Deep Graph Spectral Evolution Networks for Graph Topological Evolution

Negar Etemadyrad [1], Qingzhe Li [1,*], Liang Zhao [1,2,†]
[1] George Mason University, Fairfax, VA, 22033
[2] Emory University, Atlanta, GA, 30307
netemady@gmu.edu, qli10@gmu.edu, liang.zhao@emory.edu

Abstract

Characterizing the underlying mechanism of graph topological evolution from a source graph to a target graph has attracted fast increasing attention in the deep graph learning domain. However, it is very challenging to build expressive and efficient models that can handle global and local evolution patterns between source and target graphs. On the other hand, graph topological evolution has been investigated in the graph signal processing domain historically, but it involves intensive labors to manually determine suitable prescribed spectral models and prohibitive difficulty to fit their potential combinations and compositions. To address these challenges, this paper proposes the deep Graph Spectral Evolution Network (GSEN) for modeling the graph topology evolution problem by the composition of newly-developed generalized graph kernels. GSEN can effectively fit a wide range of existing graph kernels and their combinations and compositions with the theoretical guarantee and experimental verification. GSEN has outstanding efficiency in terms of time complexity ($O(n)$) and parameter complexity ($O(1)$), where n is the number of nodes of the graph. Extensive experiments on multiple synthetic and real-world datasets demonstrate outstanding performance.

Introduction

Understanding the mechanism of graph generation and evolution has significant importance in many applications (Guo and Zhao 2020; Guo et al. 2020; You et al. 2018), such as brain simulation, mobility network simulation, and social network modeling and intervention (Zhao 2020). Beyond the traditional methods from network science domain, graph generation and evolution have been attracting fast increase attention by deep graph generative models due to their great potential of learning the underlying known generation and evolution mechanism in an end-to-end fashion. Existing deep generative models for graphs are typically unconditional and generate graphs from random noise. But in many cases, we need a "conditional setting", by generating a target graph given a source graph. Based on deep graph generative models, the graph generation problem is considered as decoding a graph based on latent variables following some underlying distribution while graph evolution can be modeled as a mapping to a target graph topology given a source graph topology. Graph evolution based on deep graph learning is a very challenging problem and is still in its nascent stage because of the extremely high dimensionality of the data. An ideal model should be able to capture both the local and global characteristics of source graph and be able to determine the existence or weight of potential edges for each pair of nodes. Models that are both expressive and efficient are in urgent demand.

The domain which has investigated graph evolution for a long time is graph signal processing, where well-defined mathematical framework and various techniques such as graph wavelets and kernels that abstract the graph process in frequency domain have been hypothesized and verified in many applications. For example, Kunegis et al. have demonstrated that triangle-closing kernels fit very well to the evolution of graph spectrum during the "befriending process" in some social networks ((Leskovec et al. 2008)). Most recently, neuroscience researchers found that the functional connectivity shares the same graph Fourier basis with structural connectivity in several special situations.

Although graph signal processing allows powerful and concise models to characterize many graph evolution processes, it requires to first determine the potentially suitable type of graph kernel and then fit the parameters of it. However, this raises up serious challenges: First, it is difficult to discover or select suitable kernel types for various applications. Graph kernels are proposed based on the analyses and abstraction of the prior knowledge on various graph phenomena. But until now, quite a lot of phenomena have not yet been analyzed or interpreted by humans. For example, it is unclear whether and how the spectrum of resting-state functional connectivity transforms into task-specific functional connectivity in human brain (Hermundstad et al. 2013). Moreover, for many sophisticated phenomena, the graph process typically involves the combination and composition of multiple graph processes corresponding to multiple kernels. For example, the evolution of a social networks might involve not only the triangle closing process (i.e., two friends of a person tend to be friends) by triangle-closing kernels, but also the behavior diffusion process which can be characterized by diffusion kernels. Also, the involvement of different kernel process might be simultaneous or sequential, and hence prohibitively difficult to manually determine or combinatorially optimize.

[*]Equally contributing as first author
[†]Corresponding author

To address these challenges, this paper proposes a novel end-to-end model Deep Graph Spectral Evolution Network (GSEN) to optimally fit the graph evolution process by the composition of newly-developed generalized graph kernels. The generalized graph kernels widely cover existing graph kernels besides their combination and composition as special cases, and hence are able to fit them with outstanding expressiveness. Along with this high expressiveness, GSEN is highly concise in terms of small parameter and time complexity for training. Specifically, the number of parameters and memory complexity of GSEN are independent of the graph size (number of nodes) while the time complexity for the training of GSEN is linear to the graph size. This largely outperforms the state-of-the-art, which typically requires $O(n^2)$ time complexity and memory complexity. Extensive experiments on several synthetic datasets and multiple real-world datasets in two domains have been conducted. The results demonstrate the superior accuracy of GSEN over existing deep generative models and models based on graph signal processing, besides higher efficiency of GSEN compared to existing deep generative models.

Related Work

Spectral Graph Translation Problems

Spectral based approaches in graph translation have been the focus in many researches over the past decades. To model how networks are translated, the spectral evolution model was introduced by (Kunegis, Fay, and Bauckhage 2010). The growth of large networks is analyzed by studying the changes in the spectral characteristics of the graph. These changes are explained using the eigendecomposition of the graph adjacency matrix or its Laplacian. The new link prediction approach shows how eigenvectors stay constant while the eigenvalues are evolved over the transition. This model can also generalize several graph kernels which are expressed as spectral transformations. (Li, Yu, and Liu 2011) proposes the MERW (maximum entropy random walk) approach to the link prediction problem. MERW based approaches are introduced as various algorithms that could use four separate graph kernels, in addition to a class of similarity measures, to capture the proximity between two nodes. The resulting methods perform the prediction, while maintaining the centrality of the nodes. In (Symeonidis et al. 2013), the link prediction problem for protein-protein interaction networks and online social networks is considered. The SpectralLink algorithm is proposed to compute the similarity between every two nodes, by exploiting the top few eigenvectors of the laplacian matrix, which eliminates the redundant and noisy information. The link prediction is then performed faster and more accurate. Variants of the aforementioned method are also derived for signed and directed graphs. Spectral Graph analysis has been useful in a behavior related link prediction problem (Spiegel et al. 2011; Zhao 2020), where there's a need to predict whether and how much a user is likely to rate an item. Multiple network snapshots with temporal trends are captured and tensor factorization is used to extract hidden trends within a multi-dimension array. The higher-order data is then factorized into a lower dimension, using Parafac model. The spectral evolution model is then applied, where the spectrum does change, while the eigenvectors stay constant.

