

TDNetGen: Empowering Complex Network Resilience Prediction with Generative Augmentation of Topology and Dynamics

Chang Liu

Department of Electronic Engineering,
BNRist, Tsinghua University
Beijing, China
lc23@mails.tsinghua.edu.cn

Yiwen Song

Shenzhen International Graduate School,
Tsinghua University
Shenzhen, Guangdong, China
songyw23@mails.tsinghua.edu.cn

Jingtao Ding*

Department of Electronic Engineering,
BNRist, Tsinghua University
Beijing, China
dingjt15@tsinghua.org.cn

Yong Li*

Department of Electronic Engineering,
BNRist, Tsinghua University
Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Predicting the resilience of complex networks, which represents the ability to retain fundamental functionality amidst external perturbations or internal failures, plays a critical role in understanding and improving real-world complex systems. Traditional theoretical approaches grounded in nonlinear dynamical systems rely on prior knowledge of network dynamics. On the other hand, data-driven approaches frequently encounter the challenge of insufficient labeled data, a predicament commonly observed in real-world scenarios. In this paper, we introduce a novel resilience prediction framework for complex networks, designed to tackle this issue through generative data augmentation of network topology and dynamics. The core idea is the strategic utilization of the inherent joint distribution present in unlabeled network data, facilitating the learning process of the resilience predictor by illuminating the relationship between network topology and dynamics. Experiment results on three network datasets demonstrate that our proposed framework TDNetGen can achieve high prediction accuracy up to 85%-95%. Furthermore, the framework still demonstrates a pronounced augmentation capability in extreme low-data regimes, thereby underscoring its utility and robustness in enhancing the prediction of network resilience. We have open-sourced our code in the following link, <https://github.com/tsinghua-b-lab/TDNetGen>.

CCS CONCEPTS

Computing methodologies; Knowledge representation and reasoning; Applied computing; Physics

KEYWORDS

Complex Network; Resilience Prediction; Diffusion models; Semi-supervised Learning; Data Augmentation

*Jingtao Ding and Yong Li are corresponding authors.

This work is licensed under a Creative Commons Attribution International 4.0 License.

KDD '24, August 25-29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0490-1/24/08
<https://doi.org/10.1145/3637528.3671934>

Figure 1: Resilience of complex networks. \bar{x}_i denotes the averaged nodal state of the network.

ACM Reference Format:

Chang Liu, Jingtao Ding, Yiwen Song, and Yong Li. 2024. TDNetGen: Empowering Complex Network Resilience Prediction with Generative Augmentation of Topology and Dynamics. Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25-29, 2024, Barcelona, Spain, ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3637528.3671934>

1 INTRODUCTION

Real-world complex systems across various domains, such as ecological [19], gene regulatory [8], and neurological networks [51, 52], are often described as complex networks composed of interconnected nodes with weighted links. A fundamental characteristic of these systems is their resilience [7, 38], that is, the ability to maintain functionality in the face of disruptions. From the perspective of dynamical systems, nodal state evolution of complex networks is driven by underlying nonlinear dynamics. Specifically, with the functionality of each node represented by its state value, a resilient network can recover from disruptions (on its nodes) and dynamically evolve into a stable phase where all nodes operate at a high level of activity (see Figure 1). Understanding and predicting this critical property of resilience in complex networks not only enhances our ability to analyze and intervene in natural and social systems [7, 42, 44, 59] but also offers valuable insights for the design of engineered infrastructures [53].

To predict network resilience, theories grounded in nonlinear dynamical systems have been developed [17, 26, 30, 60]. These frameworks strive to separate the influences of network structure and dynamics to derive analytical solutions for complex, high-dimensional systems [6, 16, 41]. However, theoretical approaches

often presuppose a detailed understanding of network evolution dynamics, which is usually not available in practical scenarios. In contrast, data-driven methods are capable of extracting both structural and dynamic information about networks directly from observational data [12, 32, 35, 36, 50], allowing for resilience predictions without the need for predefined knowledge. From this perspective, the task of predicting network resilience can be reinterpreted as a graph classification problem based on data of network structure and dynamics using machine learning techniques. Nonetheless, the crucial role of resilience in system functionality means that collecting extensive labeled datasets from real-world complex networks is both expensive and impractical. As a result, the majority of network observations remain unlabeled, possessing information on network topology and nodal state trajectories but lacking resilience labels.

In this paper, we focus on addressing the problem of predicting network resilience amidst a scarcity of labeled data, identifying two primary obstacles:

Firstly, designing models for resilience prediction is inherently complex due to the intricate interplay between network structure and dynamics. A network is considered resilient if it can consistently return to a state where all nodes are active following a prolonged period of self-evolution and neighborly interactions. However, while topological data is readily available, constructing a practical model requires the capability to make accurate predictions based on partial evolution trajectories collected from a short time window.

Secondly, enhancing prediction accuracy in the face of scarce labels involves leveraging the intrinsic information embedded in unlabeled data regarding network structure and dynamics. Existing methodologies mainly include pseudo-labeling [31, 45], exemplified by self-training [25], and self-supervised learning [21, 28, 48, 54]. Pseudo-labeling tends to underperform with high model uncertainty, and self-supervised learning often overlooks the critical interplay between structure and dynamics, treating state evolution trajectories merely as node attributes. The graph data augmentation method [18] emerges as a leading technique by utilizing unlabeled data distribution to generate diverse augmented samples for improved training. However, the challenge of comprehensively characterizing the distribution of both topology and dynamics in unlabeled data has yet to be tackled, especially with limited observations, such as a few labeled networks and incomplete evolution trajectories.

To fully resolve these challenges, we introduce a novel resilience prediction framework called TDNetGen, which utilizes generative augmentation of network topology and dynamics. The core of TDNetGen is a neural network-based predictor that integrates a graph convolutional network-based topology encoder together with a transformer-based trajectory encoder, capturing the complex relationship between network structure and dynamics. This predictor is further refined through training on an augmented dataset comprising resilient and non-resilient samples, networks with topology information and evolution trajectories.

TDNetGen leverages a generative data augmentation approach by 1) capturing the underlying joint distribution of topology and dynamics in unlabeled data, and 2) obtaining the corresponding conditional distribution for each class label through a classifier-guided approach [1]. To facilitate effective generative learning in the vast joint space of topology and dynamics, we decouple the generation

process into topology generation using a topology denoising diffusion module and dynamics simulation with a dynamics learning module. To ensure robust learning with limited observations, we incorporate a re-tuning step for the resilience predictor on generated trajectories, thereby improving its generalization ability on unseen data. To summarize, our main contributions are as follows.

We tackle the critical problem of predicting complex network resilience under label sparsity issue and provide a novel perspective of improving by data augmentation.

We design a generative augmentation framework that benefits resilience predictor learning of interplay between network topology and dynamics by exploiting the underlying joint distribution in unlabeled data.

Empirical results on three network datasets demonstrate the superiority of our TDNetGen over state-of-the-art baselines in terms of increasing network resilience prediction accuracy up to 85%-95%. Moreover, aided by a generative learning capability of both topology and dynamics, TDNetGen can provide robust augmentation in low-data regimes, maintaining 93% of performance even when dynamic information cannot be observed in unlabeled data.

2 PRELIMINARIES

2.1 Resilience Prediction

Network resilience articulates that a resilient system is characterized by its invariable convergence towards a desired, non-trivial stable equilibrium following perturbation [7]. Formally, given a complex network $G = (V, A)$, where $V = \{v_1, v_2, \dots, v_N\}$ represents its node set and A denotes the adjacency matrix. The state of node i can be represented as x_i , usually governed by the following non-linear ordinary differential equations (ODEs) as the nodal state dynamics:

$$\frac{dx_i}{dt} = F^i(x_i^0) + \sum_{j=1}^N A_{ij} G^j(x_i, x_j^0), \quad (1)$$

where $F^i(x_i^0)$ represents the self-dynamics of nodes and $G^j(x_i, x_j^0)$ denotes interaction dynamics. The complex network is considered resilient if it consistently converges to only the desired nodal state equilibrium as time approaches infinity, irrespective of any perturbation and varying initial conditions with the exception of its fixed points.

