

Machine learning prediction of network dynamics with privacy protection

Xin Xia

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

Yansen Su

*The Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,
School of Artificial Intelligence, Anhui University, Hefei 230601, China*

Linyuan Lü

*Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610054, China*Xingyi Zhang *The Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,
School of Artificial Intelligence, Anhui University, Hefei 230601, China*Ying-Cheng Lai *School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA*Hai-Feng Zhang **The Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education,
School of Mathematical Science, Anhui University, Hefei 230601, China*

(Received 15 June 2022; accepted 12 October 2022; published 2 November 2022)

Predicting network dynamics based on data, a problem with broad applications, has been studied extensively in the past, but most existing approaches assume that the complete set of historical data from the whole network is available. This requirement presents a great challenge in applications, especially for large, distributed networks in the real world, where data collection is accomplished by many clients in a parallel fashion. Often, each client only has the time series data from a partial set of nodes, and the client has access to only partial time stamps of the whole set of time series data and the partial structure of the network. Due to privacy concerns or license-related issues, the data collected by different clients cannot be shared. Accurately predicting the network dynamics while protecting the privacy of different parties is a critical problem in modern times. Here, we propose a solution based on federated graph neural networks (FGNNs) that enables the training of a global dynamic model for all parties without data sharing. We validate the working of our FGNN framework through two types of simulations to predict a variety of network dynamics (four discrete and three continuous dynamics). As a significant real-world application, we demonstrate successful prediction of state-wise influenza spreading in the USA. Our FGNN scheme represents a general framework to predict diverse network dynamics through collaborative fusing of the data from different parties without disclosing their privacy.

DOI: [10.1103/PhysRevResearch.4.043076](https://doi.org/10.1103/PhysRevResearch.4.043076)**I. INTRODUCTION**

This paper deals with the problem of predicting complex network dynamics from distributed data without compromising the privacy of the data sources. In particular, given a large

network whose dynamics are unknown and given that only *local* historical data or time series are available through measurements conducted by different parties (clients or agents), the objective is to accurately predict the dynamical evolution of the network for a number of time steps under the constraint of no data sharing of any kind among the clients. To paraphrase, among the clients who performed the measurements, there can be no communication of any sort because of privacy considerations. This problem of predicting network dynamics without privacy disclosure is significantly more challenging than previously studied inverse problems in the field of reverse engineering of networked dynamical systems. The main contribution of this paper is the articulation and

*haifengzhang1978@gmail.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

validation of an effective machine-learning-based solution to this problem.

Inferring or reconstructing the dynamical process on a network based on time series data has been an active field in the past two decades [1,2]. A diverse array of methodologies have been proposed, including those based on the collective dynamics [3–7], stochastic analysis [8,9], compressive sensing [10–15], and machine learning [16–19]. In nonlinear dynamics, the research on data-based identification and forecasting of system dynamics has an even longer history [20,21]. For example, an earlier approach focused on approximating a nonlinear system by various linear equations in different regions of the phase space so that the local Jacobian matrices can be constructed [20,22,23] or the ordinary differential equations can be found to fit the data [24]. Methods based on chaotic synchronization [25] for estimating the system parameters were also investigated. Of particular importance is the approach to finding the system equations (hence the system dynamics) from data. This “natural” approach dates back to the original work of Crutchfield and McNamara [26], who exploited the concept of qualitative information to deduce the effective equations of motion of the system responsible for the deterministic portion of the observed random behavior. An inverse Frobenius-Perron approach to generate a dynamical system close to the original system in the sense of the invariant density was proposed [27]. The Kronecker product representation was also used for modeling and nonlinear parameter estimation [28]. In the past decade, sparse optimization methods, e.g., compressive sensing [29,30], were introduced for finding the system equations from data [10–12,15,31] for nonlinear and complex dynamical systems whose velocity fields or mapping functions are describable by a number of fairly elementary mathematical functions. This equation-finding approach, while appealing and satisfying from a mathematical point of view, may not have significant practical value in real-world applications, as the dynamical processes there often cannot be described by a collection of simple functions. Even in cases where an approximate set of equations can be found, sensitivity to small errors typically seen in nonlinear and complex dynamical systems can lead to large deviations between the dynamics as predicted by the equations and the ground truth. In these situations, machine learning has gained recent attention as a viable approach to predicting dynamics from data [17,18,32–41].

To our knowledge, the increasingly critical issue of privacy has not been addressed in the literature on data-based prediction of network dynamics. In fact, a tacit assumption employed in the current literature on data-based prediction of network dynamics is that the observed data are transparent and available to all the observers, which include the network structure and the time series data, as schematically illustrated in Fig. 1(a). If all the data are collected by a single client, privacy is not an issue. However, in applications, practical limits such as the cost of observations and the timeliness render it necessary to employ different clients to collect the data [42]. For example, disease-related data at different times and/or in different regions are often collected by multiple parties, which cannot be shared due to the requirements of data security and privacy protection, leading to the emergence of so-called “data islands” [43,44]. In the era of big

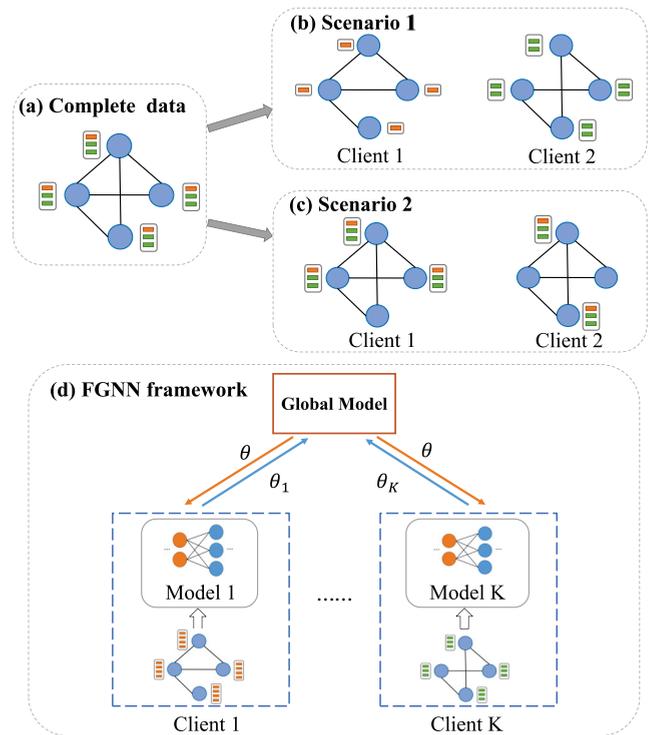


FIG. 1. Scenarios of multiparty data collection and the proposed FGNN framework for predicting the network dynamics without compromising privacy. (a) Data collection without privacy protection. Clients 1 and 2 are two local clients. The rectangular boxes represent the time series data of the nodes with the color of the box indicating the data at different time stamps. In this case, each client has complete time series data of the nodes and complete information about the network structure; so the information possessed by either of the clients is transparent to the other client, i.e., there is no privacy. (b) Each client collects only partial time stamps of the time series data with incomplete knowledge about the network structure (scenario 1). (c) Each client collects only the time series data from partial nodes but with complete knowledge about the network structure (scenario 2). (d) Proposed FGNN framework for predicting network dynamics without compromising privacy.

data analytics and machine learning, data have significant commercial, security, and applied value. While the pertinent entities are able to better accomplish their goals with more data, privacy protection puts a limit on how many data any individual client is able to acquire. For network dynamics, more time series data and more information about the network structure are certainly beneficial to achieving higher accuracies in predicting the dynamical evolution, but the data need to be collected by individual clients for whom privacy may be of great importance. A key question with practical significance is how to coordinate the network structure and time series data distributed in different organizations to improve the accuracy of dynamics prediction, without compromising privacy.

Existing cooperative data-learning methods include secure multiparty computing [45], multitask learning [46], and federated learning [47]. As a distributed machine learning technology with privacy protection where data fusion can be achieved among the parties without leaking their private data

[48], federated learning has gained much interest in applications such as traffic flow prediction [49] and recommendation systems [50]. Federated learning has also been exploited for graph representation learning [51,52] and its downstream tasks such as node classification, link prediction, and graph classification [53–55]. The basic principle underlying federated learning is that each client trains a local model with local data, and a joint global model is generated by aggregating the parameters of the local models [56]. Whether the federated learning framework can predict the network dynamics and how to use it to predict the network dynamics based on multiparty data have not been considered. To this end, we set out to develop a framework based on federated graph neural networks (FGNNs) to predict the network dynamics from distributed time series data through jointly learning an optimal global dynamics model without exposing the data of each party, i.e., without compromising privacy.

The problem of predicting the network dynamics without compromising privacy is significantly more challenging than that of inferring the network structure, for the following three reasons. First, different clients record only partial time series data (partial time stamps or partial nodes) of the network dynamics. Second, it is necessary to learn the rules of the dynamical evolution from the data. Third, the network data held by different clients are in general not independent and identically distributed (iid). To make the forecasting problem with full privacy protection addressable at the present, in this paper we focus on two data-collection scenarios. In scenario 1, each client collects only the partial time series data of all nodes and the partial structure of networks, as illustrated in Fig. 1(b). In scenario 2, each client has only the time series data from a subset of nodes, but each client has complete knowledge about the structure of the network, as shown in Fig. 1(c). We demonstrate that the prediction performance of the FGNN framework far exceeds that of the local models obtained by the individual local clients through local data. We also show that the FGNN framework is capable of predicting the evolution of a diverse array of discrete and continuous network dynamical processes. Overall, the FGNN framework represents a powerful machine-learning-based approach for predicting the global network dynamics from only local data with guaranteed privacy protection.