Deep Learning Methods in Graph Spectral Domain

There is a large body of research on deep graph learning, for tasks such as the embedding and classification of nodes and graphs. (Kipf and Welling 2016) proposed a localized graph convolutional neural networks (CNNs) based on semi-supervised learning for graph-structured data, where labels are available for a small subset of nodes. A neural network model is designed based on a layer-wise propagation rule. The model is then trained on the supervised target which includes all labeled nodes. A novel spectral graph CNN approach is proposed in (Li et al. 2018) to graph data that varies in both size and connectivity. To capture the variation in the input graph topology, training process includes applying a customized graph Laplacian to each sample input. The Laplacian then becomes trainable by parameterizing the distance metrics that measure vertex similarity. Deep convolutional approaches have been applied to data domains with irregularities which lack fundamental statistical properties in (Henaff, Bruna, and LeCun 2015), to solve for large scale classification problems. In (Defferrard, Bresson, and Vandergheynst 2016), CNNs are presented in the context of spectral graph theory, and fast localized convolutional filters are designed.

Deep Learning Methods for Graph Transformation

Graph Transformation modeling based on deep neural networks has attracted fast-increasing attention recently, where existing methods are based on spatial domain by operating the explicit connectivity among the nodes ((Guo et al. 2019), (Guo, Wu, and Zhao 2018), (Do, Tran, and Venkatesh 2019)). A systematic survey can be found here (Guo and Zhao 2020). The prediction in most cases is performed either on node attributes of the graph or its topology while the other is fixed. (Guo et al. 2019) proposes NEC-DGT (Node-Edge Co-evolving Deep Graph Translator) framework as a novel technique to approach the simultaneous prediction challenge. A portion of this research is only tailored for specific applications and domains ((Do, Tran, and Venkatesh 2019)). For example, (Do, Tran, and Venkatesh 2019) and (Jin et al. 2018) proposed methods only for transferring molecular graphs. Spatio-temporal dependencies in traffic flow are modeled as a diffusion process in a directed graph through DCRNN (Diffusion Convolutional Recurrent Neural Network) model (Li et al. 2017). Using bidirectional random walks and encoder-decoder architecture the spatial and temporal dependencies are captured respectively. In (Li, Guo, and Mei 2016), the authors propose DeepGraph model to learn topological structure of a network from the raw adjacency matrix as input, and use that to predict the growth of the network. However, until now there is no work in this domain that models the graph topological transformation in spectral domain.

Graph Topology Transformation via Spectral Evolution

This paper focuses on a problem of using the topology of a source graph to predict that of a target graph by characterizing spectral graph evolution.

Problem Formulation

Define a source graph as an undirected weighted graph $G = (V, E, A)$ where V is the set of nodes with size of $|V|$, $E \subseteq V \times V$ is the set of edges, and $A \in \mathbb{R}^{|V|\times|V|}$ is the adjacency matrix that defines the weights of the edges. The adjacency matrix A can be normalized by defining the normalized adjacency matrix $N = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where $D \in \mathbb{R}^{|V|\times|V|}$ is the diagonal matrix where each diagonal element is the degree of corresponding node. Besides, the Laplacian matrix of the source graph G is defined as $L = D - A$, and the normalized Laplacian matrix is defined as $Z = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - N$ where I is the identity matrix. A target graph is defined as $G' = (V', E', A')$, where the set of nodes $V' = V$, edges E', adjacency matrix A', normalized adjacency matrix N', the Laplacian matrix L', the normalized Laplacian matrix Z'.

Definition 1 (Graph Transformation via Spectral Evolution)**.** *The spectral graph translation problem states that the graph topological transformation $G' \leftarrow F(G)$ from a source graph G to target graph G' can be modeled by a change in graph's spectrum, while graph Fourier basis remains the similar.*

To determine the function F, various graph kernels based on the existing research on graph wavelets can be utilized, including heat kernels $\mathcal{K}_{\text{HEAT}}(L) = \exp(-\alpha L)$ and many others such as those illustrated in Table 1. Several such kernels have been empirically demonstrated to model specific graph processes effectively. For example, the evolution of a social networks typically involves triangle closing process (i.e., two friends of a person tend to be friends), which has been verified to be effectively modeled by triangle-closing kernels (Leskovec et al. 2008). Path count kernels have been verified to fit very well into the link prediction problem in some email networks (Kunegis, Fay, and Bauckhage 2010)).

Generalized Graph Kernels

We propose a new nonparametric kernel that is highly expressive to cover various graph kernels as well as their compositions. We first formulate the learning of such expressive kernel as an optimization problem as follows.