2.2 Problem Formulation

Considering the challenge of obtaining detailed knowledge of the underlying equations that govern nodal state dynamics in real-world scenarios, in this work, we advocate for a purely data-driven approach to predict network resilience. In the context of the resilience prediction task, our dataset comprises network samples from which we can extract both topology and the initial steps of M nodal state trajectories prior to reaching a steady state. Formally, for a network comprising N nodes, the topology is represented by an adjacency matrix $A \in \mathbb{R}^{N \times N}$, while the observed nodal state trajectories are denoted as $\mathcal{X} \in \mathbb{R}^{M \times N \times T}$. As demonstrated in Section 2.1, determining the resilience of a network precisely necessitates knowledge of its steady-state conditions, a requirement that is often prohibitive to meet due to the high observational costs (e.g., long-term species population growth). Consequently, only a limited

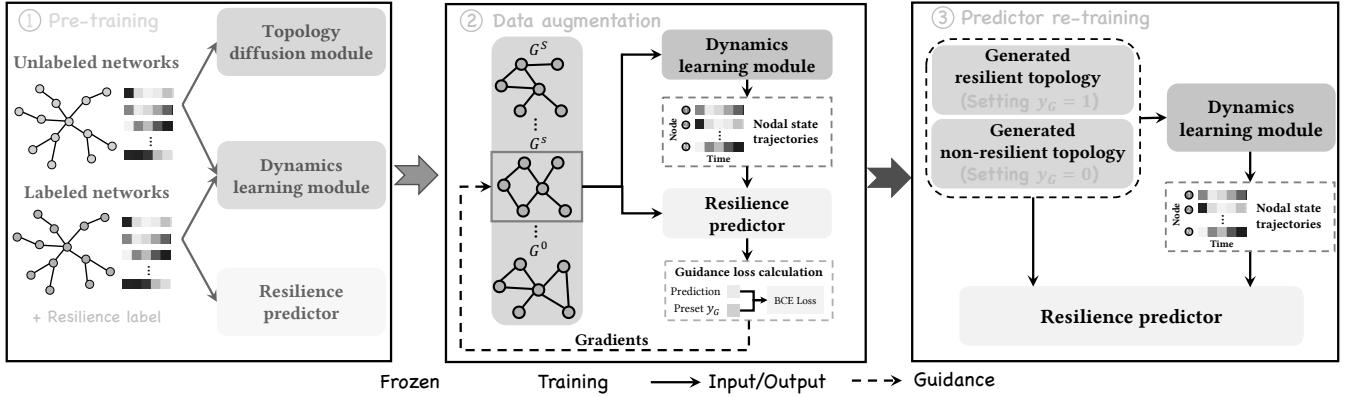


Figure 2: Overview of the proposed framework TDNetGen.

subset of network samples are labeled, denoted \mathcal{P} , with the majority remaining unlabeled, denoted \mathcal{Q} , where $|\mathcal{P}|$ is significantly smaller than $|\mathcal{Q}|$. The reliance on a narrow labeled dataset for training the resilience prediction model could result in sub-optimal performance due to the constrained sample size. In this work, we endeavor to leverage the untapped potential of the unlabeled data to enhance the training process of the resilience prediction model, with the objective of achieving superior predictive accuracy.

3 METHODOLOGY

3.1 Overview of Proposed Framework

In this section, we propose an effective method named TDNetGen to address the problem of complex network resilience prediction with limited labeled data samples via generative augmentation of topology and dynamics. Figure 2 illustrates the holistic design of TDNetGen, which consists of the following components:

Topology diffusion module. To facilitate resilience prediction performance and address the lack of labeled data, we design a diffusion module to model the distribution of unlabeled network topology. Therefore, we can sample new network topologies from the learned distribution.

Dynamics learning module. We propose a neural ODE[58]-based dynamics learning module to learn nodal state changes of networks from observed trajectories. It can simulate nodal state trajectories for the generated topologies from the topology diffusion module.

Resilience predictor. We design a resilience predictor empowered by Transformer and graph convolutional networks (GCNs), which jointly models nodal state dynamics and node interactions from observed trajectories and network topologies, respectively. It learns a low-dimensional embedding for each network and predicts its resilience based on this representation.

In our proposed framework, we first train both the dynamics learning module and the topology diffusion module utilizing unlabeled as well as labeled nodal state trajectories and network topologies, respectively, which is then followed by the pre-training of the resilience predictor using accessible labeled data. Subsequently, we generate new samples facilitated by the topology diffusion module and dynamics learning module, with the guidance provided by the resilience predictor. The newly generated samples further enhance

the training of the resilience predictor, thereby creating a synergistic feedback loop that significantly improves its predictive accuracy.

3.2 Topology Diffusion Module

Existing continuous graph diffusion models[27, 40] undermine the sparsity nature of topologies and usually result in complete graphs lacking physically meaningful edges. Consequently, they fail to capture the structural properties of complex networks. Therefore, we propose to model the distribution of network topologies using the discrete-space diffusion model[49], as illustrated in Figure 3. Different from diffusion models for images with continuous Gaussian noise, here we apply a discrete type of noise on each edge, and the type of each edge can transition to another during the diffusion process. Here, we define the transition probabilities of all edges at time steps as matrix Q^s , where $Q_{ij}^s = q^1 e^s = j j e^s - 1 = i^0$ denotes the type of edge transits to j from i at time steps. The forward process of adding noise of each time step to graph structure G is equivalent to sampling the edge type from the categorical distribution, formulated as:

$$q^1 G^s | G^0 = E^s - 1 Q^s, q^1 G^s | G^0 = E^s - 1 Q^s, \quad (2)$$

where $E \in \mathbb{R}^{N \times N \times 2}$ is the expanded adjacency matrix from its last dimension is a 2-D one-hot vector where e_{ij}^1 denotes an edge exists between the corresponding nodes, while e_{ij}^0 denotes there is no edge $Q^s = Q^1 \dots Q^s$. The reverse process aims to gradually recover the clean graph G given a noisy graph G^s . Towards this end, inspired by existing works[5, 49], we train a parameterized neural network h_θ which takes the noisy graph G^s as input and predicts the structure of the clean graph G , i.e., all the probability \hat{p}_{ij} of the existence of an edge e_{ij} between node i and j in the clean graph G . We use the cross-entropy loss to optimize parameters formulated as follows:

$$L_{BCE} = \frac{1}{N^2} \sum_{i,j} \text{CrossEntropy}(e_{ij}, \hat{p}_{ij}^0). \quad (3)$$

For the parameterization of h_θ , we employ the widely-recognized backbone of multi-layer graph transformers proposed by Dwivedi et al. [14]. Intuitively, node features are updated in each layer through the self-attention mechanism, and edge features are updated from

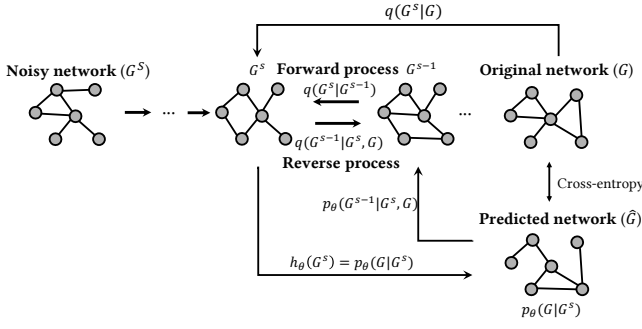


Figure 3: Illustration of topology diffusion module.

the information of its head and tail nodes. We describe the details of the parameterization network in Appendix A.1.

Once we train the neural network \mathbf{k}_θ , it can be applied to generate new network topologies. Specifically, the reverse process needs to estimate $p_\theta(G^{S-1} | G^S)$, which can be decomposed as follows,

$$p_\theta(G^{S-1} | G^S) = \prod_{i,j \in N} p_\theta(e_{ij}^s | G^S). \quad (4)$$

Each term in Equ. (4) can be formulated as,

$$p_\theta(e_{ij}^s | G^S) = \sum_{e \in E} p_\theta(e_{ij}^s | e) p_\theta(e | G^S), \quad (5)$$

where

$$p_\theta(e_{ij}^s | e, G^S) = \begin{cases} q(e_{ij}^s | e) & \text{if } q(e_{ij}^s | e) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

can be calculated with Bayesian rule. After sampling for preset steps, we can generate new network topologies which follow the distribution of the training dataset.

3.3 Dynamics Learning Module

Through the topology diffusion module, we can generate new network topologies for the training of resilience predictor. Nonetheless, we also need to obtain their nodal states trajectories to predict their resilience. As illustrated in Section 2.1, nodal state dynamics in complex networks usually have the generalized form of an ordinary differential equation (ODE) as:

$$\frac{d\mathbf{x}^1 t^0}{dt} = f^1(\mathbf{x}, G, W, t^0), \quad (7)$$

where $\mathbf{x}^1 t^0 \in \mathbb{R}^N$ represents nodal states of N -nodes network G at time step t^0 , f^1 denotes the dynamics function, and W denotes all dynamics parameters. Therefore, we develop a dynamics learning module designed to infer changes in nodal states solely from data, which learns nodal state dynamics in the expressive hidden space based on neural-ODE [58]. Given the initial state $\mathbf{x}^1 t^0$ of all network nodes, for each time step, the process initiates by mapping the state of the nodes to a latent space through an encoder f_e . Subsequently, graph neural networks (GNNs) are utilized as a parameterization technique to facilitate the learning of dynamics within this latent space. The transition from latent space representation back to the nodal state at each time step is accomplished by employing a decoder function f_d , which decodes the hidden space

embeddings to reconstruct the nodal states. The procedure can be represented as:

$$\mathbf{x}_h^1 t^0 = f_e^1(\mathbf{x}^1 t^0), \quad (8)$$

$$\frac{d\mathbf{x}_h^1 t^0}{dt} = \text{GNN}^1(\mathbf{x}_h^1 t^0), \quad (9)$$

$$\hat{\mathbf{x}}^1 t, \delta^0 = \mathbf{x}^1 t^0, \int_t^{t+\delta} f_d^1 \frac{d\mathbf{x}_h^1 t^0}{dt} dt, \quad (10)$$

where GNN can be implemented as an arbitrary design type of graph neural network layers. In our works, without the loss of generality, we choose to implement both encoder and decoder f_e and f_d functions using MLPs. Furthermore, GNN is instantiated through graph convolutional networks [29], thereby leveraging their robust capabilities in capturing and processing the inherent topological features of graphs. We use \mathcal{L}_1 loss to train the dynamics learning module, formulated as follows:

$$\mathcal{L}_1 = \frac{1}{|P| \cdot |Q|} \sum_{i=1}^{|P|} \int_0^T |\mathbf{x}_i^1 t^0 - \hat{\mathbf{x}}_i^1 t^0|. \quad (11)$$

As shown in Equ. (11), we train the dynamics learning module on both labeled data P and unlabeled data Q to achieve a better performance. It is noteworthy that in Section 4.2, we demonstrate the dynamics learning module can also perform well even when the nodal states of unlabeled data are inaccessible.