II. THE PROPOSED FGNN FRAMEWORK

In this section, the architectures of the FGNN framework and its details are described.

A. Basic principles of the proposed FGNN framework

Let $G = \{V, E\}$ denote a given network, where $V = \{v_i | i = 1, \dots, n\}$ and $E = \{e_{ij} | i, j = 1, \dots, n\}$ are the sets of nodes and edges, respectively. The available time series are organized into a data matrix \mathcal{X} , where each row corresponds to the time series of one node and each column is associated with one time stamp of all nodes. The data matrix \mathcal{X} is generated by an unknown dynamical process \mathcal{D} on the underlying network (in this paper, four discrete and three continuous dynamics are considered, and a detailed description of the various dynamical processes is presented in Appendix B).

Let X^t be a vector that stores the states of all nodes at time t (i.e., a column of matrix \mathcal{X}), and the nodal state evolution over time t is given by

$$X^{t+1} = \mathcal{D}(X^t, G). \tag{1}$$

Suppose there are K clients, a client k views the network as its own local network structure G_k , and the available time series data are denoted as \mathcal{X}_k (i.e., a submatrix of \mathcal{X}). The time series data of client k are expressed as $\mathcal{X}_k = \{X_k^1, \dots, X_k^{T_k}\}$, where T_k is the length of the time series recorded by client k . In scenario 1, each client has partial time stamps of the whole time series data \mathcal{X} , and $X_k^t \in R^{n \times 1}$ denotes the states of all nodes at time t by client k . In scenario 2, the time series data held by each client are at the same time stamp: $T_k = T, k = 1, 2, \dots, K$. Let V_k be the nodal set whose time series data can be observed by client k , where $X_k^t \in R^{|V_k| \times 1}$. The state of the node i at time t is expressed as $x_k^t(i) = (X_k^t)_i$.

Each client k can train a local machine learning model M_k to generate the network dynamics based on its own data, with a training parameter set denoted as θ_k . Because of the incomplete observation, the learning capability of any local model in capturing the network dynamics is limited. To overcome this limitation, we build up an FGNN framework by combining the information of $\{G_1, \dots, G_K\}$ and $\{\mathcal{X}_1, \dots, \mathcal{X}_K\}$ to train a global model M enabling us to better learn the dynamical process and to accurately predict the future nodal states of nodes. In particular, we have

$$M(X^t, G, \theta) \approx \mathcal{D}(X^t, G), \tag{2}$$

where θ denotes the parameter set of the global model.

The overall framework of FGNN is illustrated in Fig. 1(d), and it involves four main steps.

Step 1. The central unit initializes the model parameter set θ^0 and distributes it to each client.

Step 2. At the t th iteration, each client k uses the new parameter set θ^t to update its local model M_k . Each client k takes the current states X_k^t as the input at time t and outputs at the next time step the nodal states $\widehat{Y}_k^t = M_k(X_k^t, G_k, \theta_k^t)$ based on its own local network structure G_k . The real states used for training are $Y_k^t = X_{t+1}^k$. To train the model, the loss function of the local model is constructed by the real states and the output as

$$\mathcal{L}(\theta_k^t) = \frac{1}{|V_k|} \sum_{i \in V_k} L(y_k^t(i), \widehat{y}_k^t(i)), \tag{3}$$

where $L(y_k^t(i), \widehat{y}_k^t(i))$ is the loss function of node i between the real state and the predicted output. Once the loss function is defined, the parameter set θ_k^{t+1} of the local models can be updated by a standard back-propagation neural network model (see Sec. II B for further details).

Step 3. The updated parameters of all local models are sent to the central unit, and the parameters of the global model θ are aggregated by the weights $\{\theta_1, \dots, \theta_K\}$ expressed as

$$\theta^{t+1} = \sum_{k=1}^K w_k \theta_k^{t+1}, \tag{4}$$

TABLE I. Neural network architectures for discrete and continuous dynamics. ReLU, rectified linear unit.

	Discrete dynamics	Continuous dynamics
Input layer	One-hot (1, S) Linear (S , 32) ReLU	Linear (1,32) ReLU
Hidden layer	GCN (32,32) or GATN (32,32) ReLU	GCN (32,32) or GATN (32,32) ReLU
Output layer	Linear (32, S) Softmax	Linear (32,1) ReLU
Loss function	Cross-entropy loss	Mean-square loss

where w_k is the aggregation weight measuring the quality of the data of client k ; details are given in Sec. II C. The new updated parameter set θ^{t+1} is sent back to each client.

Step 4. Steps 2 and 3 are repeated until the loss function converges or a given number of training times is reached, and the global FGNN model is the trained joint model.

B. Local model and loss functions

We use a three-layer neural network to construct the local model. To address the fact that our framework can incorporate different graph neural network (GNN) models, two widely used GNNs, i.e., the graph convolutional network (GCN) and the graph attention network (GATN), are used in the hidden layer. The input of the model is the nodal states data matrix \mathcal{X} . For discrete dynamics, one-hot coding is used as the inputs. For continuous dynamics, the continuous dynamical variables are taken directly as the inputs. The states of the nodes are embedded into a d -dimensional feature space through a linear layer $f_1 : R^S \rightarrow R^d$, where S denotes the dimension of the inputs. The second layer utilizes GCN or GATN to aggregate the information from the neighbors of a node. The model outputs the prediction \hat{Y} through a linear layer.

The output of the discrete dynamics is the normalized probability vector \hat{P}_i that the node i belongs to different discrete states, with the element $\hat{p}_{i,m}$ being the prediction probability of node i in the m th state. The state with the highest probability in the vector is taken as the predicted state $\hat{y}(i)$ of node i . The cross-entropy (CE) loss function is used for discrete dynamics and is defined as

$$L_{\text{CE}}(P_i, \hat{P}_i) = - \sum_m p_{i,m} \ln \hat{p}_{i,m}, \quad (5)$$

where P_i is the one-hot coding of the true state of node i and $p_{i,m}$ is the m th element of the vector.

The output of the continuous dynamics \hat{y} is a one-dimensional continuous variable. For continuous dynamics prediction, we use the mean-square-error (MSE) loss function:

$$L_{\text{MSE}}(y(i), \hat{y}(i)) = (y(i) - \hat{y}(i))^2. \quad (6)$$

The specific architectures of the neural networks for the discrete dynamics and continuous dynamics are summarized in Table I.

C. Weighted aggregation of parameters

We introduce a unified method to evaluate the quality of the local data. Assuming that the amount of time series data owned by each client k is $|D_k|$ and the amount of network structure data corresponds to the number of edges in those data, denoted by $|E_k|$, we set the aggregation weight in model k as

$$w_k = \frac{1}{2} \left(\frac{|D_k|}{|D_1| + \dots + |D_K|} + \frac{|E_k|}{|E_1| + \dots + |E_K|} \right). \quad (7)$$

For scenario 1, each client k has the time series data of all nodes at different time stamps (i.e., T_k) and the partial structure of the network (G_k). The aggregation weight in Eq. (4) can be rewritten as

$$w_k = \frac{1}{2} \left(\frac{T_k}{T_1 + \dots + T_K} + \frac{|E_k|}{|E_1| + \dots + |E_K|} \right). \quad (8)$$

For scenario 2, all clients have the same network structure: $G_k = G$, $i = 1, 2, \dots, K$, so it is not necessary to consider the data quality. The length of time series data in all clients is the same (the whole length of the original data), but each client only records the time series data on a subset of nodes. Consequently, the amount of data in each client k can be simply denoted as $|V_k|$, yielding

$$w_k = \frac{|V_k|}{|V_1| + \dots + |V_K|}. \quad (9)$$

D. Evaluation metrics

For discrete dynamics, we use the accuracy index (ACC) to measure the prediction accuracy, defined as

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

where $\mathbf{I}(\cdot)$ is an indicator function and $|V_T|$ is the number of nodes in the test set V_T . A larger value of ACC indicates a higher prediction capability of the model.

For continuous dynamics, the differences between the true and predicted values of the dynamical variables are taken to be the prediction error. We choose two metrics to characterize the error: the mean-square error (MSE) and the mean absolute percentage error (MAPE). In particular, we use the MAPE to quantify the error for the mutualistic interaction dynamics among species in ecology (Mutualistic) and the gene regulatory dynamics (Gene). For coupled map lattice dynamics (CML), the values of the nodal dynamical variables

TABLE II. Five-step prediction performance of the GCN model—a basic component of the FGNN—for different types of network dynamics.