Lemma 0.1. *Using matrices (e.g., adjacency matrix, graph Laplacian, etc.) X and X' to represent the graph topology G and G', we have $X = U\Lambda U^\intercal$ and $X' = U'\Lambda' U'^\intercal$ according to eigen-decomposition. Then the spectral graph translation problem in Definition 1 can be explicitly formulated as $F(X) \to X'$ by an analytic function F, which is learned by the following equation, given $[F(\Lambda)]_{kk} = f(\Lambda_{kk})$:*

$$\min_F \|F(X) - X'\|_2^2 = \min_f \sum_k (f(\Lambda_{k,k}) - \Lambda'_{k,k})^2 \quad (1)$$

Proof. The training purpose of $F(\cdot)$ is to minimize the squared loss against the real target graph:

$$\begin{aligned}
&\|F(X) - X'\|_2^2 = \|F(U\Lambda U^\intercal) - X'\|_2^2 \\
&= \|\textstyle\sum_{k=0}^{\infty} \frac{F^{(k)}(\gamma)}{k!}(U\Lambda U^\intercal)^k - X'\|_2^2 \quad \text{(Power Expansion)} \\
&= \|U\left(\sum_{k=0}^{\infty} \frac{F^{(k)}(\gamma)}{k!}(\Lambda)^k\right)U^\intercal - X'\|_2^2 = \|UF(\Lambda)U^\intercal - X'\|_2^2 \\
&= \|U \cdot diag([F(\Lambda_{1,1}), \cdots, F(\Lambda_{|V|,|V|})])U^\intercal - X'\|_2^2 \\
&= \|diag([F(\Lambda_{1,1}), \cdots, F(\Lambda_{|V|,|V|})]) - U^\intercal X' U\|_2^2 \quad (UU^\intercal = I) \\
&= \|f(\Lambda_{k,k}) - U^\intercal U'\Lambda' U'^\intercal U\|_2^2 \\
&= \min_f \sum_k (f(\Lambda_{k,k}) - \Lambda'_{k,k})^2 \quad (\|U - U'\| \to 0 \text{ by Definition 1})
\end{aligned}$$

The proof is completed. □

We propose the following new generalized graph kernel:

$$F(\Lambda) = \sum_{k=1}^{\infty} \alpha_k \Lambda^k + \gamma_k D^{-k}\Lambda^k + \beta I \quad (2)$$

The following introduces some important properties of this operation.

Lemma 0.2. *The generalized graph kernel in Equation 2 has the following properties:*

1. *Various existing graph kernels such as those listed in Table 1 are special cases of our operation.*
2. *The additive combinations and compositions of the existing graph kernels are special cases of our operation.*

Proof. Property 1. Graph kernels are typically under four types, namely Laplacian L, adjacency matrix $A = D - L$, normalized Laplacian $Z = D^{-1/2}LD^{-1/2}$, and normalized adjacency matrix $N = I - D^{-1/2}LD^{-1/2}$, based on their immediate inputs. Without loss of generality, we assume we use graph Laplacian to represent the graph and construct our kernel, $L = U\Lambda U^\intercal$, though other forms can also accomplish the proof. For those kernels based on L and A, they can be transformed to $[F(\Lambda)]_{i,i} = \sum_{k=0}^{\infty} \frac{m_{k,i}}{k!}\Lambda_{i,i}^k$, where $m_{k,i} = f^{(k)}(0)$ for Laplacian while $m_{k,i} = f^{(k)}(D_{i,i})$ for adjacency matrix. Therefore, both of them can be fit into our generalized graph kernel by setting $\gamma_k := 0$ for all $k = 0, 1, \cdots$. For those kernels based on N and Z, they can be transformed to $[F(\Lambda)]_{i,i} = \sum_{k=0}^{\infty} \frac{m}{k!}(D_{i,i}^{-k}\Lambda_{i,i})^k$, where $m = 0$ for Z while $m = 1$ for N. Hence, both of them can be fit by setting $\alpha_k := 0$ for all $k = 0, 1, \cdots$.

Property 2. Consider two generalized graph kernels $F_a(\Lambda)$ and $F_b(\Lambda)$. Specifically, for summation, two graph kernels can be transformed into corresponding generalized kernels whose sum is also a generalized kernel (i.e., Equation 2), and for composition, each kernel can be transformed into a polynomial (by Equation 2) and a polynomial of a polynomial is still a polynomial, and is covered by Equation 2. □

Deep Graph Spectral Evolution Networks

Here, a neural network model based on the proposed generalized graph kernel is established, by reducing order of the polynomials from infinity to K which is independent of and

Kernel Name	Matrix Function	Spectral Function
Laplacian Commute-time Kernel	$\mathcal{K}_{\text{Com}}(L) = L^+$	$U\Lambda^{-1}U^\intercal$, define $\Lambda_{i,i}^{-1} = 0$ if $\Lambda_{i,i} = 0$
Normalized Laplacian Commute-time Kernel	$\mathcal{K}_{\text{Com}}(Z) = Z^+$	$U\Lambda^{-1}U^\intercal$, define $\Lambda_{i,i}^{-1} = 0$ if $\Lambda_{i,i} = 0$
Normalized Adjacency Exponential Kernel	$\mathcal{K}_{\text{Exp}}(N) = e^{\alpha N}$	$Ue^{\alpha\Lambda}U^\intercal$
Generalized Laplacian Kernel	$\mathcal{K}_{\text{Gen}}(L) = (\sum_{k=0}^{\infty} \alpha_k L^k)^+$	$U(\sum_{k=0}^{\infty} \alpha_k \Lambda^k)^{-1}U^\intercal$
Generalized Normalized Laplacian Kernel	$\mathcal{K}_{\text{Gen}}(Z) = \sum_{k=0}^{\infty} \alpha_k (I - Z)^k$	$U(\sum_{k=0}^{\infty} \alpha_k (I - \Lambda)^k)U^\intercal$
Heat Diffusion Kernel	$\mathcal{K}_{\text{Heat}}(L) = e^{-\alpha L}$	$Ue^{-\alpha\Lambda}U^\intercal$
Normalized Heat Diffusion Kernel	$\mathcal{K}_{\text{Heat}}(Z) = e^{-\alpha Z}$	$Ue^{-\alpha\Lambda}U^\intercal$
Normalized Adjacency Neumann Kernel	$\mathcal{K}_{\text{Neu}}(N) = (I - \alpha N)^{-1}$	$U(I - \alpha N)^{-1}U^\intercal$
Normalized Adjacency Path Count Kernel	$\mathcal{K}_{\text{Path}}(N) = \sum_{k=0}^{\infty} \alpha_k N^k$	$\sum_{k=0}^{\infty} \alpha_k (UD^{-\frac{1}{2}}\Lambda D^{-\frac{1}{2}}U^\intercal)^k$
Regularized Laplacian Kernel	$\mathcal{K}_{\text{Reg}}(N)(I + \alpha N)^{-1}$	$U(I + \alpha\Lambda)^{-1}U^\intercal$
Normalized Regularized Laplacian Kernel	$\mathcal{K}_{\text{Reg}}(Z)(I + \alpha Z)^{-1}$	$U(I + \alpha\Lambda)^{-1}U^\intercal$

