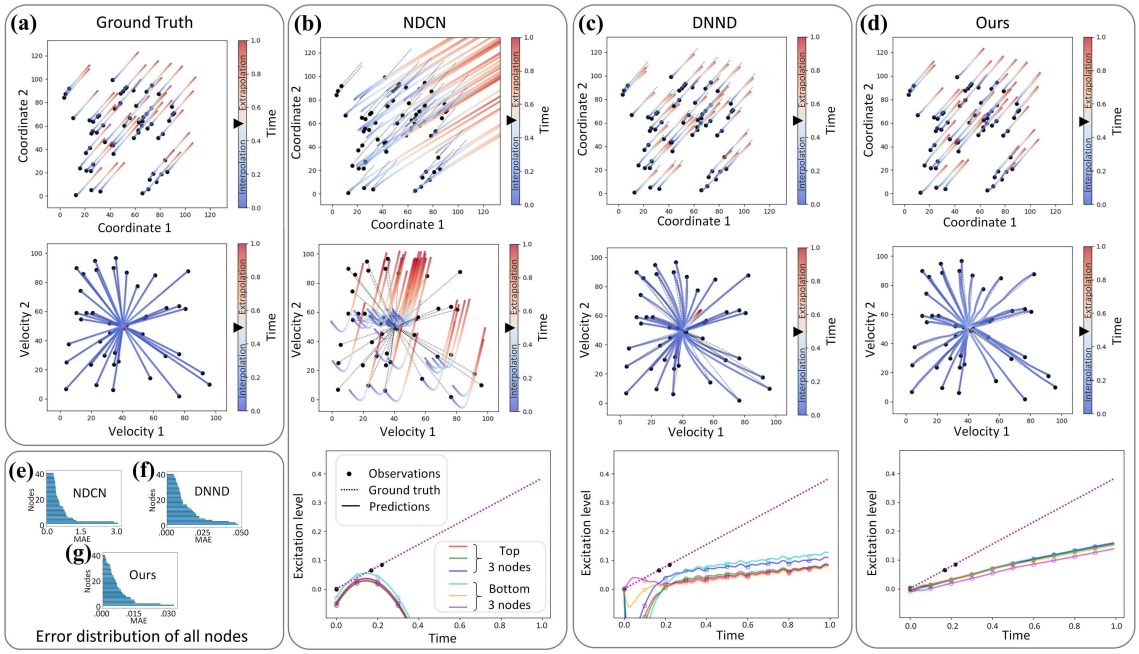

Figure 5 (Color online) Interpolation and extrapolation results on phototaxis dynamics. (a) The ground truth and testing results of (b) NDCN, (c) DNND, and (d) our NDP4ND. The ratio of observations in this testing network ODE is 3.23%. The number of nodes and the maximum value of all observed times (T_o) are 40 and 0.5, respectively. The predictive five states, including coordinate 1, coordinate 2, velocity 1, velocity 2 and excitation level, are reported. The NDCN has not learned effective network dynamics from a few observations. The DNND fits well at the observed locations, but its velocity eventually converges to the wrong solution. Our NDP4ND ultimately stabilizes at the correct velocity and the learned dynamics are relatively stable and close to the ground truth. (e)–(g) The distributions of MAE between predictions and ground truth for all nodes, demonstrating the high-precision predictions from our method.

network ODEs is 4.65%. We do not compare the LG-ODE due to its poor performance. We see that our NDP4ND has the optimal results for both interpolation and extrapolation tasks, which means it can learn effective network dynamics (see Table C1 in Appendix C.1 of Supporting information). The predictive five states on an instance, including coordinate 1, coordinate 2, velocity 1, velocity 2 and excitation level, are reported in Figure 5. The NDCN cannot learn effective network dynamics from a few observations (Figure 5(b)). The DNND fits well at the observed locations, but its velocity eventually converges to the wrong solution (Figure 5(c)). Our NDP4ND ultimately stabilizes at the correct velocity, can learn relatively stable and accurate network dynamics (Figure 5(d)), and has the lowest predictive errors (Figures 5(e)–(g)).