	ACC (discrete dynamics)				σ (continuous dynamics)		
	SIR	SIS	Threshold	Kirman	Gene	Mutualistic	CML
$T + 1$	0.87	0.85	0.80	0.92	0.672	1.168	0.025
$T + 2$	0.83	0.78	0.75	0.85	0.756	1.460	0.024
$T + 3$	0.78	0.75	0.73	0.83	0.820	1.647	0.024
$T + 4$	0.82	0.71	0.74	0.79	0.963	1.740	0.033
$T + 5$	0.80	0.70	0.72	0.81	1.001	1.718	0.030

are in the unit interval, and $y(i)$ is in the denominator of the MAPE metric. The value of the MAPE can be large when $y(i)$ converges to zero; so we use the MSE metric for CML, where σ is defined as

$$\sigma = \begin{cases} \frac{1}{|V_T|} \sum_{i \in V_T} \frac{|y(i) - \hat{y}(i)|}{y(i)}, & \text{Mutualistic or Gene} \\ \frac{1}{|V_T|} \sum_{i \in V_T} (y(i) - \hat{y}(i))^2, & \text{CML.} \end{cases} \quad (11)$$

III. PERFORMANCE CHARACTERIZATION AND DEMONSTRATION

A. Generation of training data

First, the time series data of all nodes are generated by certain dynamics. If a dynamical process leads to a steady state, it is not possible to uncover the network dynamics. To ensure that sufficiently long time series data can be obtained, the states of nodes are reinitialized after some steps of evolution so as to prevent the states of nodes from entering any steady state. The states of the first time stamps are stored in the data

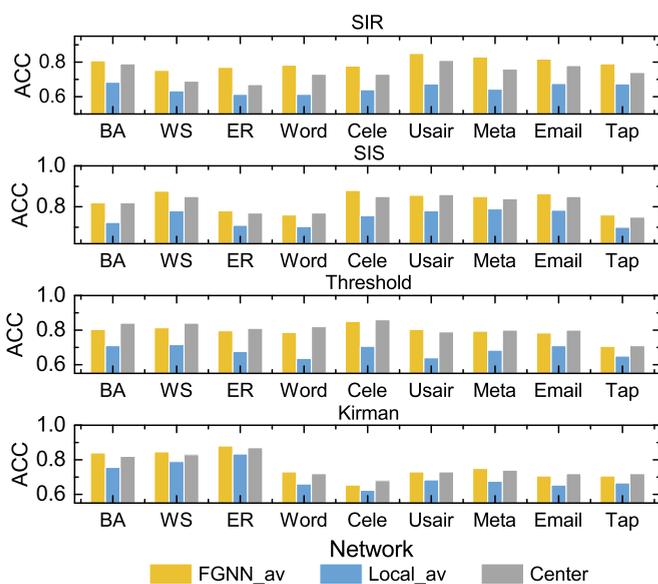


FIG. 2. Comparison of prediction performance of FGNN, local, and centralized models for discrete dynamics for scenario 1. Shown are the average values of ACC from different models. The FGNN models significantly outperform the local models and demonstrate a similar performance level to that of the centralized model but with the desired advantage of full privacy protection.

matrix \mathcal{X} as the input of the model, and the next time stamps are stored in \mathcal{Y} as the real states of nodes. For each client, the time series data are generated by intercepting the original data. Specifically, in scenario 1, each client uses the data of all nodes in different time stamps as the training set. In scenario 2, each client records the whole length of the time series data on some nodes only, and the states of the missing nodes in the client are set as zero. This may lead to conflicts with the state of nodes for discrete dynamics. Our solution is to assign continuous values to the discrete states when training a discrete dynamics model in scenario 2. By so doing, the model and train method are the same as for continuous dynamics. In addition, to reflect that each client only knows the partial structure of the original network in scenario 1, we randomly remove some edges following a uniform distribution.

B. Simulation settings of the FGNN

To present our results in a concise and clear way, here in the main text we include results with the GCN (results with GATN are deferred to Appendix C). Numerical experiments are performed on three synthetic networks (scale-free (BA) [57], small-world (WS) [58], and Erdős-Rényi (ER) random [59]) and six real-world networks: Word [60], *Caenorhabditis elegans* (Cele) [61], U.S. air transportation (USAir) [62], Metabolic (Meta) [63], Email [62], and Tap [64] (see Appendix A for the structural information of these networks).

For dynamical processes on networks, we use a diverse array of discrete and continuous dynamics to demonstrate the general applicability of our FGNN framework. In particular, we test four types of discrete dynamics (susceptible-infected-recovered (SIR) epidemic spreading dynamics [65], susceptible-infected-susceptible (SIS) dynamics [66], threshold dynamics [67], and Kirman dynamics [68]) and three types of continuous processes (i.e., gene regulatory dynamics (Gene) [69], mutualistic interaction dynamics among species in ecology (Mutualistic) [70], and coupled map lattice dynamics (CML) [37,71,72]). Detailed descriptions of these dynamical processes are summarized in Appendix B.

In our simulations, the number of clients is $K = 3$, where each client trains a local model individually based on the local data. The local models can be conveniently chosen as the baseline models, which are referred to as Local_1, Local_2, and Local_3 for the three clients, respectively. A centralized model (referred to as Center) is trained by using the full network structure and the full time series data. The model Center serves only the purpose of comparison; it has no privacy protection as it utilizes all information of

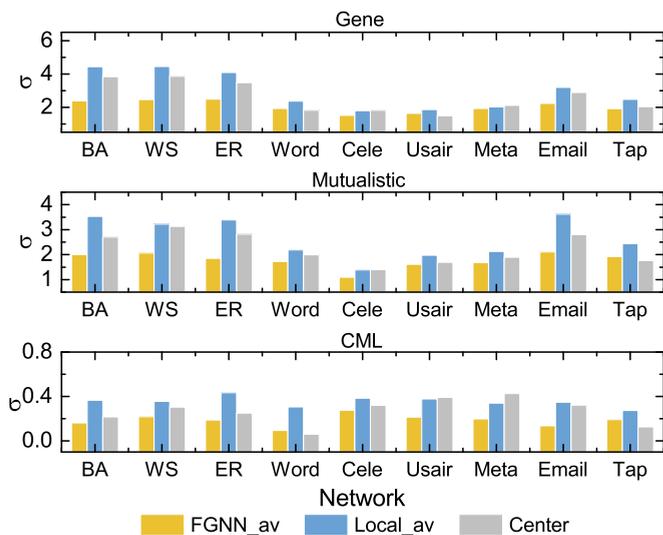


FIG. 3. Comparison of prediction performance of FGNN, local, and centralized models for continuous dynamics for scenario 1. Shown are the average errors σ for different models. As for the case of discrete dynamics in Fig. 2, the FGNN models significantly outperform the local models and present a similar performance level to that of the centralized model but without compromising privacy.

the clients. Unless specified otherwise, the number of iterations for federated aggregation in our FGNN framework is set to be 10, and the learning rates of training the GCN and GATN models are 0.001 and 0.0001, respectively. The test set contains 20 time pairs $(t, t + 1)$ of data. To reduce the statistical uncertainties, the values of the results are averaged over 20 realizations.

C. Results for simulated network dynamics

In our FGNN framework, a basic component is a centralized GNN. To demonstrate that GNNs are capable of predicting the network dynamics, we use the complete time series data and network structure to train the parameters in the GNNs. The length of the time series in discrete dynamics is set as $T = 200$, and that for continuous dynamics is $T = 100$. In the trained GNNs model, the nodal states at time T are then taken as the inputs with the states at time $T + 1$ as the

prediction. Then, the predicted states at $T + 1$ are inputs to the model to yield the predicted states at $T + 2$, and so on. Table II presents the five-step prediction results of the GCN model on a scale-free network for time from $T + 1$ to $T + 5$. It can be seen that, for the four types of discrete dynamics, when the true states of nodes are inputted at T , the value of ACC at $T + 1$ is persistently larger than 80% (for the Kirman dynamics, the value is about 92%). Subsequently, when the predicted states are used as inputs, the ACC values somewhat decrease, but they are still above 70% for the next five time steps. For the continuous dynamics, Table II shows the error σ between the predicted and true state values for times from $T + 1$ to $T + 5$.

The results from predicting the continuous dynamics indicate uniformly small prediction errors, regardless of the specific types of processes. The results in Table II thus demonstrate that the GCN model is capable of making accurate short-term state predictions for different types of network dynamical processes, paving the way for the model to be incorporated into our federated learning framework in which the prediction task is accomplished by fusing data from different clients with their privacy fully protected.

We can now present the prediction results of our FGNN framework. For scenario 1 of data collection, each client k has its own network structure G_k , which is not shared with others, leading to three global prediction models ($K = 3$), denoted as $FGNN_k = M(\mathcal{X}, \theta, G_k)$ for $k = 1, 2, 3$. For this case, the sampling probabilities of edges in the three clients are set as 80, 60, and 50%, respectively. All clients are assumed to have the same nodal set, and the nodes without edges are treated as the isolated nodes. For the four types of discrete dynamics, the lengths of the time series in the three clients are set as 50, 30, and 20, respectively. For the three types of continuous dynamics, the lengths of time series data in the three clients are set as 20, 15, and 15, respectively. The three local models [$Local_k = M_k(X_k, \theta_k, G_k)$] and the centralized model (Center) serve as the baseline models for comparison. For clarity, here we present the average results from the three FGNNs and three local models, denoted as FGNN_av and Local_av, respectively.

Figure 2 shows, for data scenario 1, the ACC values for the FGNN, local, and centralized models in predicting the discrete dynamics taking place on different networks. It can

TABLE III. For scenario 2, prediction performances of different models in terms of the MSE for four types of discrete dynamics for synthetic and real-world networks. The best performance in each row is highlighted in bold.

Dynamic	SIR			SIS			Threshold			Kirman		
	FGNN	Local_av	Center	FGNN	Local_av	Center	FGNN	Local_av	Center	FGNN	Local_av	Center
Scale-free	1.371	1.560	1.495	0.923	1.533	1.216	0.654	4.627	1.531	0.992	5.487	4.402
Small-world	1.111	1.619	1.530	0.818	1.331	1.065	0.449	0.767	0.473	0.432	0.924	0.466
ER random	1.328	1.768	1.654	1.024	1.341	1.311	0.663	1.972	1.279	0.722	1.012	0.790
Word	2.243	3.549	2.229	1.218	2.380	1.651	0.416	0.656	0.557	0.363	0.386	0.386
Cele	1.629	2.232	1.430	1.309	2.193	1.688	0.331	0.367	0.330	0.526	0.458	0.436
USAir	1.879	3.085	1.716	1.291	1.538	1.271	0.465	0.981	0.502	0.458	0.612	0.514
Meta	1.358	1.220	1.241	1.276	1.645	1.106	0.516	0.596	0.507	0.860	1.815	0.468
Email	1.255	2.042	1.666	1.140	1.522	1.058	0.422	0.535	0.514	0.463	0.594	0.494
Tap	1.206	1.215	1.370	1.452	1.722	1.437	0.501	0.878	0.779	0.633	0.839	0.918

TABLE IV. For scenario 2, prediction performances of different models in terms of σ for three types of continuous dynamics for synthetic and real-world networks. The best performance in each row is highlighted in bold.