Table 1: Existing kernels for graph spectral translation problem

typically far less than the graph size. Moreover, our neural network is composed by stacking multiple such generalized graph kernels as a special type of multi-order 1-D convolution operation, as illustrated in Figure 1.

Specifically, each layer can be expressed as follows.

$$F_l(\Lambda) = H_l\left(\sum\nolimits_{k=1}^{K}(\alpha_k I + \gamma_k D^{-k})F_{l-1}(\Lambda)^k + \beta I\right) \tag{3}$$

where the function $H_l(\cdot)$ is an activation function which performs element-wise activation based on the most common units such as ReLU, sigmoid, or linear. An equivalent scalar form of the above equation is expressed as $f_l(\Lambda_{i,i}) = h_l\left(\sum_{k=1}^{K}(\alpha_k + \gamma_k D_{i,i}^{-k})f_{l-1}(\Lambda_{i,i})^k + \beta\right)$, where $h_l(\cdot)$ is a scalar version of $H_l(\cdot)$.

As shown in Figure 1, we implement the neural network through an M-layer convolution operations from the source to target graph. Specifically, the input, namely $F_0(\Lambda)$, is Λ that is the graph spectrum. For the $l-1$th layer, the diagonal vectors of the matrices $I, F_{l-1}(\Lambda), F_{l-1}(\Lambda)^2, \cdots, F_{l-1}(\Lambda)^K$ are calculated and concatenated as shown in the orange region in Figure 1. Similarly, the diagonal vectors of the matrices $I, D \cdot F_{l-1}(\Lambda), D^2 \cdot F_{l-1}(\Lambda)^2, \cdots, D^K \cdot F_{l-1}(\Lambda)^K$ are calculated and concatenated as shown in the yellow region in Figure 1. Then these two regions are convoluted by the kernels $\alpha^{(l)}$ and $\gamma^{(l)}$, respectively, to obtain $F_l(\Lambda)$ after performing activation function. Such convolution operation is repeatedly performed until M-th layer, which outputs the predicted graph spectrum $F_M(\Lambda)$ for the target graph.

Complexity and Efficiency: Training neural network amounts to solving the optimization problem in Equation 1, which can be handled by backpropagation. Our method largely and effectively reduces the number of parameters, to $2 \cdot K \cdot M$, which is small and independent of the size of the graph and hence is highly memory-efficient and scalable. In terms of the time complexity, the calculation of the powers of graph spectrum has a complexity of $O(K \cdot N \cdot M)$ while the convolution operations involves another $O(K \cdot N \cdot M)$ so the total time complexity of the neural network is $O(K \cdot N \cdot M)$. Note that we focus on the training runtime (i.e., backpropagation), and eigen-decomposition is pre-computed outside backpropagation, so is excluded. Also, the generation of the input data involves eigen-decomposition of the adjacency matrix (or graph Laplacian), which could be time-consuming for large graphs. To address this issue, we can leverage reduced eigen-decomposition to only involve the calculation of lower-rank matrix and hence largely speed up this process.

Experiment

In this section, the experimental settings are first introduced, then the performance of the proposed method is presented through a set of comprehensive experiments, conducted on a 64-bit machine with 40 GB memory, a 4-core Intel ® CPU and an Nvidia ® RTX-2080 Ti GPU. The proposed method is implemented with Pytorch deep learning framework.[1]

Experimental Setup

We evaluate the effectiveness on synthetic and real-world datasets on brain network prediction and malware confinement in the Internet of Things (IoT) task. The datasets, evaluation and comparison methods are elaborated in turn.

Datasets Three datasets are involved for evaluations. • **Synthetic Datasets:** For each of the 11 synthetic datasets, we generate 1000 source-target graph pairs by first generating 1000 unweighted and undirected random graphs with 50 nodes and 200 edges as the source, using Erdős–Rényi model (Erdos and Renyi 1960). Each edge in this graph is assigned a random weight between 0 and 1. Finally, 1000 target graphs are generated by applying one of the kernels in Table 1. Each synthetic dataset utilizes one distinct kernel.