5.1.3 *Brain dynamics*

We also apply our method to brain activities governed by the FitzHugh-Nagumo dynamics [55], which has a quasi-periodic characteristic. Topological structures with power-law distribution are used to simulate the interactions among brain functional areas [18]. The average ratio of observations in the testing network ODEs is 5.07%. We see that our NDP4ND has the lowest predictive error and highest similarity (see Table C2 in Appendix C.1 of Supporting information). From a specific testing ODE instance, we see that the NDCN fits well at the observed locations, but cannot learn the inherent periodic behavior from a few observations (Figure 6(a)) and the DNND still underestimates the task (Figure 6(b)). As a comparison, the predictive results produced by our NDP4ND exhibit periodic behavior, being support of learning effective network dynamics (Figure 6(c)). The lowest MAE error also confirms the effectiveness of our method's prediction (Figures 6(d)–(f)).

5.1.4 *Compartment models in epidemiology*

Compartment models [56, 57] are often used to simulate the spread of epidemics on networks. Herein, we conduct experiments on three commonly used compartment models: susceptible-infectious-susceptible (SIS), susceptible-infectious-recovered (SIR), and susceptible-exposed-infectious-susceptible (SEIS). The observed ratio of testing tasks is around 6.0%. Although our method is slightly inferior to the NDCN