Dynamic method	Gene			Mutualistic			CML		
	FGNN	Local_av	Center	FGNN	Local_av	Center	FGNN	Local_av	Center
Scale-free	1.165	1.432	1.406	1.245	1.781	1.480	0.529	0.839	0.426
Small-world	1.331	1.714	1.648	1.815	2.419	1.941	0.358	0.990	0.356
ER random	0.797	0.878	0.766	1.577	1.847	1.616	0.648	0.888	0.395
Word	0.800	1.046	0.870	1.113	1.275	1.223	0.538	0.959	0.581
Cele	0.889	1.448	0.793	1.574	2.003	2.098	0.404	0.878	0.416
USAir	1.230	1.393	1.415	1.619	2.192	2.240	0.481	0.829	0.645
Meta	1.402	1.323	1.076	1.992	1.967	1.719	0.462	0.702	0.477
Email	1.161	1.480	1.289	1.522	2.152	1.886	0.298	0.325	0.281
Tap	1.322	1.540	1.587	2.372	2.921	2.779	0.413	1.376	1.279

be seen that the FGNN models persistently outperform the local models in terms of the ACC values and can present a similar performance level to that of the centralized model. Considering that there is no privacy protection in the centralized model as it requires complete time series data and complete information about the network structure, our FGNN models are desired as they offer full privacy protection. Figure 3 further compares the performance of different models in predicting the continuous network dynamics in terms of the measure σ , which again indicates that our FGNN framework gives significantly better prediction results than those from the local models and similar performance to that of the centralized model. The results in Figs. 2 and 3 thus demonstrate that our FGNN framework can effectively combine data from multiple parties or fuse local data information to train a better global network dynamic model without disclosing private data, regardless of the types of network dynamics and structure. For data scenario 2, different clients collect the time series data from a subset of nodes, but each client possesses the same global network structure, so there is only one global federated learning model, denoted as $FGNN = M(\mathcal{X}, \theta, G)$. For this case, the length of the time series for the four types of discrete dynamics is set to be 50, and the length of time series data for the three types of continuous dynamics is set to be 20. In addition, the percentages of the nodes with data in the three clients are 70, 80, and 80%, respectively. As we have mentioned in Sec. III B, for scenario 2, we treat the discrete state as the continuous state to avoid the adverse consequence. Therefore the MSE index is selected as the evaluation metric for the discrete network dynamics rather than the ACC index. Table III presents the prediction results of discrete network dynamics in terms of the MSE from different models. It can be seen that in most cases the prediction results of our FGNN framework with privacy protection are better than those of the local models. For the centralized model, in spite of its use of the complete data, the prediction performance is not significantly better than that of our FGNN framework. Table IV illustrates the prediction results for the three types of continuous dynamics from different models. Similar to the results in Table III, our FGNN model can predict the true nodal states better than the local models, and its performance is close to that of the centralized model.

To further demonstrate the robustness of our FGNN framework, we study the impacts of the number of clients, the length of time series data, and the embedding dimension d in the linear layer on the prediction performance. To be illustrative, we consider scenario 1 and use the SIR dynamics on scale-free networks. Figure 4(a) shows the effect of the number of clients, for $K = 3$ (the sampling probabilities of edges are 80, 60, and 50%, and the lengths of the time series are 50, 30, and 20), $K = 4$ (the sampling probabilities of edges are 80, 60, 50, and 60%, and the lengths of the time series are 50, 30, 20, and 15), and $K = 5$ (the sampling probabilities of edges are 80, 60, 50, 60, and 50%, and the lengths of the time series are 50, 30, 20, 15, and 10). As the number of clients increases, the performance of the local models decreases, as each client possesses less information, i.e., fewer time series data points and less structural information about the network. In contrast, with more clients, our FGNN model

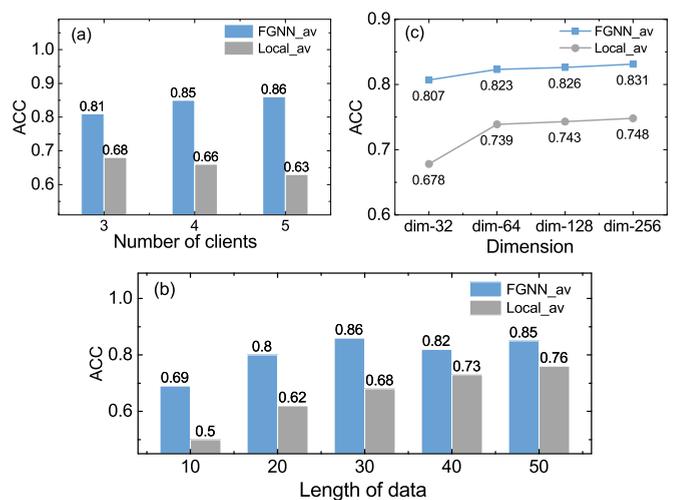


FIG. 4. Impacts of the number of clients (a) and the length of time series (b) and the embedding dimension (dim) in the linear layer (c) on the prediction performance for different models. The dynamical process is of the SIR type, the network is of the scale-free type, and data collection follows scenario 1. In all cases, our FGNN model yields significantly better results than the local models.

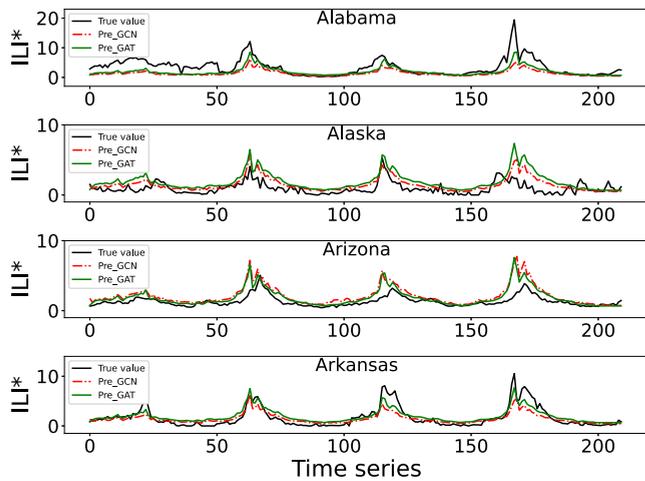


FIG. 5. Performance of two types of GNNs (without privacy protection) in predicting the evolution of the influenza spreading in four states in the USA. The black curves are the ground truth, and the red dash-dotted and green curves are the predicted ILI^* values using the GCN (Pre_GCN) and GATN (Pre_GAT).

yields increasingly better performance—an intrinsic feature of federated learning in general. Figure 4(b) presents the effect of the time series length on the prediction performance for the case of $K = 3$, where the length for each client is the same for each simulation (10, 20, 30, 40, and 50). There is no overlapping in the data for different clients. For example, if the total length of the time series is $T = 30$ and there are three clients, we set the data length for each client to be 10. The sampling probabilities of edges in the three clients are set as 80, 60, and 50%, respectively. It can be seen from Fig. 4(b) that the ACC values from our FGNN model are persistently higher than those from the local models for all cases. For shorter time series, the advantage of our FGNN model is more pronounced. As expected, as the data length increases, the prediction performance of the local models is improved as each client has more data to train the neural network. Figure 4(c) shows the impact of the embedding dimension for the case of $K = 3$; that is, the embedding dimension d is set as 32, 64, 128, and 256, respectively, the sampling probabilities of edges are 80, 60, and 50%, and the lengths of the time series are 50, 30, and 20. As demonstrated in Fig. 4(c), the performance of the FGNN model is robust to the embedding dimensions; in particular, its performance is always better than that of the local model for different embedding dimensions.

TABLE V. MSE values of the global FGNN model and the baseline models under data scenario 1 in predicting the evolution of influenza in four states in the USA.

MSE	FGNN_1	FGNN_2	FGNN_3	Local_1	Local_2	Local_3	Center
GCN	1.591	1.642	1.570	1.678	1.756	1.741	1.594
GATN	0.685	0.814	0.822	1.413	1.394	1.506	1.271

D. Results for real-world network dynamics: Predicting influenza evolution

We further apply our FGNN framework to predicting the dynamical states for a real-world problem: the outbreak of influenza. The time series are from the U.S. weekly influenza-like illness (ILI) database [73] for the 5-year time period from the 40th week of 2011 to the 39th week of 2016, which records the weekly number of ILI-related visits to all public health and clinical laboratories in the country by the Centers for Disease Control and Prevention (CDC). For data preprocessing, we normalize the weekly ILI-related visits in each state to calculate the ILI ratio, i.e., the percentage of the patients with influenza among all visiting patients in each state [74]. We then use the data values in which the percentage of the ILI ratio is removed to represent the state of the underlying dynamical system, recorded as ILI^* . The underlying network supporting the influenza spreading across the different states in the USA is identified by taking advantage of the commuting traffic data in different states in 2015 to generate a network of the population commuting between the states [75]. The raw data are available from the website of the U.S. Bureau of Statistics, and the data for commuting between the states are obtained by aggregating the raw data (a detailed description of data processing and the data settings is provided in Appendix D).

As for the various cases of synthetic dynamic data treated in Sec. III C, the first step is to validate the effectiveness of the GNNs. For this purpose, we use the influenza data at the *previous time steps* to train the GNNs. The trained GNNs and the ILI^* of nodes (i.e., states) at the *current time step* are the inputs for predicting the trend of the influenza at the *next time step*. Figure 5 shows the true and the predicted values in terms of ILI^* for the first four states in an alphabetic list of U.S. states. It can be seen that the predicted values fit well the real evolution of the influenza in these states, and the peaks of the influenza at several representative time points (i.e., outbreaks) can also be predicted. The results in Fig. 5 thus demonstrate that the predicted dynamics can fit the evolution of the influenza to a satisfactory extent.