• **Real-world HCP Dataset:** Here, the source and the target graphs respectively reflect structural connectivity (SC) and functional connectivity (FC) of the same subject's brain network. In particular, both types of connectivity are processed from the Magnetic Resonance Imaging (MRI) data obtained from the human connectome project (HCP) (Van Essen et al. 2013) [2]. By following the preprocessing procedure in (Wang et al. 2019), the SC data is constructed by applying probabilistic tracking on the diffusion MRI data using the Probtrackx tool from FMRIB Software Library (Jenkinson

[1]https://github.com/netemady/GSEN

[2]http://www.humanconnectomeproject.org/

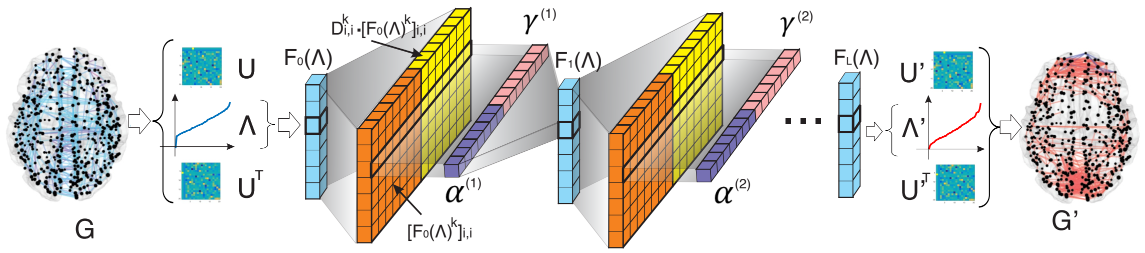


Figure 1: The architecture of Deep Graph Spectral Evolution Networks.

	Dataset											
Method	$\mathcal{K}_{\text{Com}}(L)$	$\mathcal{K}_{\text{Com}}(Z)$	$\mathcal{K}_{\text{Exp}}(N)$	$\mathcal{K}_{\text{Gen}}(L)$	$\mathcal{K}_{\text{Gen}}(Z)$	$\mathcal{K}_{\text{Heat}}(L)$	$\mathcal{K}_{\text{Heat}}(Z)$	$\mathcal{K}_{\text{Neu}}(N)$	$\mathcal{K}_{\text{Path}}(N)$	$\mathcal{K}_{\text{Reg}}(L)$	$\mathcal{K}_{\text{Reg}}(Z)$	Overall
$\mathcal{K}_{\text{Com}}(L)$	-	-	0.28	-0.04	-0.15	-0.26	-0.26	0.23	0.26	-0.02	-0.23	0.16
$\mathcal{K}_{\text{Com}}(Z)$	-	-	0.28	-0.04	-0.15	-0.26	-0.26	0.23	0.26	-0.02	-0.23	0.16
$\mathcal{K}_{\text{Exp}}(N)$	0.23	0.23	-	0.25	0.69	**1.00**	**1.00**	-0.88	-0.91	0.13	-0.88	0.17
$\mathcal{K}_{\text{Gen}}(L)$	-0.11	-0.28	-**1.00**	-	0.83	0.06	0.99	-0.94	-0.99	**0.80**	0.95	0.11
$\mathcal{K}_{\text{Gen}}(Z)$	-0.02	-0.18	-0.96	-0.05	-	0.21	0.84	-0.84	-0.92	-0.02	0.98	0.00
$\mathcal{K}_{\text{Heat}}(L)$	0.23	0.23	**1.00**	0.25	0.69	-	-	-0.88	-0.91	0.13	-0.88	0.17
$\mathcal{K}_{\text{Heat}}(Z)$	0.23	0.23	**1.00**	0.25	0.69	-	-	-0.88	-0.91	0.13	-0.88	0.17
$\mathcal{K}_{\text{Neu}}(N)$	0.25	0.29	0.93	-0.08	-0.76	-0.87	-0.99	-	0.98	0.02	-0.91	-0.01
$\mathcal{K}_{\text{Path}}(N)$	0.01	0.29	0.95	0.03	-0.77	-0.18	-0.99	**1.00**	-	0.01	-0.91	0.04
$\mathcal{K}_{\text{Reg}}(L)$	-0.02	-0.23	-0.99	0.36	0.96	-0.11	0.91	-0.91	-0.97	-	**1.00**	0.09
$\mathcal{K}_{\text{Reg}}(Z)$	-0.02	-0.23	-0.99	0.31	0.97	-0.10	0.91	-0.91	-0.97	0.65	-	0.06
GT-GAN	0.12	0.18	0.26	0.00	0.93	0.48	0.25	0.53	0.69	0.18	0.53	0.38
C-DGT	-0.05	0.16	**1.00**	-0.02	0.80	**1.00**	**1.00**	0.92	0.98	-0.02	0.92	0.61
Baseline	-0.14	-0.28	-0.99	-0.05	0.27	0.07	0.17	0.76	0.99	-0.03	0.63	0.13
GSEN	**0.97**	**0.72**	**1.00**	**0.80**	**1.00**	0.89	**1.00**	**1.00**	**1.00**	0.71	0.85	**0.90**

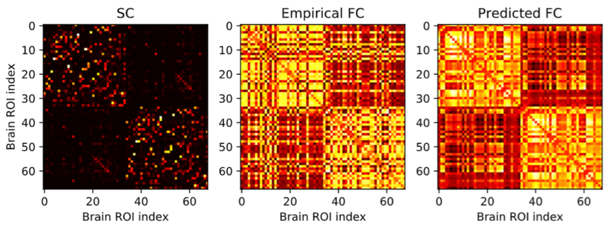
Table 2: Pearson correlation between predicted and empirical graph on synthetic datasets. Each column denotes Pearson correlation of the synthetic dataset generated by kernel function of the second row. Each row denotes Pearson correlation of the prediction method of first column. Some cells are "gold standard" because the predictor and synthetic data generator use the same graph kernels, hence are marked as "-". For those cells the right-most column denotes the average Pearson correlation among all synthetic datasets. The highest Pearson correlation in each column/dataset is highlighted in bold font while the second highest Pearson correlation is marked with underline.

et al. 2012) with 68 predefined regions of interests (ROIs). Then, the FC is defined as the Pearson's correlation between two ROIs' blood oxygen level-dependent time obtained from the resting-state functional MRI data. All the 823 pairs of SC and FC adjacency matrices are normalized as defined in Section . • **Real-world IoT Datasets:** In these datasets, the nodes represent the Internet of Things (IoT) devices and the edges denote the communication links between two devices. Each source graph reflects the communication status of the network, and some of the nodes in the network are infected by some types of malware. To limit the devices that are infected by the malware propagating to other devices, the malware confinement is conducted by cutting some of the links while maximizing the functionality of the network. The confined network is considered as the target graph that corresponds to the source graph. The IoT datasets contain three datasets, namely IoT-20, IoT-40, and IoT-60, which include 20, 40, and 60 devices. There are 343 source-target graph pairs in each IoT dataset.