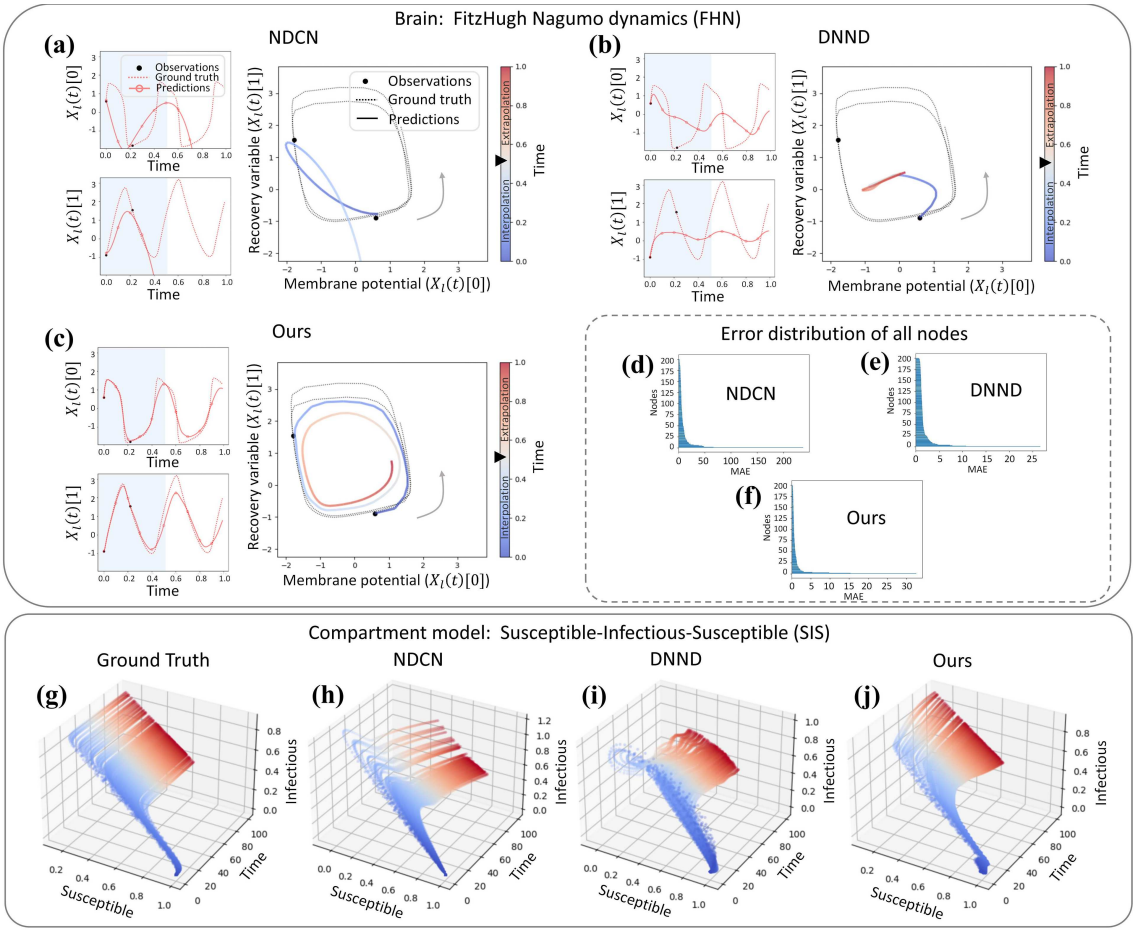

Figure 6 (Color online) Interpolation and extrapolation results on brain dynamics and SIS. The testing results of (a) NDCN, (b) DNND, and (c) our NDP4ND on brain dynamics. The ratio of observations in this testing network ODE is 3.23%. The number of nodes and the maximum value of all observed times (T_o) are 200 and 0.5, respectively. Two-dimensional states, i.e., membrane potential ($X_l(t)[0]$) and recovery variable ($X_l(t)[1]$), are reported. Our NDP4ND can learn effective dynamics that reveal the expected periodicity in both interpolation and extrapolation regions. (d)–(f) The distributions of MAE between predictions and ground truth for all nodes on brain dynamics, demonstrating the high-precision predictions from our method. (g) The ground truth and testing results of (h) NDCN, (i) DNND, and (j) our NDP4ND on SIS dynamics. The number of nodes and the maximum value of all observed times (T_o) are 200 and 50, respectively. The time series predicted by our method are closest to the ground truth and meet the constraint of Susceptible+Infectious=1.

in the interpolation region in some scenarios, the results in the extrapolation region are far superior to the compared methods (see Table C3 in Appendix C.1 of Supporting information), showing its power to extrapolate and cover various epidemic scenarios. An instance of the SIS dynamics is shown in Figure 6(g). Both the NDCN and DNND cannot produce the effective predictions (Figure 6(h)). Especially, the results generated by the DNND do not meet the constraint of Susceptible + Infectious = 1 (Figure 6(i)). As a comparison, the shape of the time series predicted by our method is closest to the ground truth and the results are on the plane of Susceptible + Infectious = 1 (Figure 6(j)). More instances can be found in Figure C1 in Appendix C.2 of Supporting information.

5.1.5 *Ablation studies*

By comparing existing NPs models [33, 41], the results show that our proposed method and its variants outperform the NPs models (see Table 1), demonstrating that modeling the interactions of temporal dynamics is necessary. Note that the NPs [41] and NDPs [33] are not specifically designed for network dynamics learning. Using them as comparison methods is to verify the necessity of introducing interactions of time dynamics on networks. Due to their limited flexibility, they cannot handle networked systems with arbitrary sizes. We give them the power to process network dynamics by utilizing the graph neural network to encode the observations flexibly.

We also compare with two variants, i.e., NDP4ND w/o ode and NDP4ND w/o z. The former removes

the ODE flow in our NDP4ND, i.e., deleting $L_{l_i^{\mathbb{T}}}(t_i^{\mathbb{T}})$ from the input of neural network $\boldsymbol{d}$. This is to explore the impact of the ODE process of the NDP4ND on the results. The latter removes $\tilde{z}$ from the input of neural network $\boldsymbol{d}$ in the architecture. This is to explore the impact of skip connection (i.e., add an information flow path that skips the ODE process) on the results. From Table 1, the NDP4ND achieves better results than its variants, which confirms the effectiveness of our computational architecture design and explicitly modeling the ODE flow describing network dynamics.