3.4 Resilience Predictor

We design a resilience predictor to jointly model the dynamics and topology of networks, which leverages stacked Transformer [47] encoder layers and graph convolutional layers [29] to encode the temporal correlations of nodal states and learn spatial interactions within network topology, respectively. We illustrate its architecture in Figure 4. Specifically, for a network with N nodes, we denote the nodal states with observed steps and trajectories of node u as $\mathbf{x}_u \in \mathbb{R}^{d \times M}$. For the k -th trajectory $\mathbf{x}_{u,k} \in \mathbb{R}^d$, we first input its states of each time step to a feed-forward layer, and further encode the temporal correlation between time steps with Transformer encoder layers, formulated as follows,

$$\mathbf{z}_{u,k} = \text{TransformerEncoder}(\mathbf{x}_{u,k}) W_1 + \mathbf{b}_1^0, \quad (12)$$

where $W_1 \in \mathbb{R}^{1 \times d_e}$ and $\mathbf{b}_1^0 \in \mathbb{R}^{d_e}$ are trainable parameters. After that, we integrate the embedding of the terminal time step of all nodes in the network, denoted $\mathbf{z}_k \in \mathbb{R}^{N \times d_e}$, as their k -th trajectory embeddings.

To capture the interactions of nodes within the topology, we design a graph convolutional network (GCN) empowered by multi-layer message-passing. Given the adjacency matrix of network topology A , we first calculate the Laplacian operator $L = D_{in} - A$ and D_{out} , where the diagonal D_{in} and D_{out} represent the in- and out-degree of nodes. We input the k -th trajectory embeddings of nodes to the graph convolutional network. The l -th layer message passing of the designed GCN can be represented as follows:

$$\mathbf{z}_k^{l+1} = f^1(\mathbf{z}_k^l), \quad (13)$$

where f^1 and g^1 are implemented as MLPs. Such message-passing design is motivated by Equ. (1), aiming to more precisely model the effects from both the node itself and its neighborhood on a specific node.

It is noteworthy that during the aforementioned procedure, different trajectories are processed in parallel. We further introduce a trajectory attention module to integrate the information from different trajectories for network-level representation. Specifically, we treat node embedding matrix of different trajectories after layers' message passing as a combination of feature maps, and denote the results after mean and max pooling $Z_{avg}^{1L_0} \in \mathbb{R}^{M \times d_e}$ and $Z_{max}^{1L_0} \in \mathbb{R}^{M \times d_e}$, respectively. After that, we feed them into a shared MLP, add and activate the outputs to compute attention weights of trajectories, formulated as:

$$\alpha = \sigma^1 \text{MLP}^1 Z_{avg}^{1L_0}, \text{MLP}^1 Z_{max}^{1L_0}, \quad (14)$$

where $\alpha \in \mathbb{R}^M$, and σ^1 denotes the sigmoid activation function. Therefore, the fused node embedding matrix can be derived from:

$$Z = \sum_{k=1}^M \alpha_k \cdot Z_k^{1L_0}, \quad (15)$$

where $Z \in \mathbb{R}^{N \times d_e}$, α_k is the attention weight for the k -th trajectory, and \cdot denotes Hadamard product. We use a readout function to derive the embedding of the entire network, $e_{net} = \text{Readout} Z^0$. Here, we implement the readout function as mean pooling between nodes. We then predict the resilience of the network using e_{net} as follows:

$$\hat{y} = \text{MLP}^1 e_{net}^0. \quad (16)$$

Then we can train the resilience predictor with binary cross-entropy (BCE) loss,

$$L_{BCE} = \sum_{i=1}^{|\mathcal{N}|} y_i \log^1 \hat{y}_i^0 + (1 - y_i) \log^1 (1 - \hat{y}_i^0), \quad (17)$$

where y_i and \hat{y}_i denote the ground truth and the prediction result of the i -th network. $|\mathcal{N}|$ is the number of networks used for training. However, its predictive performance typically falls below the optimal level, primarily attributed to the scarcity of data.

It is noteworthy that after training resilience predictor on labeled data, we further re-tune the predictor on identical topologies wherein the nodal state trajectories are generated through the neural-ODE of the dynamics learning module. It enables the resilience predictor to accurately accommodate the minor discrepancies observed between the ground-truth trajectories and those generated through simulation, thereby ensuring the robust predictive performance.

3.5 Joint Data Augmentation of Topology and Dynamics

The above modules enable us to generate network samples with both topology and nodal state trajectories. However, it is important to note that the simulated nodal states are confined to the initial temporal period, corresponding to the maximal duration present within the training dataset, and compelling the dynamics learning module to simulate time steps beyond its training scope yields results of questionable reliability. Consequently, the principal challenge arises from the inability to ascertain the steady-state conditions of the generated networks. This limitation obstructs the direct acquisition of resilience labels, presenting a significant impediment to the data augmentation. To overcome this problem, we advocate for the strategy of guiding the topology diffusion module, enabling it to

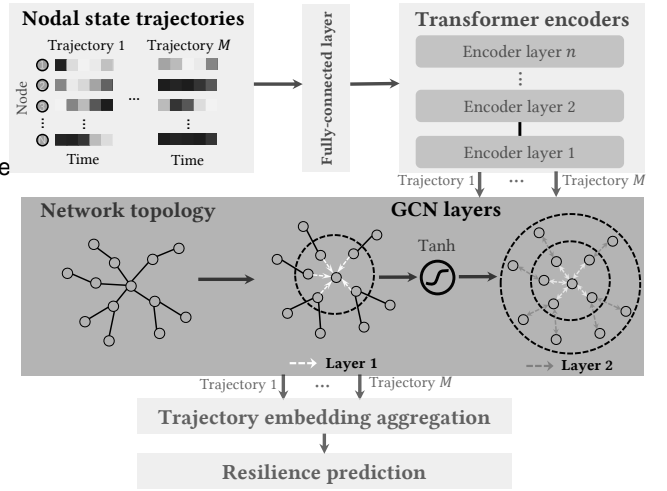


Figure 4: Architecture of the resilience predictor. generate networks with predefined resilience characteristics. More precisely, we integrate classifier guidance [11] into the topology diffusion model, which leverages signals derived from the resilience predictor trained on the labeled dataset. The conceptual basis of the guidance mechanism involves that the resilience predictor provides the resilience condition of the clean samples from the intermediate samples generated by the diffusion model, which in turn, steers the generation process towards exhibiting desired resilience characteristics. To formally define the guided diffusion process, we provide the following lemma from [11]:

Lemma 3.1. Denote the forward process conditioned on \hat{q} , and the unconditional forward process as G . Given the reasonable assumption $\hat{q}^1 G^s j_G, y_G^0 = q^1 G^s j_G^0$, we have

$$\hat{q}^1 G^s j_G^s, y_G^0 / q^1 G^s j_G^{s_0} \hat{q}^1 y_G j_G^s \approx 1^0 \quad (18)$$

An direct estimation of $\hat{q}^1 G^s j_G^{s_0} \hat{q}^1 y_G j_G^s \approx 1^0$ is to use

$$p_\theta^1 G^s j_G^{s_0} p_\eta^1 y_G j_G^s \approx 1^0, \quad (19)$$

where p_η are parameterized by the resilience predictor. However, we cannot evaluate all possible values of \hat{q} . A viable method is to treat G as a continuous tensor of order 2, and use the first-order approximation from Taylor expansion [49], as

$$\log \hat{q}^1 y_G j_G^s \approx \log \hat{q}^1 y_G j_G^{s_0} + \langle r_G \log \hat{q}(y_G j_G^s), G^s \rangle \quad (20)$$

$$\sum_{i,j} \left(r_{e_{ij}} \log \hat{q}(y_G j_G^s), e_{ij}^s \right), C^1 G^{s_0}, \quad (21)$$

where $C^1 G^{s_0}$ is a function that only relates to G^s . Assume that $\hat{q}^1 y_G j_G^{s_0} \sim \text{Bernoulli}(f_\eta^1 G^s)$, where f_η is the resilience predictor, we have

$$r_G \log \hat{q}(y_G j_G^s) / r_G \approx L_{BCE}^1 \hat{y}, y_G^0 \quad (22)$$

Drawing upon the aforementioned theoretical framework, at the steps of the reverse process, we first employ the resilience predictor f_η to predict y_G , i.e., $\hat{y}_G = f_\eta^1 G^{s_0}$, and estimate the $\hat{q}^1 y_G j_G^{s_0}$ as

$$p_\eta^1 y_G j_G^s \approx \exp^{-\lambda r_G L_{BCE}^1 \hat{y}_G, y_G^0}, G^s \approx 1^0, \quad (23)$$

where λ represents the guidance intensity. Hence, we can sample G^{s-1} from

$$G^{s-1} \sim p_{\theta} \circ G^{s-1} \circ j_{G^{s-1}} \circ p_{\eta} \circ y_G \circ j_G \circ G^{s-1} \circ \lambda, \quad (24)$$

where $p_{\theta} \circ G^{s-1} \circ j_{G^{s-1}}$ can be calculate from Equ. (4)-(6).