Having demonstrated the performance of the two GNNs, we can proceed to test the predictive power of our FGNN framework and that of the baseline methods with respect to the two data scenarios. The number of clients is set to be $K = 3$. For data scenario 1, partial time stamps of all nodes are collected, and the information about the structure of the commuting network is incomplete. For scenario 2, the time series data of partial nodes are used. Table V shows, for scenario 1, the MSE values associated with the FGNN predictions with the GCN and GATN models, the local models, and the

TABLE VI. MSE values from the global FGNN and the baseline models under data scenario 2 in predicting the evolution of influenza in four states in the USA.

MSE	FGNN	Local_1	Local_2	Local_3	Center
GCN	1.600	1.622	2.584	2.499	1.852
GATN	1.227	1.646	1.352	1.808	1.684

centralized model. Regardless of whether the global model is GCN or GATN, our FGNN models of the three clients (FGNN_1, FGNN_2, and FGNN_3) can achieve better prediction performance than those of the local models (Local_1, Local_2, and Local_3). Table VI summarizes the MSE values for the FGNN and baseline methods for scenario 2. It can be seen that our global FGNN model also yields significantly better prediction results than the baseline models.

IV. DISCUSSION

In our modern times, time series data generated from network dynamical processes and the underlying structure of the network are typically owned by different parties. There are two possible data distribution scenarios that are amenable to simulations and analysis at the present: scenario 1, in which different clients have the data time stamps of all nodes but each client has only partial information about the structure of the network, and scenario 2, in which different clients have the data from only a subset of nodes in the network but each client has full knowledge of the network structure. To combine multiparty data to better predict network dynamics while protecting the privacy of the clients is a problem of great importance and interest. We have proposed an FGNN framework to address this problem. The essence of our prediction framework is to combine the data from different clients to jointly train a high-quality global model without disclosing any private data. The framework is compatible with existing GNNs to learn network dynamics and predict the states of nodes into the near future. We have used two classical GNN models (GCN and GATN) to demonstrate the power and effectiveness of our FGNN framework through extensive numerical simulations of a good number of synthetic and real-world networks as well as a variety of discrete and continuous network dynamics. Our systematic comparison of the performances of our global FGNN model and baseline local and centralized models under different conditions reveals that the global model learned through the FGNN framework has superior accuracy in predicting the dynamical evolution of the network and is thus capable of better capturing the complex relationship between dynamics and network structure. The impacts of the number of clients, the length of time series data, and the embedding dimension on the dynamics prediction have been investigated, revealing the general applicability and robustness of our FGNN framework.

A practically significant contribution of our work is the demonstration of successful prediction of the evolutionary trend of influenza in the USA by our articulated FGNN framework without compromising privacy. In particular, any state in the USA is regarded as a node in the network which we have reconstructed using the state-crossing commuting data, and

TABLE VII. Basic structural information of six real networks.

Network	n	M	$\langle k \rangle$	CC	H
Word	112	425	7.589	0.173	1.815
Cele	297	2148	14.465	0.292	1.801
USAir	332	2126	12.807	0.749	3.464
Metabolic	453	2025	8.940	0.647	4.485
Email	1133	5451	9.622	0.220	1.942
Tap	1373	6833	9.953	0.529	1.644

the short-term dynamical evolution of the influenza spreading dynamics on this network has been predicted. Our results indicate that not only is the FGNN framework capable of reproducing the actual time evolution of the influenza in the states, but also the outbreaks (corresponding to peaks of the dynamical evolution) can be faithfully predicted.

The two data scenarios treated in this paper are somewhat specific. A more general scenario is that each client has partial time series data from a subset of nodes and incomplete information about the network structure. To predict the dynamical evolution of the network without compromising data privacy under this general scenario is an open question warranting further investigation.

ACKNOWLEDGMENTS

The authors thank Dr. Sen Pei for sharing the website of the influenza data in the USA. H.-F.Z. acknowledges the National Natural Science Foundation of China (Grant No. 61973001) and the University Synergy Innovation Program of Anhui Province (Grant No. GXXT-2021-032). L.L. acknowledges the MOST 2030 Brain Project (Grant No. 2022ZD0211400), the Xplorer Prize, and the Special Project for the Central Guidance on Local Science and Technology Development of Sichuan Province (Grant No. 2021ZYD0029). The work at Arizona State University was supported by AFOSR under Grant No. FA9550-21-1-0438.

APPENDIX A: STRUCTURAL INFORMATION OF THE NETWORKS

Experiments are performed on three synthetic networks and six real-world networks. The sizes of the synthetic networks are 100 nodes. In particular, the scale-free (BA) networks are generated with $m = 2$, where m is the number of edges connecting to the existing nodes at each preferential attachment step. The small-world (WS) networks are generated with the average degree $\langle k \rangle = 4$ and reconnection probability $p = 0.3$. The Erdős-Rényi (ER) random networks are generated with the edge connection probability $p = 0.08$. The structural information of the six real-world networks is summarized in Table VII, where n and M are the numbers of nodes and edges of the network, respectively, CC is the clustering coefficient, $H = \langle k^2 \rangle / \langle k \rangle^2$ is the network heterogeneity, and $\langle k^2 \rangle$ is the second moment of the degree distribution.

APPENDIX B: SEVEN TYPES OF NETWORK DYNAMICS

We describe the four types of discrete and three types of continuous networked dynamical processes used in this paper,

TABLE VIII. Five-step dynamics prediction results of the GATN model based on complete time series data and complete network structure.

	ACC (discrete dynamics)				σ (continuous dynamics)		
	SIR	SIS	Threshold	Kirman	Gene	Mutualistic	CML
$T + 1$	0.85	0.86	0.89	0.84	0.598	0.958	0.017
$T + 2$	0.73	0.80	0.84	0.81	0.602	1.086	0.021
$T + 3$	0.81	0.75	0.81	0.82	0.609	1.276	0.024
$T + 4$	0.82	0.74	0.74	0.83	0.724	1.512	0.027
$T + 5$	0.80	0.74	0.72	0.85	0.822	1.601	0.028

which all have been well studied to gain insights into a variety of network phenomena in natural or social sciences.

SIR dynamic model. In the SIR model [65], at any time a node can be in one of three states: susceptible (S), infectious (I), and recovery (R). The I -state node can infect its S -state neighbors with the infection rate λ . If an infection event is successful, the infected node will change its state from S to I ; otherwise it will remain in the S state. The infected nodes will recover to the R state with the recovery rate μ , and the R -state nodes will not be infected again. In this paper, the parameter values are set as $\lambda = 0.2$ and $\mu = 0.1$. The states of all nodes are randomly initialized after every five time steps.

SIS dynamic model. In the SIS model [66], there are two distinct dynamical states only: S and I . An I -state node can infect its susceptible neighbors with the infection rate λ and recover to the S state with the recovery rate μ . In this paper, the parameter values are set as $\lambda = 0.2$ and $\mu = 0.1$. The

states of all nodes are randomly initialized after every ten time steps.

Threshold dynamic model. In the threshold model [67], nodes can be in one of two states: inactive (0) or active (1). An inactive node is activated when the fraction of its active neighbors is greater than an activation threshold of the node, and an active node will not be restored again. In this paper, the activation threshold for all nodes is set to be 0.5, and the states of all nodes are reinitialized after every five time steps.

Kirman dynamic model. In the Kirman model [68], a node can be in one of two states: 0 or 1. The transition between the two states is based on two transfer functions. In particular, the transfer function of the node from 0 to 1 is given by $c_1 + dm_1$, and that from 1 to 0 is $c_2 + d(k - m_1)$, where c_1 and c_2 quantify the individual behavior of the node that is independent of the neighbors' states, k is the nodal degree, d describes the probability of the node replicating a