5.2 Learning network dynamics from sparse and noisy observations

We also test our method's tolerance for the observed ratio. By predicting system behaviors for 20 new sampled network ODE instances per observed ratio on mutualistic interaction dynamics, as the observation density increases, the predictive error of the NDP4ND appears a decreasing trend, where the prediction error occurs a cliff like decline when the observed ratio is around 6%, reaching around 0.3 (see Figure C2(a) in Appendix C.2 of Supporting information). By contrast, the prediction errors of the LG-ODE, NDCN, and DNND at an observed ratio of 60% are 3.99±3.59, 0.70±0.20, and 0.52±0.24, respectively. This indicates that our NDP4ND's pretraining on a large amount of relevant data is helpful in significantly reducing the required observations when learning new network dynamics. To test the robustness of our NDP4ND for various noise levels, we add white Gaussian disturbances to observations. Our method is tested on sampled 20 new network ODE instances per level. Results show that noise indeed weakens predictions. Benefiting from uncertainty modeling, our NDP4ND is relatively stable and has significantly outperformed the NDCN and DNND (see Figure C2(b) in Appendix C.2 of Supporting information).

5.3 Learning from empirical systems

As impressive network dynamics, emerging and re-EIDs are significant burdens on global economies and public health [8, 58]. We collect daily global spreading data on three real diseases, including H1N1 [59], SARS [59], and COVID-19 [60], and the worldwide airline network retrieved from OpenFights as a directed and weighted empirical topology to build the empirical systems of real-world global epidemic transmission. Only early data before government intervention, i.e., the first 50 days, is considered here to keep the spread features of the disease itself. We allow H1N1 data as training set according to various time intervals and test on all three diseases. As a result, testing on H1N1 can be seen as predicting re-EIDs; after our model is trained on H1N1, testing on SARS and COVID-19 can be seen as predicting EIDs. For short-term forecasting, models use partial observations from the previous 5 days to predict the case number on the next day (Figures 7(a)–(f)). Long-term forecasting requires those from the previous 10 days to predict the next 10 days (Figures 7(g)–(i)). We see that our NDP4ND achieves smoother and better results than the NDCN and DNND in both short-term and long-term predictions. This demonstrates its expressive power for empirical data and its potential for processing real-world emerging and re-emerging tasks.

6 Discussion

We propose the NDP4ND, a new class of stochastic processes governed by stochastic data-adaptive network dynamics, to represent the trajectory function space generated by network dynamics to provide a reasonable solution for learning network emerging dynamics from sparse, irregularly-sampled, partial, and noisy data. Intensive experiments demonstrate the effectiveness of our method in adapting to many simulated and real-world tasks. However, the accuracy of the current method is still limited by the approximate numerical integration in the ODE flow. Therefore, how to alleviate error accumulation is an outstanding problem, especially for the neural-ODE-based methods [32]. It may be a promising attempt to map raw dynamics to linear ODE through an invertible neural network [61], and then speed up the integration process or directly calculate the closed-form solution of latent linear ODE [62].

Real-world complex networked systems are ubiquitous, ranging from celestial movement to epidemic spreading. Such a large number of scenarios have their own unique characteristics, such as the periodicity of brain activity data [55] and the equilibrium of thermal dynamics [63]. Therefore, how to beyond our method and to integratively and systematically represent the trajectory function space produced by abundant network dynamics scenarios and build a pretrained network dynamics model with richer