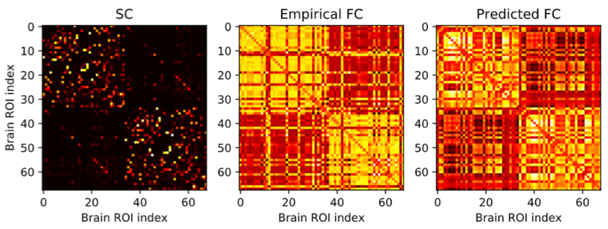
Methods and Settings The comparison methods include: **Graph spectral transformation kernels:** We compared our method with all single kernel methods defined in Table 1 on synthetic datasets. The parameters α or $\{\alpha_k\}_{k=1}^{K}$ in Table 1 is learned from the training data. **Baseline method:** For this method, the eigenvalue transformation function $\mathcal{F} : \Lambda \rightarrow \Lambda'$ is learned by a fully connected four-layer perceptron activated by tanh function. Each hidden layer contains $4n$ neurons, where n is the number of nodes in the graph. This fully connected network is optimized by the ADAM algorithm with the learning rate of 0.001 and 1000 epochs. The mean squared error loss is utilized for the baseline method. **Brain network prediction methods:** We consider four classic brain network prediction methods that use SC to FC (Galán 2008; Abdelnour, Voss, and Raj 2014; Meier et al. 2016; Abdelnour et al. 2018). (Abdelnour, Voss, and Raj 2014) and (Abdelnour et al. 2018) considered the graph spectral transformation kernels by assuming that SC and FC share the identical eigenvectors on their Laplacians. The remaining two methods directly consider the graph translation between SC and FC. **GT-GAN**: Graph Translation-Generative Adversarial Networks (GT-GAN) by (Guo, Wu, and Zhao 2018) is a newly proposed general-purpose graph topology translation method based

	Dataset									
Method	IoT-20		IoT-40		IoT-60		SC-FC		Overall	
	PR	R^2	PR	R^2	PR	R^2	PR	R^2	PR	R^2
Galan2008	0.74	0.54	0.79	0.60	0.81	0.65	0.23	-5.7	0.64	-0.99
Abdelnour2014	0.73	-1.83	0.76	-0.00	0.81	-0.64	0.23	-0.88	0.63	-0.83
Meier2016	0.74	0.54	0.78	0.60	0.81	0.65	0.26	-3.55	0.65	-0.44
Abdelnour2018	0.73	-0.00	0.76	-0.00	0.81	-0.00	0.23	-0.88	0.63	-0.22
GT-GAN	0.80	**0.66**	0.74	0.48	0.64	0.18	**0.45**	-1.03	0.66	0.07
C-DGT	0.81	0.64	0.82	0.67	**0.84**	**0.71**	0.14	-4.1	0.65	-0.53
Baseline	0.70	0.41	0.72	0.46	0.74	0.51	0.33	-0.75	0.62	0.16
GSEN (Ours)	**0.82**	0.63	**0.84**	**0.71**	**0.84**	**0.71**	0.35	**-0.58**	**0.71**	**0.36**

Table 3: Pearson correlation (PR) and R^2 between predicted and real graphs on real-world datasets



(a) Case study on Subject 121.



(b) Case study on Subject 88.

Figure 2: Qualitative analyses of the source, predicted, and real target graph topologies.

on the graph generative adversarial network. **C-DGT** : node-edge Co-evolving Deep Graph Translator (C-DGT) by (Guo et al. 2019) considers both node and edge attributes that are regularized in the spectral domain. For the datasets without node and edge attributes, the attributes are assigned as all-ones. The parameter settings of GT-GAN, C-DGT, and our method are detailed in supplementary materials, where we also included key parameter sensitivity analyses.

Evaluation Metrics For the effectiveness experiments, the Pearson correlation is computed between the upper triangular values of the normalized adjacency matrix of the real target graph and that of the predicted target graph. For R^2, the higher value it has, the better the performance will be. The mean squared error (MSE) results are also compared for all real datasets. For all comparison and our methods, 5-fold cross-validation is performed, where for each run we select one subset as test and the remaining 4 as training set. In the training set, 20% is randomly selected as validation to determine hyperparameters through a grid search. For the efficiency experiments, as the training time depends on the data and maximum number of epochs for the gradient-based optimization algorithms (e.g. SGD, ADAM), we use per-epoch training time on CPU as the evaluation metric. We use CPUs to make fair comparisons, as the non-deep learning-based comparison methods neither have a GPU-version nor enjoy the speed-up by GPU.

Performance

In this section, the performance of the proposed method, namely GSEN, as well as other methods on effectiveness and efficiency on both 11 synthetic and 4 real-world datasets are elaborated. In addition, the case studies and the sensitivity tests on the real-world datasets are also presented.

Performance on synthetic datasets For synthetic datasets, we compare the Pearson correlation between the target graph generated by various kernels and the graphs predicted by various methods. Table 2 summarizes the effectiveness comparison for 11 synthetic datasets. Our method achieves 0.90 Pearson correlation on average among all 11 synthetic datasets, outperfoming the second best method, namely C-DGT, by around 50%. Also, our GSEN achieves best performance in 8 out of 11 datasets among 14 methods. The traditional graph spectral kernel functions cannot perform well on most of the datasets that do not follow its prescribed graph transformation rules. Their average performance are thus worse than the deep learning-based methods. The deep learning-based graph translation method C-DGT perform much better than the other deep learning-based GT-GAN and fully-connected baseline methods. This is because the C-DGT method partially considers the spectral property as a regularization term such that it can have relatively good performance (e.g. > 0.7) on 7 out of 11 synthetic datasets, but not as good as our GSEN, which typically perform better on normalized Laplacian matrix Z than original Laplacian matrix L. This is because the eigenvalues of the normalized Laplacian matrix are between 0 and 2, which can have a good estimation when using Taylor expansion to estimate $\mathcal{F}(\Lambda)$.