Consequently, by setting $y_G = 1$ and $y_G = 0$, we can generate novel labeled network topologies guided by the resilience predictor. These topologies subsequently serve as inputs to simulate their respective nodal state trajectories via the dynamics learning module. This approach facilitates the augmentation of our datasets with additional fully labeled data, which, in turn, allows for the re-training of the resilience predictor. Such a method is anticipated to significantly enhance the predictive accuracy of the resilience predictor, ensuring a more reliable assessment of network resilience under conditions of data sparsity.

3.6 Time Complexity Analysis

We define N as the number of nodes in a graph, and analyze time complexity of each module in our framework as follows.

Topology disunion module is parameterized using Graph-Transformer layers (Appendix A.1). It exhibits a time complexity of O^1N^{20} per layer, attributable to the computation of attention scores and the prediction process for each edge.

Dynamics learning module is based on neural-ODE and parameterized through GCN layers. This module also demonstrates a time complexity of O^1N^{20} resulting from convolution operations and the application of a fourth-order Runge-Kutta ODE solver.

Resilience predictor leverages stacked Transformer encoder layers to capture temporal correlations among nodal states, while spatial interactions within the network topology are discerned through GCN layers. Time complexities of the Transformer encoder layers and GCN layers are O^1NT^{20} and O^1N^{20} , respectively, with T representing the trajectory length. Typically, T is significantly smaller than N for most graph structures.

Consequently, the overall time complexity of TDNetGen is dominantly O^1N^{20} , signifying its scalability and efficiency in processing large graph structures. In practical experiments, our framework takes about 10 seconds to generate a 200-nodes graph with nodal state trajectories and 20 milliseconds to predict its resilience. Since resilience inference is not a real-time task, such time complexity is acceptable for application.

4 EXPERIMENTS

In this section, we demonstrate the superior performance of our framework TDNetGen, aiming to answer the following research questions:

RQ1: How does our framework TDNetGen compare to potential baseline methods of harnessing unlabeled data to enhance predictive performance?

RQ2: How do different designs of TDNetGen affect the model performance?

RQ3: How does TDNetGen perform across limited numbers of original labeled samples and lengths of nodal state trajectories?

RQ4: How does TDNetGen perform with different network types and scales?

4.1 Experimental Settings

Table 1: Statistics of network datasets.

	Mutualistic	Regulatory	Neuronal
#Unlabeled networks	1900	1900	1900
#Labeled networks	100	100	100
Average #nodes	36	44	45
Average #edges	99	115	112

4.1.1 Dataset. To construct the dataset, we synthesize complex networks with three nodal state dynamics from physics and life sciences. Denote x_i^t as the state of node i at time step t , the dynamics are as follows,

Mutualistic dynamics. The mutualistic dynamics [9] $\frac{dx_i}{dt} = B_i x_i \left(1 - \frac{x_i}{K}\right) \left(\frac{x_i}{C} + 1\right) + \sum_{j=1}^N A_{ij} \frac{x_i x_j}{D_i + E x_i + H x_j}$ describes the alterations in species populations that are engendered by the migration term B_i , logistic growth term with environment capacity K [57], Allee effect [2] term with threshold C , and mutualistic interaction between species with interaction network A .

Regulatory dynamics. The regulatory dynamics, also called Michaelis-Menten dynamics [3], is described by $\frac{dx_i}{dt} = B x_i^f - \sum_{j=1}^N A_{ij} \frac{x_j^h}{x_j^h + 1} \cdot f$. f represents the degradation ($f = 1$) or dimerization ($f = 2$). Additionally, the second term in the equation is designed to capture genetic activation with Hill coefficient h , which serves to quantify the extent of gene regulation collaboration.

Neuronal dynamics. The neuronal dynamics, also called Wilson-Cowan dynamics [51, 52], is described by the equation $\frac{dx_i}{dt} = x_i \left(\sum_{j=1}^N A_{ij} \frac{1}{1 + e^{\mu \delta x_j}} \right)$. For each node in the network, it receives cumulative inputs from its neighbors. The second term of the equation represents the activation signal that is collectively contributed by all neighboring nodes.

For each dynamics, we synthesize Erdős-Rényi networks with edge creation probability uniformly sampled in $[0, 0.15]$ and use the fourth-order Runge-Kutta stepper [4] to simulate their nodal state trajectories. For more details, please refer to the Appendix A.2. We create 2000 network samples for training, 200 for validation, and another 200 samples for testing. In the training stage, we randomly select 100 (5%) samples as labeled data and keep the other 1900 samples as unlabeled. The statistics of datasets are shown in Table 1.

4.1.2 Baselines and metrics. In the following parts, we define the model trained only on original labeled data as the vanilla model. Besides this, there are mainly three kinds of baseline methods designed to leverage unlabeled data for enhancing the predictive performance of the vanilla predictor.

Self-training methods. They utilize the predictor to assign pseudo labels to unlabeled data, thereby augmenting the labeled training dataset. We abbreviate these method as ST.

Self-supervised learning methods. They employ hand-crafted tasks to derive insights from unlabeled data, thereby facilitating the pre-training of model parameters. Subsequently, they undergo further supervised training on the labeled dataset. This approach is predicated on the premise that integrating pre-training phases with subsequent supervised learning phases leverages both unlabeled and labeled datasets, thereby enhancing the model's learning efficacy and predictive accuracy. Such methods include

Table 2: Overall predictive performance of models. w/o trajectories represents that nodal state trajectories of unlabeled data are unknown. The performance of TDNetGen is marked in bold, and the best baseline is underlined.

# Training samples		Mutualistic 100		Regulatory 100		Neuronal 100	
Model		F1	ACC	F1	ACC	F1	ACC
Vanilla model		0.838	0.848	0.806	0.780	0.775	0.784
Self-training		0.807	0.827	0.780	0.735	0.728	0.764
ST							
EdgePred [21]		0.840	0.851	0.813	0.791	0.776	0.784
AttrMask [21]		0.831	0.845	0.817	0.793	0.770	0.779
Self-supervised learning							
ContextPred [21]		0.843	0.847	0.815	0.789	0.772	0.781
InfoMax [48]		0.829	0.815	0.875	0.870	0.787	0.805
GraphLog [54]		0.808	0.796	0.796	0.769	0.772	0.732
D-SLA [28]		0.810	0.799	0.855	0.840	0.780	0.805
Graph data augmentation							
TRY [17]		<u>0.891</u>	<u>0.886</u>	<u>0.896</u>	<u>0.898</u>	<u>0.818</u>	<u>0.833</u>
G-Mixup [18]		0.875	0.888	0.900	0.899	0.786	0.812
TDNetGen (w/o trajectories)		0.913	0.913	0.922	0.923	0.805	0.810
TDNetGen		0.929	0.934	0.944	0.946	0.845	0.873
Improvement		4.26%	5.18%	4.89%	5.23%	3.30%	4.80%

EdgePred, AttrMask, ContextPred [21], InfoMax [48], GraphLog [54], and D-SLA [28].

Graph data augmentation (GDA) methods. They incorporate new graphs with labels to train the model, including theory-guided method (TRY [17], detailed in Appendix A.3) and G-Mixup [18].

For both self-training and graph data augmentation methods, the quantity of newly generated samples is the same as that produced by our method. Similarly, within the realm of self-supervised learning, we also select the same count of unlabeled samples as the volume of new samples generated by our framework. We use F1-score (F1) and Accuracy (ACC) to evaluate the predictive performance of the resilience predictor.

4.1.3 Implementation details. We implement our model in PyTorch and complete all training and test tasks on a single NVIDIA RTX 4090 GPU. With our framework, we generate 100 new networks assigning 500 resilient and 500 non-resilient networks. Subsequently, we randomly select half of these networks to serve as the augmented data. We set the guidance intensity = 2000. In our study, model parameters are optimized using the Adam optimizer, coupled with Xavier initialization, which ensures a robust starting point for learning. For each experiment, we conduct a minimum of at least 5 times employing distinct random seeds and report the average value.