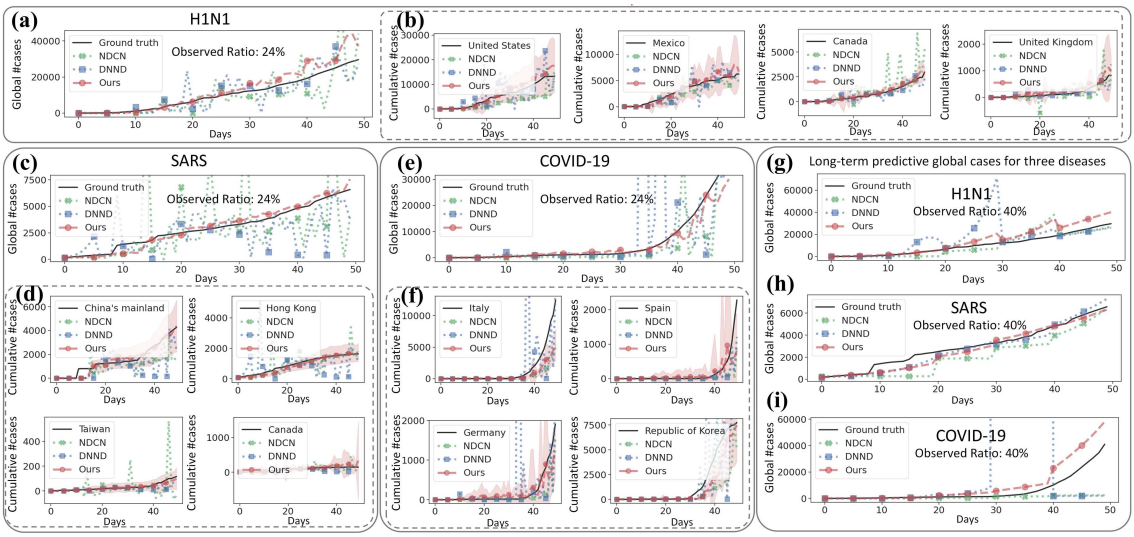

Figure 7 (Color online) Testing results on real-world global epidemic transmission. Note that, testing on H1N1 can be seen as predicting re-EIDs, but testing on SARS and COVID-19 as predicting EIDs. (a)–(f) The short-term forecasting results of the methods for three different diseases. From the global case number ((a), (c), (e)), our NDP4ND has smoother and better results than others. The predictions on the case number in major countries or regions ((b), (d), (f)) also support this performance. (g)–(i) The long-term forecasting results of the methods for three different diseases. Our NDP4ND achieves relatively better results, especially for EID COVID-19. Other methods either fail to predict obvious outbreak trends (NDCN) or predict terrifying drastic erroneous fluctuations (DNND at 30–40 days) from sparse observations, while our method is closer to real epidemic outbreak data and has similar trends.

expressive power for more diverse downstream tasks still remains an open question, such as heterogeneous [64], high-order [65], and controllable network dynamics [66].

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2021ZD0112500), National Natural Science Foundation of China (Grant Nos. U22A2098, 62172185, 62206105, 62202200), and Jilin Province Youth Science and Technology Talent Support Project (Grant No. QT202225).

Supporting information Appendixes A–C. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zhou T J, Zhang W X, Chen D L, et al. Understanding and building upon pioneering work of Nobel Prize in Physics 2021 laureates Syukuro Manabe and Klaus Hasselmann: from greenhouse effect to Earth system science and beyond. Sci China Earth Sci, 2022, 65: 589–600

2 McCright A M, Dunlap R E, Xiao C. The impacts of temperature anomalies and political orientation on perceived winter warming. Nat Clim Change, 2014, 4: 1077–1081

3 Chen B, Wang X, Zhang W, et al. Public opinion dynamics in cyberspace on Russia-Ukraine War: a case analysis with Chinese Weibo. IEEE Trans Comput Soc Syst, 2022, 9: 948–958

4 Liu Y, Gu Z, Xia S, et al. What are the underlying transmission patterns of COVID-19 outbreak? An age-specific social contact characterization. EClinicalMedicine, 2020, 22: 100354

5 Newman M, Barabási A-L, Watts D J. The Structure and Dynamics of Networks. Princeton: Princeton University Press, 2006

6 Barzel B, Barabási A L. Universality in network dynamics. Nat Phys, 2013, 9: 673–681

7 Gosak M, Markovič R, Dolenšek J, et al. Network science of biological systems at different scales: a review. Phys Life Rev, 2018, 24: 118–135

8 Jones K E, Patel N G, Levy M A, et al. Global trends in emerging infectious diseases. Nature, 2008, 451: 990–993

9 Huigang L, Cui H, Xiaoli Z, et al. Significance of and outlook for the Biosecurity Law of the People's Republic of China. J BioSafe Biosecur, 2021, 3: 46–50

10 Prince A M, Kimball L F. Emerging Infections: Microbial Threats to Health in the United States. Washington: National Academy Press, 1992

11 Lu R, Zhao X, Li J, et al. Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. Lancet, 2020, 395: 565–574

12 Aryandoust A, Patt A, Pfenninger S. Enhanced spatio-temporal electric load forecasts using less data with active deep learning. Nat Mach Intell, 2022, 4: 977–991