Performance on real-world datasets Here metric-based evaluation, as well as the qualitative analyses on brain network dataset and malware confinement dataset are presented.

Metric-based evaluation: Table 3 shows the Pearson correlation (PR) and R^2 values by comparing the predicted with

	Ours	GT-GAN		C-DGT	
Dataset	time	time	speed up	time	speed up
IoT-20	0.06s	31s	$\times$ 517	2.44s	$\times$ 41
IoT-40	0.09s	66s	$\times$ 733	5.86s	$\times$ 65
IoT-60	0.13s	108s	$\times$ 831	12.10s	$\times$ 93
IoT-200	0.21s	174s	$\times$ 829	40s	$\times$ 190
IoT-400	0.72s	692s	$\times$ 961	-	-
IoT-600	1.60s	1611s	$\times$1007	-	-
IoT-800	2.89s	2964s	$\times$1026	-	-
IoT-1000	4.75s	4112s	$\times$866	-	-

Table 4: Training time per epoch. (-) shows out-of-memory.

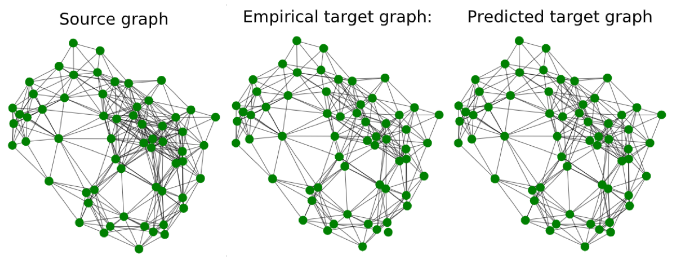


Figure 3: Case study for IOT datasets with GSEN.

empirical target graphs. Our method achieves the highest PR and R^2 on 3 out of 4 datasets, and significantly outperform all the comparison methods by over 8% in PR and more than 0.20 in R^2. Also, MSE results for GSEN outperform all comparison methods on SC-FC dataset, and on average by more than 0.20 for all real world datasets. For the malware confinement, namely the IoT datasets, our method clearly outperforms the C-DGT method in R^2 and PR, which is the state-of-the-art method on these datasets. Moreover, as also shown in the next section, our method is over 40 times faster than the C-DGT method. In addition, the C-DGT method receives low PR and R^2 on the SC-FC translation dataset whose nodes attributes are not available. The GT-GAN method achieves the highest PR and second best R^2 in SC-FC dataset, but performs worse on the other datasets. However, the GT-GAN method is the slowest method in terms of the per epoch training time. SC-FC mapping in neuroscience domain is a very challenging problem and it is not easy for the state-of-the-art (e.g., Galan2008, Abdelnour2014, Meier2016, and Abdelnour2018) in this domain to achieve a PR higher than 0.5. This might be caused by the noise in the resting-state fMRI data. We will show some insightful reason through multiple case studies below.

Qualitative analyses on the brain network SC-FC prediction dataset: Figures 2(a) and 2(b) plot two subjects in: 1) structural connectivity (djacency matrix of the source graph shown on the left column), 2) empirical functional connectivity (adjacency matrix of target graph shown on the middle column), 3) predicted functional connectivity (adjacency matrix of target graph shown on the right column. As shown in Figure 2, the predicted FC using Subject 121's SC is very close to the same subject's empirical FC. On the other hand, the predicted FC using Subject 88's SC is different from Subject 88's empirical FC, although Subject 88's SC is very similar to Subject 121's SC. This is because SC reflects human brain's anatomical neural network, which has relatively less individual differences among the human beings. Unlike SC, the FC used in this datasets reflects the Pearson correlations between two time series (i.e., Blood Oxygen Level Dependent (BOLD) signal) of different brain Regions Of Interests (ROIs), when the subject is instructed under the resting-state. In practice, it is difficult to control these subjects' brain activities, which causes the empirical FC very noisy such that affects the performance of all prediction methods. The additional cases are provided in our supplementary material due to the space limitation.

Qualitative analyses on the malware confinement (IOT) dataset: We observed numerous interesting predictions and exemplified few here and in supplementary materials. Figure 3 demonstrates one case of the source graphs, empirical target graph, and the predicted target graphs by our method from malware confinement datasets. To prevent the network ceased by malware, some of the links in the network are cut while maintaining the optimal functionality of the entire network, which formulates the empirical target graph that is sparser than the source graph. When comparing the empirical target graph with the predicted target graph, it is obvious that our method can mostly predict which link should be cut to prevent the malware propagation.

Efficiency evaluation To validate the efficiency and the scalability of the proposed method, we use three real-world IoT datasets whose number of nodes is from 20 to 60. We further enlarge the IoT-20 dataset from 200 to 1000 nodes, which generates four larger datasets, namely the IoT-200, $\cdots$, IoT-1000 datasets. We report the results in Table 4 for the mean training time per epoch using CPU for 100 epochs on the aforementioned 7 datasets. We compare the results with the two deep learning-based graph translation methods. For our network, we set both the degree of power K and the number of layers to 5. For the other two comparison methods, the default settings are applied. As shown in Table 4, our method is on average 967 times faster than the GT-GAN method and 72 times faster than the C-DGT method. Notice that the C-DGT is unable to handle the graphs with more than 400 nodes due to the out-of-memory error. The scalability of the proposed method is remarkable, which can be trained in 4.75 seconds per epoch on the graphs with 1000 nodes.