4.2 Overall Performance (RQ1)

We report the performance of our framework with mean value and standard deviation in Table 2. From the experimental results, we have the following conclusions:

Our framework effectively empowers predictive performance via generative augmentation of both topology and dynamics. The results demonstrate that with the help of our proposed data augmentation framework, the predictive performance of the resilience predictor can be effectively improved. For example, on mutualistic dataset, the F1-score of the resilience predictor previously trained on 100 labeled data increases from 0.838 to 0.929 (+10.86%), and its ACC increases from 0.848 to

0.934 (+10.14%) after training on the augmented data. Moreover, our framework also improves the best baseline among all self-training, self-supervised learning, and GDA methods. F1-score by 4.26%, 4.89%, 3.30%, and w.r.t. ACC by 5.18%, 5.23%, 4.80%, on mutualistic, regulatory and neuronal dataset, respectively. All these results demonstrate the outstanding performance of our proposed framework. We find that the best baseline methods on three datasets belong to the category of graph data augmentation. Compared with TRY and G-mixup, we achieve to jointly model network topology and dynamics in a fine-grained manner. Robustness performance without nodal state trajectories of unlabeled data. In certain contexts, the requirement to obtain the nodal states, even only for an initial phase of evolution, still proves to be difficult or costly. Consequently, we analyze scenarios where the nodal state trajectories of unlabeled data are inaccessible and we can only train the dynamics learning module on those of limited labeled data in Equ. (11). Results in Table 2 demonstrate that our framework is capable of sustaining commendable performance even under such constrained conditions and surpassing the best baseline in most scenarios. It underscores the versatility of our framework and its potential effectiveness under more limited data availability of the real world scenarios. Self-training cannot universally guarantee a positive impact on model performance. The results demonstrate that self-training methods have a relatively small positive effect on predictive performance among all three datasets compared to our framework TDNetGen. For example, on regulatory datasets, the F1-score of the resilience predictor increases 0.051, and its ACC increases 0.079 compared to the vanilla model. This is because the labels assigned to the augmented data in the self-training process originate from the model itself with sub-optimal predictive performance. This approach inherently carries the risk of generating labels that are incongruent with the ground truth and partially introduce contradictory information into the training dataset. The presence of such inaccurately labeled data can confound the learning algorithm, leading to a deterioration in the model's capacity to make accurate predictions.

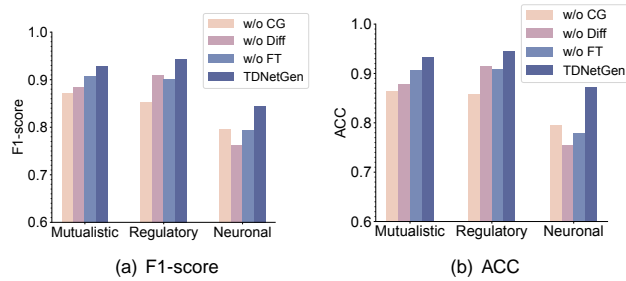


Figure 5: Ablation studies on datasets. CG: Classifier guidance, Di: Diffusion module, FT: Fine-tuning on dynamics learning module-produced trajectories.

Extracting knowledge from unlabeled data via hand-crafted self-supervised tasks offers marginal benefits to the resilience prediction. We also find that models trained on self-supervised tasks can only extract limited knowledge from unlabeled data to benefit the resilience prediction task. From the results, the improvement to vanilla model from competitive self-supervised learning methods (ContextProt and Infomax [43]) is still relatively marginal compared to our framework (4.86%, +17.12% and +9.03%, on mutualistic, regulatory, neuronal dataset, respectively). The primary reason for the observed discrepancy lies in the substantial divergence between conventional hand-crafted tasks, which only focus on the modeling of topological structures. In the resilience prediction task, however, we also need to consider nodal state dynamics of networks.

4.3 Ablation Study (RQ2)

To provide a comprehensive analysis and assess the effect of our designed components quantitatively, we conduct several ablation experiments via removing each design element, and present the evaluation results in Figure 5.

Effectiveness of Classifier guidance. We first remove the design of classifier guidance and generate new network topologies via only unconditional topology diffusion module. The nodal state trajectories are simulated utilizing the dynamics learning module, and its resilience label is determined by the resilience predictor that has been trained on the labeled dataset. The results reveal that the F1-score of the ablation model significantly declines 6.14%, 9.75%, and 5.68% compared to the full model design, which underscores the importance of guided generation.

Effectiveness of resilience predictor fine-tuning on dynamics learning module-produced trajectories. In this experiment, we remove the fine-tuning procedure of the resilience predictor on dynamics learning module-produced trajectories, instead utilizing the one trained with ground-truth trajectories. The results illustrate that fine-tuning could significantly enhance its guidance capabilities to generate higher-quality data, ultimately empowering the resilience predictor to be re-trained on it.

Architecture analysis. We compare our diffusion-based topology generation module with generative adversarial network (GAN) module. Specifically, we replace topology generation module as a GAN-based module proposed in [37]. We use the topologies of unlabeled data to train the GAN model, and sample new topologies from it. The nodal state trajectories and the resilience label are produced by our dynamics learning module and the

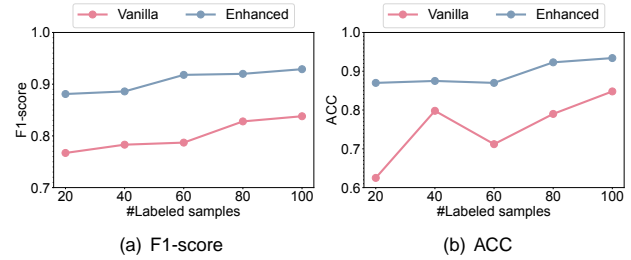


Figure 6: Model performance with less labeled samples on mutualistic dataset.

resilience predictor, respectively. Experiments demonstrate that our original design of diffusion models exhibit superior generative performance compared to GANs, which underscores the efficacy of diffusion models in capturing the underlying topology data distribution, thereby facilitating more accurate and reliable topology generation.

4.4 Augmentation with Limited Labels and Observations (RQ3)

In this section, we investigate data augmentation capabilities of our proposed framework under conditions of more limited number of labeled samples and reduced observed trajectory lengths, representing more challenging scenarios. We illustrate the results on the mutualistic dataset in Figure 6-7.

Less labeled samples. We investigate the performance of the vanilla model, where the numbers of labeled networks are in {20, 40, 60, 80, 100}, as well as the enhanced model trained on the augmented data generated by TDNetGen. From the results, we find that the predictive performance of the vanilla model is generally proportional to the number of labeled data used for training. TDNetGen is robust to the limitation of labeled data, which can still generate reasonable samples to benefit the predictive performance of the vanilla model. These findings underscore the versatility and potential of our proposed framework, particularly in scenarios characterized by a scarcity of labeled data, which constitutes a small portion of the available dataset.

Shorter nodal state trajectories. We also investigate the performance of the vanilla model and TDNetGen while using shorter nodal state trajectories, which contain 3, 4, 5, 6 time steps. We discover that the performance of the vanilla model improves with the increase in trajectory length since the model can extract more knowledge about nodal state dynamics from data to make more accurate resilience predictions. In this scenario, TDNetGen can also help to augment the model's performance, which suggests that even in situations where nodal state trajectories are costly to acquire, our framework remains applicable and effective for data augmentation purposes of simultaneously generating plausible topologies and nodal state trajectories of complex networks.

4.5 Robustness against Different Network Types and Scales (RQ4)

We consider other network models, including Barabási-Albert model [1], $S^1 \times \mathbb{H}^2$ model [43], and stochastic block model (SBM) [20], which have more complex and heterogeneous structural properties. Moreover, we also evaluate the scalability of our framework

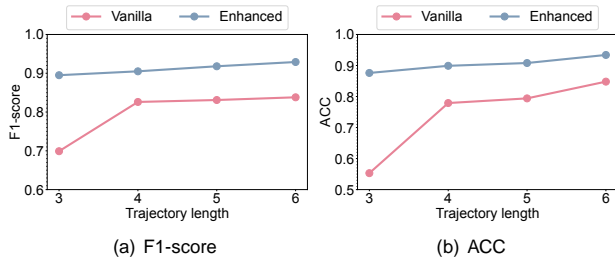


Figure 7: Model performance with shorter nodal state trajectories on the mutualistic dataset.

Table 3: Overall predictive performance of models on BA, S^1 , SBM, and brain networks.