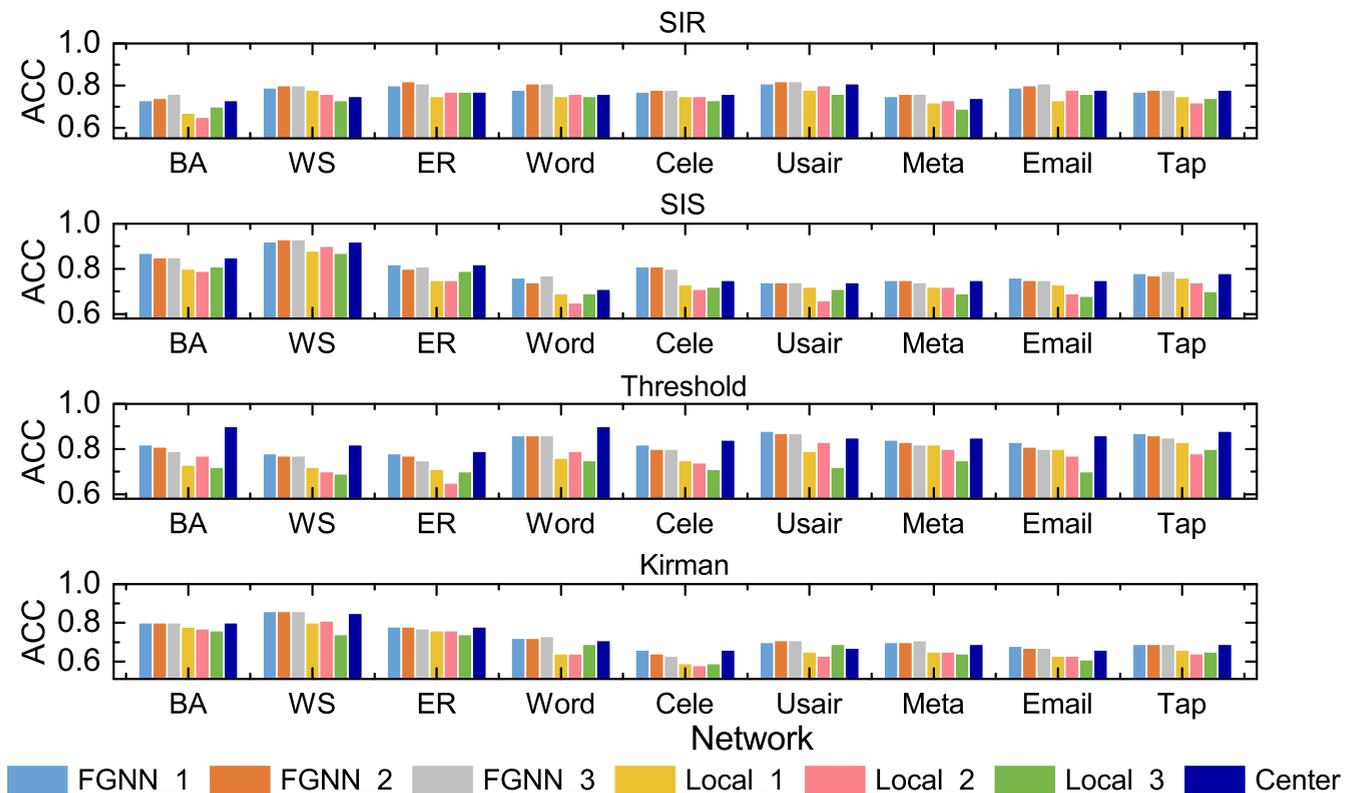


FIG. 6. GATN-based FGNN model performance. Shown are the values of ACC from the global models and the baseline methods for the four types of discrete dynamical processes under data scenario 1.

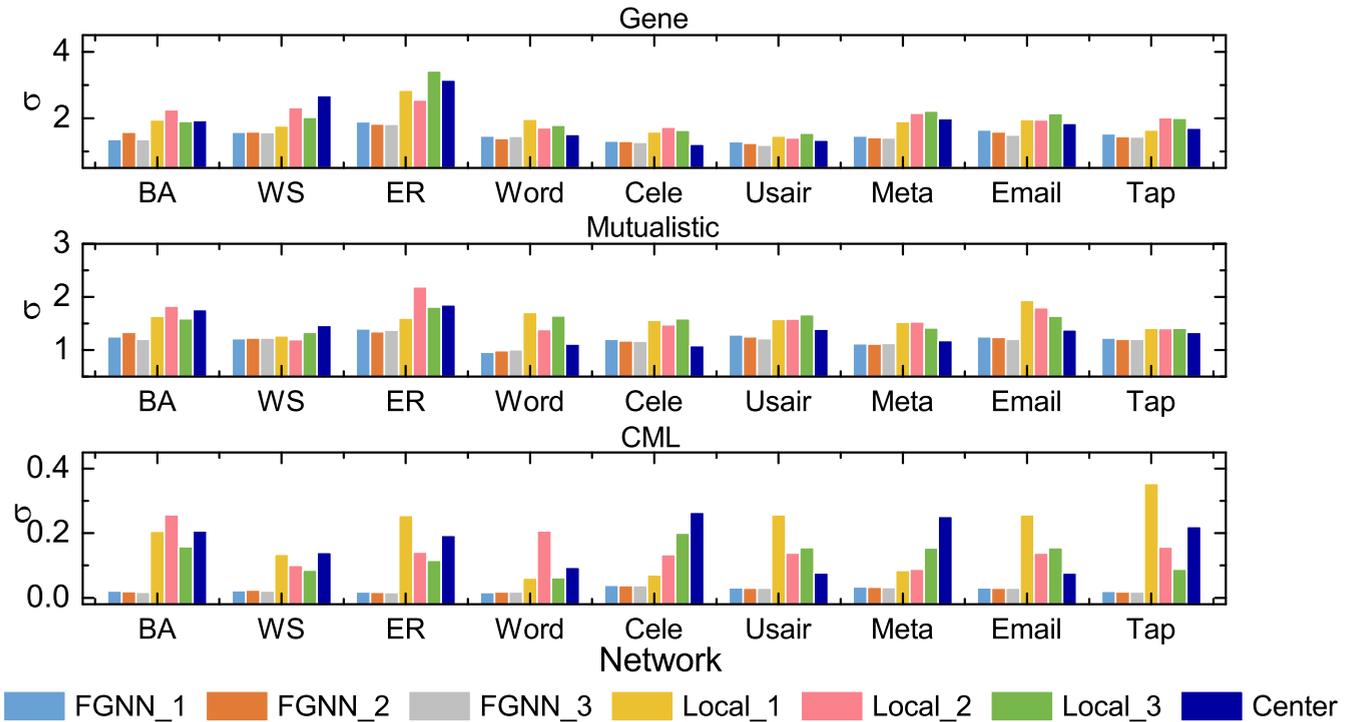


FIG. 7. GATN-based FGNN model performance. Shown are the prediction error σ from the global models and the baseline methods for the three types of continuous dynamical processes under data scenario 1.

neighbor’s state, and m_1 is the number of “1”-state neighbors. In our study, the parameters are set as $c_1 = 0.1$, $c_2 = 0.1$, and $d = 0.08$.

The continuous networked dynamics employed in this paper are one-dimensional with the state variable $x_i(t) \in R$ for node i at time t . The states of all nodes at time t can be represented by the vector $X(t) = [x_1(t), \dots, x_n(t)]^T \in R^n$. The differential equations governing the three types of continuous dynamics in this paper are as follows.

Gene regulatory dynamics (Gene). The gene regulation dynamics are expressed by the Michaelis-Menten equation [69]:

$$\frac{dx_i(t)}{dt} = -u_i x_i(t) + \sum_{j=1}^n A_{ij} \frac{(x_j(t))^h}{(x_j(t))^h + 1}, \quad (B1)$$

where the first item controls the decay of the current node state, u_i is the decay rate, and the second term captures gene activation characterized by the Hill coefficient h . We use $u_i = 1$ and $h = 2$. The states of all nodes are reinitialized after 50 time steps.

Mutualistic interaction dynamics (Mutualistic). The differential equations governing the evolution of mutualism in ecological systems [70,76] are

$$\begin{aligned} \frac{dx_i(t)}{dt} = & u_i + x_i(t) \left(1 - \frac{x_i(t)}{l_i} \right) \left(\frac{x_i(t)}{z_i} - 1 \right) \\ & + \sum_{j=1}^N A_{ij} \frac{x_i(t)x_j(t)}{\alpha_i + \beta_i x_i(t) + \gamma_i x_j(t)}, \end{aligned} \quad (B2)$$

TABLE IX. Values of the MSE from the global and baseline models for the four types of discrete dynamics for the GATN-based FGNN under data scenario 2.

Dynamic	SIR			SIS			Threshold			Kirman		
	FGNN	Local_av	Center									
Scale-free	0.843	0.867	0.865	0.177	0.434	0.261	0.413	0.450	0.427	0.249	0.320	0.259
Small-world	0.897	1.034	1.220	0.146	0.172	0.136	0.245	0.301	0.311	0.259	0.308	0.342
ER random	0.649	0.669	0.655	0.242	0.267	0.211	0.224	0.261	0.302	0.242	0.301	0.265
Word	0.699	0.826	0.791	0.249	0.339	0.249	0.295	0.416	0.341	0.251	0.269	0.271
Cele	0.496	0.507	0.478	0.270	0.277	0.257	0.301	0.360	0.316	0.254	0.294	0.317
USAir	0.699	0.770	0.716	0.246	0.320	0.252	0.401	0.501	0.419	0.251	0.321	0.250
Meta	0.681	0.767	0.735	0.232	0.264	0.250	0.311	0.387	0.290	0.251	0.307	0.283
Email	0.696	0.749	0.771	0.251	0.291	0.284	0.257	0.366	0.759	0.251	0.316	0.290
Tap	0.632	0.669	0.701	0.253	0.288	0.254	0.250	0.284	0.245	0.246	0.298	0.347

TABLE X. Values of the prediction error from the global and baseline models for the three types of continuous dynamics for the GATN-based FGNN under data scenario 2.

Dynamic method	Gene			Mutualistic			CML		
	FGNN	Local_av	Center	FGNN	Local_av	Center	FGNN	Local_av	Center
Scale-free	1.160	1.316	1.122	1.741	1.668	1.636	0.123	0.202	0.192
Small-world	1.258	1.242	1.139	1.141	1.254	1.264	0.156	0.222	0.313
ER random	0.774	0.807	0.743	1.349	1.297	1.308	0.191	0.259	0.101
Word	0.774	1.045	0.884	1.655	1.675	1.442	0.229	0.240	0.146
Cele	0.686	0.912	0.778	1.005	1.344	1.316	0.176	0.177	0.172
USAir	0.891	1.041	0.945	1.165	1.385	1.361	0.146	0.174	0.148
Meta	0.990	1.111	1.006	1.245	1.430	1.481	0.151	0.214	0.088
Email	1.261	1.114	1.055	1.310	1.539	1.489	0.085	0.119	0.053
Tap	0.955	1.016	0.908	1.207	1.346	1.234	0.093	0.125	0.109

where the first item, u_i , represents the number of migrating individuals of species i from the adjacent habitat; the second item, l_i , describes the carrying capacity of the system growth; and z_i is the cold-start threshold. When the abundance of species i is low [i.e., $x_i(t) < z_i$], the system is characterized by a negative growth. The third item in Eq. (B2) describes the interactions among the species, which take place on a network described by the adjacency matrix \mathcal{A} . In this paper, we use the parameter values $u_i = 0.1$, $l_i = 5$, $z_i = 1$, $\alpha_i = 5$, $\beta_i = 0.9$, and $\gamma_i = 0.1$. The states of all nodes are reinitialized after every 50 time steps.