13 Pei H, Yang B, Liu J, et al. Group sparse Bayesian learning for active surveillance on epidemic dynamics. In: Proceedings of AAAI Conference on Artificial Intelligence, 2018. 800–807

14 Tang X, Yao H, Sun Y, et al. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. AAAI, 2020, 34: 5956–5963

15 Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with AlphaFold. Nature, 2021, 596: 583–589

16 Zhavoronkov A, Ivanenkov Y A, Aliper A, et al. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nat Biotechnol, 2019, 37: 1038–1040

17 Davies A, Veličković P, Buesing L, et al. Advancing mathematics by guiding human intuition with AI. Nature, 2021, 600: 70–74

18 Gao T T, Yan G. Autonomous inference of complex network dynamics from incomplete and noisy data. Nat Comput Sci, 2022, 2: 160–168

19 Mangan N M, Kutz J N, Brunton S L, et al. Model selection for dynamical systems via sparse regression and information criteria. Proc R Soc A, 2017, 473: 20170009

20 Zhang Z, Zhao Y, Liu J, et al. A general deep learning framework for network reconstruction and dynamics learning. Appl Netw Sci, 2019, 4: 110

21 Murphy C, Laurence E, Allard A. Deep learning of contagion dynamics on complex networks. Nat Commun, 2021, 12: 4720

22 Fritz C, Dorigatti E, Rügamer D. Combining graph neural networks and spatio-temporal disease models to improve the prediction of weekly COVID-19 cases in Germany. Sci Rep, 2022, 12: 3930

23 Liang B, Wang L, Wang X. Autoregressive GNN-ODE GRU model for network dynamics. 2022. ArXiv:2211.10594

24 Huang Z, Sun Y, Wang W. Coupled graph ode for learning interacting system dynamics. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2021. 705–715

25 Wang Z, Xin J, Chen Q, et al. NDCN-brain: an extensible dynamic functional brain network model. Diagnostics, 2022, 12: 1298

26 Wen S, Wang H, Metaxas D. Social ODE: multi-agent trajectory forecasting with neural ordinary differential equations. In: Proceedings of European Conference on Computer Vision, 2022. 217–233

27 Huang Z, Sun Y, Wang W. Generalizing graph ODE for learning complex system dynamics across environments. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2023. 798–809

28 Huang Z, Sun Y, Wang W. Learning continuous system dynamics from irregularly-sampled partial observations. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 16177–16187

29 Zang C, Wang F. Neural dynamics on complex networks. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020. 892–902

30 Liu B, Luo W, Li G, et al. Do we need an encoder-decoder to model dynamical systems on networks? In: Proceedings of International Joint Conference on Artificial Intelligence, 2023. 2178–2186

31 Stickler B A, Schachinger E. Ordinary differential equations: initial value problems. In: Basic Concepts in Computational Physics. Berlin: Springer, 2016

32 Chen R T Q, Rubanova Y, Bettencourt J, et al. Neural ordinary differential equations. In: Proceedings of Advances in Neural Information Processing Systems, 2018

33 Norcliffe A, Bodnar C, Day B, et al. Neural ODE processes. In: Proceedings of International Conference on Learning Representations, 2021

34 Norcliffe A, Bodnar C, Day B, et al. On second order behaviour in augmented neural ODEs. In: Proceedings of Advances in Neural Information Processing Systems, 2020

35 Floryan D, Graham M D. Data-driven discovery of intrinsic dynamics. Nat Mach Intell, 2022, 4: 1113–1120

36 Brockmann D, Helbing D. The hidden geometry of complex, network-driven contagion phenomena. Science, 2013, 342: 1337–1342

37 Poli M, Massaroli S, Park J, et al. Graph neural ordinary differential equations. 2019. ArXiv:1911.07532

38 Sanchez-Gonzalez A, Godwin J, Pfaff T, et al. Learning to simulate complex physics with graph networks. In: Proceedings of International Conference on Machine Learning, 2020. 8459–8468

39 Pfaff T, Fortunato M, Sanchez-Gonzalez A, et al. Learning mesh-based simulation with graph networks. In: Proceedings of International Conference on Learning Representations, 2021