# Training samples	BA		S^1		SBM		Brain		
	F1	ACC	F1	ACC	F1	ACC	F1	ACC	
Vanilla model	0.814	0.798	0.776	0.828	0.767	0.841	0.792	0.827	
Self-training	0.767	0.754	0.774	0.787	0.778	0.802	0.805	0.796	
EdgePred [21]	0.797	0.780	0.747	0.806	0.784	0.733	0.725	0.685	
AttrMask [21]	0.788	0.776	0.750	0.805	0.755	0.760	0.733	0.741	
Self-supervised learning	0.792	0.790	0.771	0.819	0.754	0.758	0.727	0.722	
InfoMax [48]	0.776	0.765	0.784	0.820	0.812	0.833	0.743	0.775	
GraphLog [54]	0.783	0.713	0.780	0.816	0.759	0.764	0.745	0.757	
D-SLA [28]	0.817	0.823	0.790	0.813	0.825	0.820	0.772	0.765	
Graph data augmentation	0.837	0.840	0.791	0.796	0.855	0.858	0.826	0.831	
TRY [17]	0.834	0.837	0.807	0.811	0.852	0.851	0.839	0.844	
G-Mixup [18]	0.834	0.837	0.807	0.811	0.852	0.851	0.839	0.844	
TDNetGen (w/o trajectories)	0.842	0.846	0.823	0.830	0.886	0.890	0.873	0.870	
TDNetGen	0.870	0.850	0.856	0.875	0.935	0.937	0.914	0.907	
Improvement	3.99%	1.19%	6.07%	6.71%	9.36%	9.21%	8.93%	7.46%	

on large-scale empirical brain networks with 998 nodes in maximum. For each dataset, we obtain nodal states of networks via neuronal dynamics, and other experimental settings are the same as Section 4.1. The details of dataset construction are shown in Appendix A.2. We demonstrate the results in Table 3, which indicates that our framework can still achieve the best augmentation performance on more broad types and scales of networks with complex structural properties and different network sizes.

5 RELATED WORKS

5.1 Resilience Prediction of Complex Networks

Existing works on resilience prediction are mainly categorized to analytical estimations from physical theories [7, 30, 39]. Gao et al. [17] propose to reduce the dimension of complex networks to single-parameter systems based on mean-field theory, thus we can easily analyze the equilibrium of 1-D ODE problem and predict the resilience of complex networks. Laurence et al. [10] perform dimension reduction based on spectral graph theory on the dominant eigenvalues and eigenvectors of adjacency matrices. Morone et al. [39] develop a resilience prediction methodology by quantifying the k-core structure within networks. Despite their effectiveness, they often pre-suppose a detailed understanding of nodal state dynamics, which is usually not available in practical scenarios. In our work, we design data-driven methods that extract topology and nodal state dynamics information from observational data, allowing for resilience predictions without the need for prior knowledge.

5.2 Diffusion Models on Graphs

Diffusion probabilistic models have been widely used in text, image, audio generation, etc. [34, 55, 56]. Recently, some existing

works have applied the diffusion model to the field of graph generation [9, 22, 46, 49]. Huang et al. [22] define a stochastic differential equation (SDE) that smoothly converts graphs with complex distribution to random graphs, and samples new graphs by solving the reverse-time SDE. Tseng et al. [46] propose GraphGUIDE to achieve interpretable and controllable graph generation, wherein edges in graph are clipped or set at each discrete time step. Chen et al. [9] propose to leverage graph sparsity during each step of diffusion process, which only focuses on a small portion of nodes and considers edge changes between them. In contrast to existing contributions focused primarily on graph structures, our research extends to the generation of complex networks, which encompasses not merely the graph topology but also integrates nodal state trajectories, thereby facilitating the generation of comprehensive network data.

5.3 Learning from Unlabeled Data

Typical approaches of learning from unlabeled data for graph classification include pre-training on self-supervised tasks [3, 54], self-training [4, 23, 25, 45], and graph data augmentation [8]. Although pre-training proves to be effective for vision and language-related tasks, it can hardly help the resilience prediction task because of the disparity between hand-crafted and downstream prediction tasks [24, 28]. Therefore, we still lack a universal self-supervised task that learns from unlabeled graphs and improves the performance of downstream scenarios. Self-training tasks assign pseudo-labels to unlabeled data by leveraging the model itself, followed by the retraining of the model with pseudo-labeled data. Existing works [4, 23, 45] focus on uncertainty estimation of assigned labels to minimize the impact of noisy pseudo-labels. Furthermore, Liu et al. [33] learn data distributions from unlabeled graphs with diffusion models, and to generate task-specific labeled graphs for data augmentation. Compared with their work, our proposed TDNetGen framework considers more intricate scenarios of complex networks with interplay between topology and nodal state dynamics. Our framework can extract knowledge from full unlabeled complex network samples, thereby generating high-quality augmented data that benefits the training of prediction models.

6 CONCLUSIONS

In this work, we propose an effective framework, TDNetGen, for complex network resilience prediction. It not only addresses the problem in a data-driven manner without prior knowledge about ground-truth dynamics, but also solves labeled data sparsity problem with the generative augmentation of jointly modeling network topology and dynamics. Extensive experiments demonstrate the superiority of TDNetGen and also highlight its robustness within less labeled data and dynamics information conditions. The methodology introduced in this paper provides a novel perspective for improving resilience prediction through data augmentation, that is, leveraging the untapped potential of unlabeled data to enhance the learning process.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China under U23B2030, 62272260, U21B2036.