Coupled map lattice dynamics (CML). In a coupled mapping lattice, the continuous state variables are updated at discrete times [37,71,72] according to

$$x_i(t+1) = (1-s)f(x_i(t)) + \frac{s}{k_i} \sum_{j=1}^N A_{ij} f(x_j(t)), \quad (\text{B3})$$

where s is the coupling parameter (the system degenerates into a set of independent mapping functions for $s = 0$) and k_i is the degree of node i . In this paper, we use the mapping function $f(x) = \lambda x(1-x)$ for the parameters $\lambda = 3.5$ and $s = 0.2$. The states of all nodes are reinitialized after every 50 time steps.

APPENDIX C: GATN SIMULATION RESULTS

We present the detailed simulation results of the GATN-based FGNN model and the local model of each client. Table VIII verifies the good performance of the GATN model in predicting the four discrete dynamics ($T = 200$ for GATN training) and the three continuous dynamics ($T = 100$ for GATN training) with the complete time series data and the complete network structure (for scale-free networks). Figure 6

shows the values of ACC from the FGNN global models and the baseline methods for the four types of discrete dynamics under data scenario 1. Figure 7 shows the prediction results for the three types of continuous dynamics, which also suggest that the FGNN global models have lower prediction errors than those of the local models. For data scenario 2, we present the average results from the three local models, denoted as Local_av. The results in Table IX demonstrate that the performance of our FGNN method in predicting the four types of discrete dynamics is persistently better than those of the local models. Table X shows the results of different methods in predicting the three types of continuous dynamics. In most cases, the prediction results of our FGNN method are closer to the real values than those of the local models.

APPENDIX D: INFLUENZA DATA

1. Description of influenza data

The influenzalike illness (ILI) data used in this paper are from U.S. public health and clinical laboratories [73] and were collected weekly from the 40th week of 2011 to the 39th week of 2016. We standardize the weekly ILI data by using the number of visiting patients in each state in the USA to calculate the ILI ratio [74]. In order to ensure a sufficient sample size in the data, we assume that the states whose average weekly ILI ratio is less than 1% are not representative. Accordingly, 14 states were removed. The remaining 37 states are listed in Table XI. Because the small value of the ILI ratio (about 2%) can lead to errors, we multiply the ILI ratio by a factor of 100 (labeled as ILI*) and use the resulting values for the dynamic data. The curves of ILI* for the 37 states are shown in Fig. 8. The population commuting between the states is the key to spreading dynamics. We use the commuting data in the USA by residential geography to construct the commuting network.

TABLE XI. Names of the 37 U.S. states for which data were used in this paper.

Alabama	Alaska	Arizona	Arkansas	California	Connecticut	Columbia
Georgia	Hawaii	Idaho	Illinois	Indiana	Kansas	Louisiana
Maryland	Massachusetts	Michigan	Minnesota	Mississippi	Missouri	Nebraska
Nevada	New Jersey	New Mexico	New York	North Carolina	North Dakota	Oklahoma
Pennsylvania	South Dakota	Tennessee	Texas	Utah	Vermont	Virginia
West Virginia	Wisconsin					

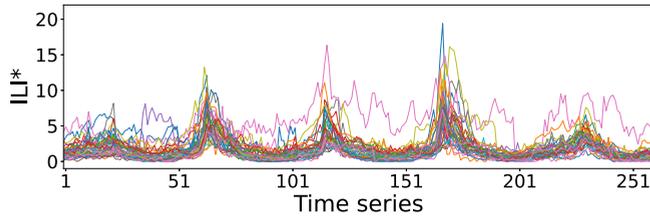


FIG. 8. Real ILI* values in 37 states from the 40th week of 2011 to the 39th week of 2016.

The commuting data between the U.S. cities are from the 2015 Census report [75]. The state level commuting data are calculated by aggregating the city data, leading to a state-level commuting network for the 37 states. The network is directed and weighted, where the nodes represent the states and the 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 edge is removed. The resulting network has 763 directed edges. Finally, we use the min-max normalization procedure [77] to obtain the weights of the edges by dividing the number of commuters by the maximum number of commuters. The constructed directed and weighted network can be visualized, as shown in Fig. 9, where the thickness of the directed edges denotes the weight.

2. Simulation settings of influenza data

Under scenario 1, the first 100 data points are selected from a total of 260 ILI* data points. The numbers of data

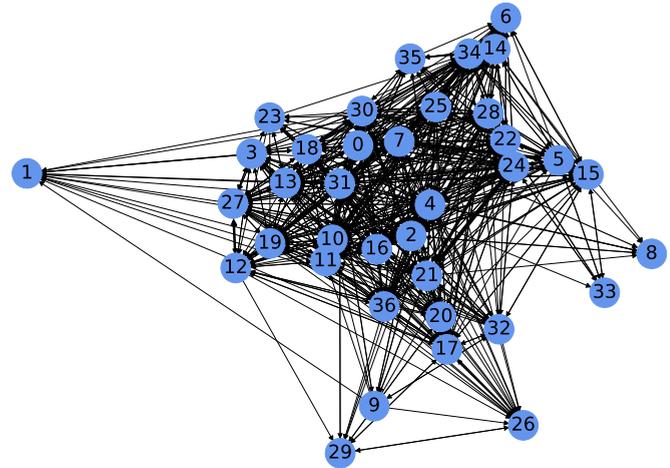


FIG. 9. The directed and weighted network constructed from the data for commuting between 37 U.S. states.

assigned to the three clients are 50, 30, and 20, respectively, and the network structures for the three clients are generated by sampling 90, 80, and 70% of edges from the commuting network. The learning rates of the GCN-based FGNN and GATN-based FGNN models are set to be 0.0001 and 0.001, respectively. Under data scenario 2, the first 50 data points are selected from a total of 260 ILI* data points. The percentages of nodes with the influenza data for the three clients are 80, 70, and 60%, respectively. The learning rates of the GCN and GATN models are set as 0.01 and 0.0001, respectively.

- [1] W.-X. Wang, Y.-C. Lai, and C. Grebogi, Data based identification and prediction of nonlinear and complex dynamical systems, *Phys. Rep.* **644**, 1 (2016).
- [2] H. 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); **13**, 3692 (2022).
- [3] D. Yu, M. Righero, and L. Kocarev, Estimating Topology of Networks, *Phys. Rev. Lett.* **97**, 188701 (2006).
- [4] D. Zhou, Y. Xiao, Y. Zhang, Z. Xu, and D. Cai, Causal and Structural Connectivity of Pulse-Coupled Nonlinear Networks, *Phys. Rev. Lett.* **111**, 054102 (2013).
- [5] M. Timme, Revealing Network Connectivity from Response Dynamics, *Phys. Rev. Lett.* **98**, 224101 (2007).
- [6] M. Timme and J. Casadiego, Revealing networks from dynamics: An introduction, *J. Phys. A: Math. Theor.* **47**, 343001 (2014).
- [7] M. Nitzan, J. Casadiego, and M. Timme, Revealing physical interaction networks from statistics of collective dynamics, *Sci. Adv.* **3**, e1600396 (2017).
- [8] C. Ma, H.-F. Zhang, and Y.-C. Lai, Reconstructing complex networks without time series, *Phys. Rev. E* **96**, 022320 (2017).
- [9] X. Li and X. Li, Reconstruction of stochastic temporal networks through diffusive arrival times, *Nat. Commun.* **8**, 15729 (2017).
- [10] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi, Predicting Catastrophes in Nonlinear Dynamical Systems by Compressive Sensing, *Phys. Rev. Lett.* **106**, 154101 (2011).
- [11] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and M. A. F. Harrison, Time-series-based prediction of complex oscillator networks via compressive sensing, *Europhys. Lett.* **94**, 48006 (2011).
- [12] Z.-S. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, Reconstructing hidden propagation networks with natural diversity and identifying hidden sources, *Nat. Commun.* **5**, 4323 (2014).
- [13] R.-Q. Su, Y.-C. Lai, and X. Wang, Identifying chaotic FitzHugh–Nagumo neurons using compressive sensing, *Entropy* **16**, 3889 (2014).
- [14] G. Mei, X. Wu, Y. Wang, M. Hu, J.-A. Lu, and G. Chen, Compressive-sensing-based structure identification for multi-layer networks, *IEEE Trans. Cybern.* **48**, 754 (2018).
- [15] Y.-C. Lai, Finding nonlinear system equations and complex network structures from data: A sparse optimization approach, *Chaos* **31**, 082101 (2021).
- [16] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, Machine learning in financial crisis prediction: A survey, *IEEE Trans. Syst., Man, Cybernet., Part C: Appl. Rev.* **42**, 421 (2011).
- [17] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Machine learning prediction of critical transition and system collapse, *Phys. Rev. Res.* **3**, 013090 (2021).