40 Bishnoi S, Bhattoo R, Jayadeva J, et al. Enhancing the inductive biases of graph neural ODE for modeling physical systemssenhancing the inductive biases of graph neural ODE for modeling physical systems. In: Proceedings of International Conference on Learning Representations, 2023

41 Garnelo M, Chwarz J, Rosenbaum D, et al. Neural processes. In: Proceedings of ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models, 2018

42 Kim H, Mnih A, Schwarz J et al. Attentive neural processes. In: Proceedings of International Conference on Learning Representations, 2019

43 Foong A, Bruinsma W, Gordon J, et al. Meta-learning stationary stochastic process prediction with convolutional neural

processes. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 8284–8295
44 Nassar M, Wang X, Tumer E. Conditional graph neural processes: a functional autoencoder approach. In: Proceedings of NIPS Workshop on Bayesian Deep Learning, 2018
45 Cangea C, Day B, Jamasb A R, et al. Message passing neural processes. In: Proceedings of ICLR Workshop on Geometrical and Topological Representation Learning, 2022
46 Carr A N, Wingate D. Graph neural processes: towards Bayesian graph neural networks. 2019. ArXiv:1902.10042
47 Liang H, Gao J. How neural processes improve graph link prediction. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, 2022. 3543–3547
48 Øksendal B. Stochastic Differential Equations. Berlin: Springer, 2003
49 Garnelo M, Rosenbaum D, Maddison C, et al. Conditional neural processes. In: Proceedings of International Conference on Machine Learning, 2018. 1704–1713
50 Udrescu S M, Tegmark M. AI Feynman: a physics-inspired method for symbolic regression. Sci Adv, 2020, 6: eaay2631
51 Gao J, Barzel B, Barabási A L. Universal resilience patterns in complex networks. Nature, 2016, 530: 307–312
52 Ha S Y, Levy D. Particle, kinetic and fluid models for phototaxis. Discrete Cont Dyn Syst-B, 2009, 12: 77–108
53 Lu F, Zhong M, Tang S, et al. Nonparametric inference of interaction laws in systems of agents from trajectory data. Proc Natl Acad Sci USA, 2019, 116: 14424–14433
54 Lin J, Lucas T A. A particle swarm optimization model of emergency airplane evacuations with emotion. Netws Heterog Media, 2015, 10: 631–646
55 Gerster M, Berner R, Sawicki J, et al. FitzHugh-Nagumo oscillators on complex networks mimic epileptic-seizure-related synchronization phenomena. Chaos-An Interd J Nonlinear Sci, 2020, 30: 123130
56 Dimitrov N B, Meyers L A. Mathematical approaches to infectious disease prediction and control. INFORMS, 2010, 1: 1–25
57 Youssef M, Scoglio C. An individual-based approach to SIR epidemics in contact networks. J Theor Biol, 2011, 283: 136–144
58 Msemburi W, Karlinsky A, Knutson V, et al. The WHO estimates of excess mortality associated with the COVID-19 pandemic. Nature, 2023, 613: 130–137
59 Nunes L. A brief comparative study of epidemics. 2020. https://www.kaggle.com/code/lnunes/a-brief-comparative-study-of-epidemics
60 Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. Lancet Infect Dis, 2020, 20: 533–534
61 Zhi W, Lai T, Ott L, et al. Learning efficient and robust ordinary differential equations via invertible neural networks. In: Proceedings of International Conference on Machine Learning, 2022. 27060–27074
62 Hasani R, Lechner M, Amini A, et al. Closed-form continuous-time neural networks. Nat Mach Intell, 2022, 4: 992–1003
63 Luikov A V. Analytical Heat Difusion Theory. Amsterdam: Elsevier, 2012
64 D'Souza R M, di Bernardo M, Liu Y Y. Controlling complex networks with complex nodes. Nat Rev Phys, 2023, 5: 250–262
65 Bianconi G. Higher-Order Networks Elements in the Structure and Dynamics of Complex Networks. Cambridge: Cambridge University Press, 2021
66 Baggio G, Bassett D S, Pasqualetti F. Data-driven control of complex networks. Nat Commun, 2021, 12: 1429