REFERENCES

- [1] Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74, 1 (2002), 47.
- [2] Warder Clyde Allee, Orlando Park, Alfred E Emerson, Thomas Park, Karl P Schmidt, et al. 1949. *Principles of animal ecology*. Number Edn 1. WB Saundere Co. Ltd.
- [3] Uri Alon. 2019. *An introduction to systems biology: design principles of biological circuits*. CRC press.
- [4] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. 2020. Deep evidential regression. *Advances in Neural Information Processing Systems* 33 (2020), 14927–14937.
- [5] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems* 34 (2021), 17981–17993.
- [6] Sai Balaji, M Madan Babu, Lakshminarayan M Iyer, Nicholas M Luscombe, and Lakshminarayan Aravind. 2006. Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast. *Journal of molecular biology* 360, 1 (2006), 213–227.
- [7] Ed Bullmore and Olaf Sporns. 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience* 12, 3 (2009), 186–198.
- [8] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. 2018. Neural ordinary differential equations. *Advances in neural information processing systems* 31 (2018).
- [9] Xiaohui Chen, Jiaxin He, Xu Han, and Li-Ping Liu. 2023. Efficient and Degree-Guided Graph Generation via Discrete Diffusion Modeling. *arXiv preprint arXiv:2305.04112* (2023).
- [10] Zhengdao Chen, Lei Chen, Soledad Villar, and Joan Bruna. 2020. Can graph neural networks count substructures? *Advances in neural information processing systems* 33 (2020), 10383–10395.
- [11] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* 34 (2021), 8780–8794.
- [12] Jingtao Ding, Chang Liu, Yu Zheng, Yunke Zhang, Zihan Yu, Ruikun Li, Hongyi Chen, Jinghua Piao, Huandong Wang, Jiazhen Liu, et al. 2024. Artificial Intelligence for Complex Network: Potential, Methodology and Applications. *arXiv preprint arXiv:2402.16847* (2024).
- [13] John R Dormand and Peter J Prince. 1980. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics* 6 (1980), 19–26.
- [14] Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09602* (2020).
- [15] Paul Erdős, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci.* 1 (1960), 17–60.
- [16] Socorro Gama-Castro, Verónica Jiménez-Jacinto, Martín Peralta-Gil, Alberto Santos-Zavaleta, Mónica I Peñaloza-Spinola, Bruno Contreras-Moreira, Juan Segura-Salazar, Luis Muniz-Rascado, Irma Martínez-Flores, Heladia Salgado, et al. 2008. RegulonDB (version 6.0): gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation. *Nucleic acids research* 36, suppl_1 (2008), D120–D124.
- [17] Jianxi Gao, Baruch Barzel, and Albert-László Barabási. 2016. Universal resilience patterns in complex networks. *Nature* 530, 7590 (2016), 307–312.
- [18] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. 2022. G-mixup: Graph data augmentation for graph classification. *International Conference on Machine Learning* PMLR, 8230–8248.
- [19] J Nathaniel Holland, Donald L DeAngelis, and Judith L Bronstein. 2002. Population dynamics and mutualism: functional responses of benefits and costs. *The American Naturalist* 159, 3 (2002), 231–244.
- [20] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social networks* 5, 2 (1983), 109–137.
- [21] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265* (2019).
- [22] Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. 2022. Graphgdp: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining (ICDM)*, 201–210.
- [23] Kexin Huang, Vishnu Sresht, Brajesh Rai, and Mykola Bordyuh. 2022. Uncertainty-aware pseudo-labeling for quantum calculations. *Uncertainty in Artificial Intelligence* PMLR, 853–862.
- [24] Eric Inae, Gang Liu, and Meng Jiang. 2023. Motif-aware Attribute Masking for Molecular Graph Pre-training. *arXiv preprint arXiv:2309.04562* (2023).
- [25] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label propagation for deep semi-supervised learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 2019, 5079.
- [26] Chunheng Jiang, Jianxi Gao, and Malik Magdon-Ismael. 2020. Inferring degrees from incomplete networks and nonlinear dynamics. *arXiv preprint arXiv:2004.10540* (2020).
- [27] Jaehyeon Jo, Seul Lee, and Sung Ju Hwang. 2022. Score-based generative modeling of graphs via the system of stochastic differential equations. *International Conference on Machine Learning* PMLR, 10362–10383.
- [28] Dongki Kim, Jinheon Baek, and Sung Ju Hwang. 2022. Graph self-supervised learning with accurate discrepancy learning. *Advances in Neural Information Processing Systems* 35 (2022), 14085–14098.
- [29] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
- [30] Edward Laurence, Nicolas Doyon, Louis J Dubé, and Patrick Desrosiers. 2019. Spectral dimension reduction of complex dynamical networks. *Physical Review X* 9, 1 (2019), 011042.
- [31] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, ICMML*. 3. Atlanta, 896.
- [32] Ruikun Li, Huandong Wang, Jinghua Piao, Qingmin Liao, and Yong Li. 2024. Predicting Long-term Dynamics of Complex Networks via Identifying Skeleton in Hyperbolic Space to appear in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2024).
- [33] Gang Liu, Eric Inae, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. 2024. Data-centric learning from unlabeled graphs with diffusion models. *Advances in neural information processing systems* 36 (2024).
- [34] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. 2022. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems* 35 (2022), 9754–9767.
- [35] Jinzhu Mao, Liu Cao, Chen Gao, Huandong Wang, Hangyu Fan, Depeng Jin, and Yong Li. 2023. Detecting vulnerable nodes in urban infrastructure interdependent network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* 617–627.
- [36] Jinzhu Mao, Dongyun Zou, Li Sheng, Siyi Liu, Chen Gao, Yue Wang, and Yong Li. 2024. Identify Critical Nodes in Complex Network with Large Language Models. *arXiv preprint arXiv:2403.03962* (2024).
- [37] Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. 2022. Spectre: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. *International Conference on Machine Learning* PMLR, 15159–15179.
- [38] Robert M May. 1977. Thresholds and breakpoints in ecosystems with a multiplicity of stable states. *Nature* 269, 5628 (1977), 471–477.
- [39] Flaviano Morone, Gino Del Ferraro, and Hernán A Makse. 2019. The k-core as a predictor of structural collapse in mutualistic ecosystems. *Nature physics* 15, 1 (2019), 95–102.
- [40] Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. 2020. Permutation invariant graph generation via score-based generative modeling. *International Conference on Artificial Intelligence and Statistics* PMLR, 4474–4484.
- [41] Je Ollerton, Duncan McCollin, Daphne G Fautin, and Gerald R Allen. 2007. Finding NEMO: nestedness engendered by mutualistic organization in anemone fish and their hosts. *Proceedings of the Royal Society B: Biological Sciences* 274, 1465–1469 (2007), 591–598.
- [42] Hillel Sanhedrai, Jianxi Gao, Amir Bashan, Moshe Schwartz, Shlomo Havlin, and Baruch Barzel. 2022. Reviving a failed network through microscopic interventions. *Nature Physics* 18, 3 (2022), 338–349.
- [43] M Ángeles Serrano, Dmitri Krioukov, and Marián Boguná. 2008. Self-similarity of complex networks and hidden metric spaces. *Physical review letters* 100, 7 (2008), 078701.
- [44] Hongyuan Su, Yu Zheng, Jingtao Ding, Depeng Jin, and Yong Li. 2024. Rumor Mitigation in Social Media Platforms with Deep Reinforcement Learning. In *Companion Proceedings of the ACM on Web Conference* 1240–1247.
- [45] Natasa Tagasovska and David Lopez-Paz. 2019. Single-model uncertainties for deep learning. *Advances in Neural Information Processing Systems* 32 (2019).
- [46] Alex M Tseng, Nathaniel Diamant, Tommaso Biancalani, and Gabriele Scalia. 2023. GraphGUIDE: interpretable and controllable conditional graph generation with discrete Bernoulli diffusion. *arXiv preprint arXiv:2302.03760* (2023).
- [47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [48] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph inductive representation learning. *arXiv preprint arXiv:1909.01317* (2019).
- [49] Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. 2023. Digress: Discrete denoising diffusion for graph generation. *ICLR* (2023).
- [50] Huandong Wang, Huan Yan, Can Rong, Yuan Yuan, Fenyu Jiang, Zhenyu Han, Hongjie Sui, Depeng Jin, and Yong Li. 2023. Multi-Scale Simulation of Complex Systems: A Perspective of Integrating Knowledge and Data. *Comput. Surveys* (2023).
- [51] Hugh R Wilson and Jack D Cowan. 1972. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal* 2, 1 (1972), 1–24.
- [52] Hugh R Wilson and Jack D Cowan. 1973. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* 13, 2 (1973), 55–80.

- [53] Mengkai Xu, Srinivasan Radhakrishnan, Sagar Kamarthi, and Xiaoning Jin. 2019. Resiliency of mutualistic supplier-manufacturer network. *Scientific reports*, 1 (2019), 13559.
- [54] Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. 2021. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*. PMLR, 11548-11558.
- [55] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. 2023. Di sound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).
- [56] Yuan Yuan, Chenyang Shao, Jingtao Ding, Depeng Jin, and Yong Li. 2024. Spatio-temporal Few-Shot Learning via Diffusive Neural Network Generation. *The Twelfth International Conference on Learning Representations*. <https://openreview.net/forum?id=QyFm3D3Tzi>
- [57] Chengxi Zang, Peng Cui, Christos Faloutsos, and Wenwu Zhu. 2018. On power law growth of social networks. *IEEE Transactions on Knowledge and Data Engineering*, 30, 9 (2018), 1727-1740.
- [58] Chengxi Zang and Fei Wang. 2020. Neural dynamics on complex networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. ACM, 92-902.
- [59] Huixin Zhang, Qi Wang, Weidong Zhang, Shlomo Havlin, and Jianxi Gao. 2022. Estimating comparable distances to tipping points across mutualistic systems by scaled recovery rates. *Nature Ecology & Evolution*, 10 (2022), 1524-1536.
- [60] Yongtao Zhang, Cunqi Shao, Shibo He, and Jianxi Gao. 2020. Resilience centrality in complex networks. *Physical Review E*, 101, 2 (2020), 022304.

A APPENDIX

A.1 Details of Parameterization Network

A.1.1 Graph Transformer. The parameterization of the denoising network h_θ employs the Graph Transformer architecture [1]. The input consists of the noisy graph features, and the output is the edge distribution of clean graphs. Each layer of the Graph Transformer can be represented as follows:

$$h_i^{l,1} = O_h^l K_{k=1}^K \left(\sum_{r_j \in 2N_{r_i}} \alpha_{ij}^{k,l} V^{k,l} h_j^l \right), \quad (25)$$

$$e_{ij}^{l,1} = O_e^l K_{k=1}^K \left(a_{ij}^{k,l} \right), \quad (26)$$

$$\alpha_{ij}^{k,l} = \text{softmax}_j \left(a_{ij}^{k,l} \right), \quad (27)$$

$$a_{ij}^{k,l} = \left(\frac{Q^{k,l} h_i^l \cdot P^{k,l} h_j^l}{\sqrt{d_k}} \right), W^{k,l} e_{ij}^l, \quad (28)$$

where h_i^l denotes the embedding of node i of the l -th layer, e_{ij}^l represents the embedding of edge connecting node i and j . O, Q, P, V , and W with different superscripts are trainable parameters. $k = 1, 2, \dots, K$ denotes the attention heads, and \cdot represents the concatenation operator. After the last layer of Graph Transformer, edge embeddings are fed into an MLP to predict the existence of edges $f_{ij} \in \{0, 1\}$, where N is the number of nodes in the network.

A.1.2 Node features. We include node features in both structural and spectral domains to enhance the performance of the Graph Transformer, and choose the same features as [4]. Specifically, structural features include the cycles, indicating how many k -cycles the node belongs to, since message-passing cannot detect cycle structures [10]. For spectral features, we first compute the graph Laplacian then consider the number of connected components and the two first eigenvectors of the non-zero eigenvalues.

A.2 Details of Data Collection

A.2.1 Mutualistic dynamics. Nodal state trajectories of networks in mutualistic dataset are simulated via the following differential

equations:

$$\frac{dx_i}{dt} = B_i x_i \left(1 - \frac{x_i}{K} \right) \left(\frac{x_i}{C} - 1 \right), \sum_{j=1}^N A_{ij} \frac{x_i x_j}{D_i + E x_i + H x_j}. \quad (29)$$

We use the fourth-order Runge-Kutta stepper, with a high initialization $x = 5$ and a low initialization $x = 0$, simulating two trajectories, which represent a thriving stable ecosystem and an ecosystem after a catastrophe, respectively. We set the terminal simulation time as $T_{max} = 50$, and the interval $\Delta t = 0.5$. Nodal states of networks with mutualistic dynamics can encounter a bifurcation [9], transitioning from a resilient phase characterized by a single, desired high equilibrium x^H to a non-resilient phase with both the desired equilibrium x^H and the low equilibrium x^L . We denote the averaged nodal states of the network from high and low initializations as \bar{x}^{1h^0} and \bar{x}^{1l^0} ; therefore, to define the resilience labels of networks, we compare \bar{x}^{1h^0} and \bar{x}^{1l^0} at the terminal time. If $|\bar{x}^{1h^0} - \bar{x}^{1l^0}| > r$, we conclude that the network cannot recover after perturbations and has two equilibria x^H and x^L , thus it is non-resilient. r is a pre-defined threshold, and we set $r = 3.5$ in our experiments.