Conclusions

This paper focuses on the problem of spectral graph topological evolution, by proposing a novel deep Graph Spectral Evolution Networks (GSEN) which achieves a compelling trade-off between model expressiveness and efficiency. GSEN can solve crucial drawbacks of the existing models in the graph topological evolution domain, which typically suffer from superlinear time and memory complexity. Experimental results on multiple synthetic and real-world datasets demonstrate the outstanding expressiveness and efficiency in terms of the graph topology prediction accuracy and runtime, besides qualitative analyses on the predicted graph topologies.

Acknowledgments

This work was supported by the National Science Foundation (NSF) Grant No. 1755850, No. 1841520, No. 2007716, No. 2007976, No. 1942594, No. 1907805, a Jeffress Memorial Trust Award, Amazon Research Award, NVIDIA GPU Grant, and Design Knowledge Company (subcontract number: 10827.002.120.04).

References

Abdelnour, F.; Dayan, M.; Devinsky, O.; Thesen, T.; and Raj, A. 2018. Functional brain connectivity is predictable from anatomic network's Laplacian eigen-structure. *Neuroimage* 172: 728–739.

Abdelnour, F.; Voss, H. U.; and Raj, A. 2014. Network diffusion accurately models the relationship between structural and functional brain connectivity networks. *Neuroimage* 90: 335–347.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, 3844–3852.

Do, K.; Tran, T.; and Venkatesh, S. 2019. Graph transformation policy network for chemical reaction prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 750–760. ACM.

Erdos, P.; and Renyi, A. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hungary. Acad. Sci.* 5: 17–61.

Galán, R. F. 2008. On how network architecture determines the dominant patterns of spontaneous neural activity. *PloS one* 3(5): e2148.

Guo, X.; Wu, L.; and Zhao, L. 2018. Deep Graph Translation. *CoRR* abs/1805.09980. URL http://arxiv.org/abs/1805.09980.

Guo, X.; and Zhao, L. 2020. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686* .

Guo, X.; Zhao, L.; Nowzari, C.; Rafatirad, S.; Homayoun, H.; and Dinakarrao, S. M. P. 2019. Deep Multi-attributed Graph Translation with Node-Edge Co-evolution. In *he 19th International Conference on Data Mining (ICDM 2019)*, to appear.

Guo, X.; Zhao, L.; Qin, Z.; Wu, L.; Shehu, A.; and Ye, Y. 2020. Interpretable Deep Graph Generation with Node-Edge Co-Disentanglement. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, 1697–1707. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379984. doi:10.1145/3394486.3403221. URL https://doi.org/10.1145/3394486.3403221.

Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* .

Hermundstad, A. M.; Bassett, D. S.; Brown, K. S.; Aminoff, E. M.; Clewett, D.; Freeman, S.; Frithsen, A.; Johnson, A.; Tipper, C. M.; Miller, M. B.; et al. 2013. Structural foundations of resting-state and task-based functional connectivity in the human brain. *Proceedings of the National Academy of Sciences* 110(15): 6169–6174.

Jenkinson, M.; Beckmann, C. F.; Behrens, T. E.; Woolrich, M. W.; and Smith, S. M. 2012. Fsl. *Neuroimage* 62(2): 782–790.

Jin, W.; Yang, K.; Barzilay, R.; and Jaakkola, T. 2018. Learning multimodal graph-to-graph translation for molecular optimization. *arXiv preprint arXiv:1812.01070* .

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Kunegis, J.; Fay, D.; and Bauckhage, C. 2010. Network growth and the spectral evolution model. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 739–748. ACM.

Leskovec, J.; Backstrom, L.; Kumar, R.; and Tomkins, A. 2008. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 462–470. ACM.

Li, C.; Guo, X.; and Mei, Q. 2016. DeepGraph: Graph Structure Predicts Network Growth. *CoRR* abs/1610.06251. URL http://arxiv.org/abs/1610.06251.

Li, R.; Wang, S.; Zhu, F.; and Huang, J. 2018. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Li, R.-H.; Yu, J. X.; and Liu, J. 2011. Link prediction: the power of maximal entropy random walk. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 1147–1156. ACM.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* .

Meier, J.; Tewarie, P.; Hillebrand, A.; Douw, L.; van Dijk, B. W.; Stufflebeam, S. M.; and Van Mieghem, P. 2016. A mapping between structural and functional brain networks. *Brain connectivity* 6(4): 298–311.

Spiegel, S.; Clausen, J.; Albayrak, S.; and Kunegis, J. 2011. Link prediction on evolving data using tensor factorization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 100–110. Springer.

Symeonidis, P.; Iakovidou, N.; Mantas, N.; and Manolopoulos, Y. 2013. From biological to social networks: Link prediction based on multi-way spectral clustering. *Data and Knowledge Engineering* 87: 226–242.

Van Essen, D. C.; Smith, S. M.; Barch, D. M.; Behrens, T. E.; Yacoub, E.; Ugurbil, K.; Consortium, W.-M. H.; et al. 2013. The WU-Minn human connectome project: an overview. *Neuroimage* 80: 62–79.

Wang, P.; Kong, R.; Kong, X.; Liégeois, R.; Orban, C.; Deco, G.; van den Heuvel, M. P.; and Yeo, B. T. 2019. Inversion of a large-scale circuit model reveals a cortical hierarchy in the dynamic resting human brain. *Science advances* 5(1): eaat7854.

You, J.; Ying, R.; Ren, X.; Hamilton, W. L.; and Leskovec, J. 2018. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *ICML*.

Zhao, L. 2020. Event Prediction in Big Data Era: A Systematic Survey. *arXiv preprint arXiv:2007.09815* .