- [18] L.-W. Kong, H.-W. Fan, C. Grebogi, and Y.-C. Lai, Emergence of transient chaos and intermittency in machine learning, *J. Phys. Complex.* **2**, 035014 (2021).
- [19] N. Kumar and M. Raubal, Applications of deep learning in congestion detection, prediction and alleviation: A survey, *Transp. Res. Part C: Emerging Technol.* **133**, 103432 (2021).
- [20] J. D. Farmer and J. J. Sidorowich, Predicting Chaotic Time Series, *Phys. Rev. Lett.* **59**, 845 (1987).
- [21] M. Casdagli, Nonlinear prediction of chaotic time series, *Phys. D (Amsterdam)* **35**, 335 (1989).
- [22] G. Gouesbet, Reconstruction of standard and inverse vector fields equivalent to a Rössler system, *Phys. Rev. A* **44**, 6264 (1991).
- [23] T. Sauer, Reconstruction of Dynamical Systems from Interspike Intervals, *Phys. Rev. Lett.* **72**, 3811 (1994).
- [24] E. Baake, M. Baake, H. G. Bock, and K. M. Briggs, Fitting ordinary differential equations to chaotic data, *Phys. Rev. A* **45**, 5524 (1992).
- [25] U. Parlitz, Estimating Model Parameters from Time Series by Autosynchronization, *Phys. Rev. Lett.* **76**, 1232 (1996).
- [26] J. P. Crutchfield and B. McNamara, Equations of motion from a data series, *Complex Syst.* **1**, 417 (1987).
- [27] E. M. Bollt, Controlling chaos and the inverse Frobenius-Perron problem: global stabilization of arbitrary invariant measures, *Int. J. Bifurcation Chaos* **10**, 1033 (2000).
- [28] C. Yao and E. M. Bollt, Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems, *Physica D (Amsterdam)* **227**, 78 (2007).
- [29] E. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Inf. Theory* **52**, 489 (2006).
- [30] E. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, *Commun. Pure Appl. Math.* **59**, 1207 (2006).
- [31] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J.-P. Ye, Network Reconstruction Based on Evolutionary-Game Data via Compressive Sensing, *Phys. Rev. X* **1**, 021021 (2011).
- [32] M. Raissi, Deep hidden physics models: Deep learning of nonlinear partial differential equations, *J. Mach. Learn. Res.* **19**, 932 (2018).
- [33] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach, *Phys. Rev. Lett.* **120**, 024102 (2018).
- [34] J. Pathak, A. Wilner, R. Fussell, S. Chandra, B. Hunt, M. Girvan, Z. Lu, and E. Ott, Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model, *Chaos* **28**, 041101 (2018).
- [35] A. Griffith, A. Pomerance, and D. J. Gauthier, Forecasting chaotic systems with very low connectivity reservoir computers, *Chaos* **29**, 123108 (2019).
- [36] T. Qin, K. Wu, and D. Xiu, Data driven governing equations approximation using deep neural networks, *J. Comput. Phys.* **395**, 620 (2019).
- [37] Z. Zhang, Y. Zhao, J. Liu, S. Wang, R. Tao, R. Xin, and J. Zhang, A general deep learning framework for network reconstruction and dynamics learning, *Appl. Netw. Sci.* **4**, 110 (2019).
- [38] H. Fan, J. Jiang, C. Zhang, X. Wang, and Y.-C. Lai, Long-term prediction of chaotic systems with machine learning, *Phys. Rev. Res.* **2**, 012080(R) (2020).
- [39] C. Zhang, J. Jiang, S.-X. Qu, and Y.-C. Lai, Predicting phase and sensing phase coherence in chaotic systems with machine learning, *Chaos* **30**, 083114 (2020).
- [40] C. Zang and F. Wang, Neural dynamics on complex networks, in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Association for Computing Machinery, New York, 2020), pp. 892–902.
- [41] J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, Teaching recurrent neural networks to infer global temporal structure from local examples, *Nat. Mach. Intell.* **3**, 316 (2021).
- [42] S. J. Mooney and V. Pejaver, Big data in public health: Terminology, machine learning, and privacy, *Annu. Rev. Public Health* **39**, 95 (2018).
- [43] K. Yu, L. Tan, X. Shang, J. Huang, G. Srivastava, and P. Chatterjee, Efficient and privacy-preserving medical research support platform against COVID-19: A blockchain-based approach, *IEEE Consumer Electron. Mag.* **10**, 111 (2020).
- [44] L. Lenert and B. Y. McSwain, Balancing health privacy, health information exchange, and research in the context of the COVID-19 pandemic, *J. Am. Med. Inform. Assoc.* **27**, 963 (2020).
- [45] M. J. Atallah and W. Du, Secure multi-party computational geometry, in *Workshop on Algorithms and Data Structures* (Springer, New York, 2001), pp. 165–179.
- [46] M.-L. Zhang and Z.-H. Zhou, A review on multi-label learning algorithms, *IEEE Trans. Knowl. Data Eng.* **26**, 1819 (2013).
- [47] J. Konečn'ý, H. B. McMahan, D. Ramage, and P. Richtárik, Federated optimization: Distributed machine learning for on-device intelligence, [arXiv:1610.02527](https://arxiv.org/abs/1610.02527).
- [48] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, Bayesian nonparametric federated learning of neural networks, in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research Vol. 97 (PMLR, Long Beach, CA, 2019), pp. 7252–7261.
- [49] Y. Liu, J. James, J. Kang, D. Niyato, and S. Zhang, Privacy-preserving traffic flow prediction: A federated learning approach, *IEEE Internet Things J.* **7**, 7751 (2020).
- [50] C. Wu, F. Wu, L. Lyu, T. Qi, Y. Huang, and X. Xie, A federated graph neural network framework for privacy-preserving personalization, *Nat. Commun.* **13**, 3091 (2022).
- [51] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, Robust and communication-efficient federated learning from non-iid data, *IEEE Trans. Neural Networks Learn. Syst.* **31**, 3400 (2019).
- [52] G. Mei, Z. Guo, S. Liu, and L. Pan, SGNN: A graph neural network based federated learning approach by hiding structure, in *2019 IEEE International Conference on Big Data (IEEE, Piscataway, NJ, 2019)*, pp. 2560–2568.
- [53] C. He, K. Balasubramanian, E. Ceyani, C. Yang, H. Xie, L. Sun, L. He, L. Yang, P. S. Yu, Y. Rong, P. Zhao, J. Huang, M. Annavaram, and S. Avestimehr, FedGraphNN: A federated learning system and benchmark for graph neural networks, [arXiv:2104.07145](https://arxiv.org/abs/2104.07145).

- [54] Q. Yang, Y. Liu, T. Chen, and Y. Tong, Federated machine learning: Concept and applications, *ACM Trans. Intell. Syst. Technol.* **10**, 12 (2019).
- [55] C. Chen, W. Hu, Z. Xu, and Z. Zheng, FedGL: Federated graph learning framework with global self-supervision, [arXiv:2105.03170](https://arxiv.org/abs/2105.03170).
- [56] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, Communication-efficient learning of deep networks from decentralized data, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (PMLR, Fort Lauderdale, FL, 2017), pp. 1273–1282.
- [57] A.-L. Barabási, Scale-free networks: A decade and beyond, *Science* **325**, 412 (2009).
- [58] M. E. Newman and D. J. Watts, Renormalization group analysis of the small-world network model, *Phys. Lett. A* **263**, 341 (1999).
- [59] J. Gómez-Gardeñes and Y. Moreno, From scale-free to Erdős-Rényi networks, *Phys. Rev. E* **73**, 056124 (2006).
- [60] M. E. Newman, Finding community structure in networks using the eigenvectors of matrices, *Phys. Rev. E* **74**, 036104 (2006).
- [61] D. J. Watts and S. H. Strogatz, Collective dynamics of small-world networks, *Nature (London)* **393**, 440 (1998).
- [62] R. 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(R) (2003).
- [63] J. Duch and A. Arenas, Community detection in complex networks using extremal optimization, *Phys. Rev. E* **72**, 027104 (2005).
- [64] A. Zeng and C.-J. Zhang, Ranking spreaders by decomposing complex networks, *Phys. Lett. A* **377**, 1031 (2013).
- [65] B. Shulgin, L. Stone, and Z. Agur, Pulse vaccination strategy in the SIR epidemic model, *Bull. Math. Biol.* **60**, 1123 (1998).
- [66] M. 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 (2015).
- [67] W. Chen, Y. Yuan, and L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in *2010 IEEE International Conference on Data Mining* (IEEE, Piscataway, NJ, 2010), pp. 88–97.
- [68] A. Kirman, Ants, rationality, and recruitment, *Q. J. Econ.* **108**, 137 (1993).
- [69] A. Ocone, L. Haghverdi, N. S. Mueller, and F. J. Theis, Reconstructing gene regulatory dynamics from high-dimensional single-cell snapshot data, *Bioinformatics* **31**, i89 (2015).
- [70] M. Scheffer, Z.-G. Huang, T. P. Seager, W. Lin, C. Grebogi, A. Hastings, and Y.-C. Lai, Predicting tipping points in mutualistic networks through dimension reduction, *Proc. Natl. Acad. Sci. USA* **115**, 639 (2018).
- [71] K. Kaneko, Spatiotemporal intermittency in coupled map lattices, *Prog. Theor. Phys.* **74**, 1033 (1985).
- [72] K. Kaneko and I. Tsuda, *Complex Systems: Chaos and Beyond, a Constructive Approach with Applications in Life Sciences* (Springer, Berlin, 2000).
- [73] The ILI data of USA, available at <https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html/>.
- [74] S. Pei, S. Kandula, W. Yang, and J. Shaman, Forecasting the spatial transmission of influenza in the United States, *Proc. Natl. Acad. Sci. USA* **115**, 2752 (2018).
- [75] The county-to-county commuting data of USA, available at <https://www.census.gov/data/tables/2015/demo/metro-micro/commuting-flows-2015.html/>.
- [76] J. Gao, B. Barzel, and A.-L. Barabási, Universal resilience patterns in complex networks, *Nature (London)* **530**, 307 (2016).
- [77] S. G. K. Patro and K. K. Sahu, Normalization: A preprocessing stage, [arXiv:1503.06462](https://arxiv.org/abs/1503.06462).