A.2.2 Regulatory dynamics. Nodal state trajectories of networks in regulatory dataset are simulated via the following differential equations:

$$\frac{dx_i}{dt} = B x_i^f, \sum_{j=1}^N A_{ij} \frac{x_j^h}{x_j^h + 1}. \quad (30)$$

Similar to mutualistic dynamics, we use the fourth-order Runge-Kutta stepper, set the terminal simulation time $T_{max} = 50$ and the interval $\Delta t = 0.5$. Regulatory dynamics in Equ. (30) has a trivial fixed point (as well as equilibrium) $x = \bar{x} = 0$, and for resilient networks with this dynamics, its nodal states have another equilibrium $\bar{x} > 0$ [3]. To avoid the fixed-point equilibrium, we randomly initialize the model with $x = \bar{x} = 1.5$ and use the terminal nodal state \bar{x} to determine its resilience. Specifically, a network is deemed resilient if $\bar{x} > 0$. Conversely, the non-resilient network can only converge to the equilibrium of $\bar{x} = 0$.

A.2.3 Neuronal dynamics. Nodal state trajectories of networks in neuronal dataset are simulated with the following differential equations:

$$\frac{dx_i}{dt} = x_i, \sum_{j=1}^N A_{ij} \frac{1}{1 + e^{\mu \delta x_j}}. \quad (31)$$

The ODE solver, terminal, and interval time settings are the same as mutualistic and regulatory dynamics. Non-resilient networks exhibit either a bi-stable phase, wherein both a high equilibrium, denoted as x^H , and a low equilibrium, denoted as x^L , can exist, or only a single low equilibrium x^L exists. On the other hand, resilient neuronal networks are distinguished by their maintenance of a high equilibrium x^H [51]. We initialize the nodal state with a high initialization $x = 5$ and a low initialization $x = 0$, simulating two trajectories. We compare \bar{x}^{1h^0} and \bar{x}^{1l^0} at the terminal time to define the resilience labels of networks. If $|\bar{x}^{1h^0} - \bar{x}^{1l^0}| > r$ or $\bar{x}^{1h^0} < m$ and $\bar{x}^{1l^0} < m$, we conclude that the network cannot recover after perturbations and have two equilibria x^H and x^L , thus it is non-resilient. Otherwise, the network is resilient and

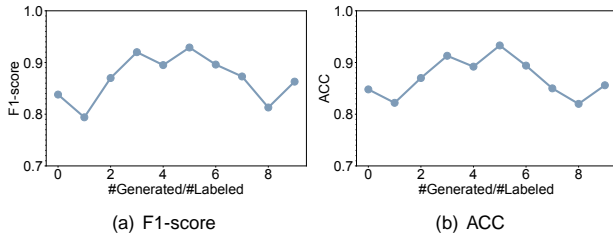


Figure 8: Model performance with different number of generated samples on mutualistic dataset.

m are pre-defined thresholds, and we set $\epsilon = 3.5$ and $m = 3$ in our experiments.

A.2.4 Barabási-Albert (BA) model. BA network starts with a small number of nodes, and at each time step, a new node with edges is added to the network. These edges link the new node to different nodes that have already present in network. The probability Π that a new node will connect to an existing nodes is proportional to the degree k_i of node i . Mathematically, $\Pi \propto \frac{k_i}{\sum_j k_j}$, where the summation is over all existing nodes in the network. This means that nodes with higher degrees have a higher likelihood of receiving new links, leading to a rich-get-richer effect. The resulting network from the BA model exhibits a power-law degree distribution, $P(k) \propto k^{-\gamma}$, where γ is typically in the range of 2 to 3. In our BA network datasets, each network consists of 1000 nodes, and we set $\epsilon = 4$.

A.2.5 S^1/H^2 model. It is also called hyperbolic geometric graph model. In this model, nodes are placed in a hyperbolic disk. Each node is assigned to a radial coordinate and an angular coordinate θ . r follows an exponential distribution capturing the heterogeneity of node degrees, while θ is uniformly distributed between 0 and 2π representing the similarity or latent feature spaces of nodes. The connection probability of node i and j depends on their hyperbolic distance $d_{ij} = r_i + r_j + 2 \log \sin \frac{\theta_{ij}}{2}$. Nodes are more likely connected if they are close in the hyperbolic space, reflecting the principle of preferential attachment and similarity. In our datasets, each network consists of 1000 nodes, and the inverse temperature controlling the clustering coefficient β , the exponent of the power-law distribution for hidden degrees, and the mean degree of the network are set to 1.5, 2.7, and 5, respectively.

A.2.6 Stochastic block model (SBM). In this model, nodes are partitioned into K distinct groups. The probability of an edge existing between any two nodes depends solely on the groups to which these nodes belong. In our SBM datasets, each network consists of 1000 communities and 20, 40% nodes (both sampled uniformly). The inter-community edge probability is 0.3, and the intra-community edge probability is 0.05.

A.2.7 Empirical brain networks. we employ an empirical brain networks [7, 42] with 998 brain regions (nodes), which represents the physical fiber bundle connections between them. The empirical network also has a natural modular structure owing to the brain's two hemispheres, indicating its complex structural properties.

We generate 1900 topologies for unlabeled data, 100 topologies for labeled data, and 200 test data by randomly remove 10% nodes from the empirical topology.

A.3 Details of Theoretical Baseline

In this section, we detail how we incorporate resilience theory from physics to provide insights on leveraging unlabeled data.

A.3.1 Gao-Barzel-Barabási (GBB) theory. From Gao-Barzel-Barabási (GBB) theory [7], for a network with N nodes, N -dimensional nodal state dynamics represented by Equ. (1) can be condensed to a 1-dimensional equation as:

$$\frac{dx_e}{dt} = F^1 x_e^0, \beta_e G^1 x_e, x_e^0, \quad (32)$$

$$x_e = \frac{1^T A x}{1^T A 1} = \frac{\mathbf{hs}^{out} \mathbf{x}_i}{\mathbf{hs}_i}, \quad (33)$$

$$\beta_e = \frac{1^T A s^{in}}{1^T A 1} = \frac{\mathbf{hs}^{out} s^{in}_i}{\mathbf{hs}_i}, \quad (34)$$

where $s^{out} = 1 s_1^{out}, \dots, s_N^{out}$ denotes the out-degrees of nodes and $s^{in} = 1 s_1^{in}, \dots, s_N^{in}$ denotes their in-degrees. $\mathbf{hs}^{out} \mathbf{x}_i = \frac{1}{N} \sum_{i=1}^N s_i^{out} x_i$ and $\mathbf{hs}_i = \mathbf{hs}^{in}_i = \mathbf{hs}^{out}_i$. Therefore, we can observe that network topology $A \in \mathbb{R}^{N \times N}$ is condensed to a scalar $\beta_e \in \mathbb{R}$, which embeds the features of network topologies. From dynamics equation, $f^1 \beta_e, x_e^0 = F^1 x_e^0, \beta_e G^1 x_e, x_e^0$, we can identify the bifurcation point of this dynamics β_e^c . If $\beta_e < \beta_e^c$, the undesired equilibrium will emerge, or the desired equilibrium will vanish. Otherwise, it has only one desirable equilibrium. Since β_e is calculated from network topology, we can conclude that a network with $\beta_e < \beta_e^c$ is non-resilient, and that of $\beta_e > \beta_e^c$ is resilient. Therefore, GBB theory provides an effective tool to predict network resilience. Its limitation is that precise analytical forms of f^1 and G^1 are required, which is hard to determine in real-world scenarios.

A.3.2 Theory-guided data augmentation. Here we discuss how to use GBB theory to perform data augmentation. For each network topology in labeled and unlabeled dataset, we calculate β_e from Equ. (34). After that, we denote the minimum β_e of resilient networks in the labeled dataset as β^* , and denote the maximum β_e of non-resilient networks in the labeled dataset as $\beta^{\#}$. Therefore, for each network in the unlabeled dataset, if $\beta_e > \beta^*$, we label it as the resilient network; if its $\beta_e < \beta^{\#}$, we label it as the non-resilient network. Then the resilience predictor can further train on these newly-labeled data to enhance its predictive performance.

A.4 Additional Experiments

Number of generated samples. We investigate the effect of used number of generated samples on the augmentation performance. The results on mutualistic dataset are shown in Figure 8, where the point in the 0 position of x-axis indicates the performance of the vanilla model. We find that there is an upper bound on the improvement introduced by data augmentation. Since we use the sub-optimal resilience predictor to guide the generation process, it is unavoidable to introduce generated data with fault labels. When the number of introduced generated data exceeds a threshold, the defect of noisy labels will exceed the positive effect of new training data, leading to the decrease of model performance.