DEEP GRAPH SPECTRAL EVOLUTION NETWORKS FOR GRAPH TOPOLOGICAL TRANSFORMATION

Anonymous authors
Paper under double-blind review

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, there lacks expressive and efficient 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 have demonstrated outstanding performance.

1 INTRODUCTION

Understanding the mechanism of graph generation and evolution has significant importance in many applications, such as brain simulation, mobility network simulation, and social network modeling and intervention. 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 Simonovsky & Komodakis (2018). 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-dimension 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, they require 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 human. 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 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 could include 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.

To address these challenges, this paper proposes a novel end-to-end model named 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 as well as their combination and composition as special cases, and hence are able to fit them with outstanding expressiveness. In addition to this high expressiveness, GSEN is also highly concise in terms of small parameter complexity and time complexity for training. Specifically, the number of parameters and memory complexity of GSEN are independent of the graph size 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 our GSEN over existing deep generative models for graph transformation and models based on graph signal processing. The higher efficiency of GSEN compared to existing deep generative models has also been verified.

2 Related Work

2.1 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 et al. (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 et al. (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), 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 finally applied, where the spectrum of decompositions change, while the eigenvectors stay constant.

2.2 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 & Welling (2016) proposed a localized graph convolutional neural networks (CNNs) based on semi-supervised learning for graph-structured data, where labels are only 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 nodes with labels. 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, the 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 et al. (2015), to solve for large scale classification problems. In Def-

Table 1: Existing kernels for graph spectral translation problem

Kernel Name	Matrix Function	Spectral Function
Laplacian Commute-time Kernel	$\mathcal{K}_{\text{Com}}(L) = L^+$	$U\Lambda^{-1}U^\intercal$, define $\Lambda^{-1}_{i,i} = 0$ if $\Lambda_{i,i} = 0$
Normalized Laplacian Commute-time Kernel	$\mathcal{K}_{\text{Com}}(Z) = Z^+$	$U\Lambda^{-1}U^\intercal$, define $\Lambda^{-1}_{i,i} = 0$ if $\Lambda_{i,i} = 0$
Normalized Adjacent 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 Adjacent Neumann Kernel	$\mathcal{K}_{\text{Neu}}(N) = (I - \alpha N)^{-1}$	$U(I - \alpha N)^{-1}U^\intercal$
Normalized Adjacent 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	$\mathcal{K}_{\text{Reg}}(Z)(I + \alpha Z)^{-1}$	$U(I + \alpha\Lambda)^{-1}U^\intercal$

ferrard et al. (2016), CNNs are presented in the context of spectral graph theory, and fast localized convolutional filters are designed.

2.3 Deep Learning Methods for Graph Transformation Problems

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 et al. (2018), Do et al. (2019)). The prediction in most cases is performed either on the node attributes of the graph or its topology while the other is fixed. Guo et al. (2019) proposes the NEC-DGT (Node-Edge Co-evolving Deep Graph Translator) framework as a novel technique to approach the simultaneous prediction challenge. A portion of these research is only tailored for specific applications and domains (Do et al. (2019)). For example, Do et al. (2019) and Jin et al. (2018) proposed methods only for transferring molecule graphs. Spatio-temporal dependencies in traffic flow are modeled as a diffusion process in a directed graph through a 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. However, until now there is no work in this domain that models the graph topological transformation in spectral domain.

3 Graph Topology Transformation via Spectral Evolution

This paper focuses on a problem of predicting the topology of a target graph based on that of a source graph by characterizing spectral graph evolution.

3.1 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 adjacent matrix that defines the weights of the edges. The adjacent matrix A can be normalized by defining the normalized adjacent 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. Define graph spectrum as $\Lambda \in \mathbb{R}^{|V|\times|V|}$ and graph Fourier basis as U, which are obtained from the eigendecomposition of Laplacian matrix $L = U\Lambda U^\intercal$. A target graph is defined as $G' = (V', E', A')$, where the set of nodes $V' = V$, edges E', adjacent matrix A', normalized adjacent matrix N', the Laplacian matrix L', the normalized Laplacian matrix Z', graph spectrum Λ' and Fourier basis U' are defined the same way as that of source graph.

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 the graph's spectrum, while the graph basis remains the same.

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 listed in Table 1. Several such graph kernels have been empirically demonstrated to model some specific graph process effectively. For example, Kunegis et al. has verified that path count kernels fit very well to the link prediction problem in some email networks (Kunegis et al. (2010)). The evolutions of social networks typically involve 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).

Existing techniques basically first determine the potentially suitable type of kernels and then fit the parameters of the kernels. However, this raises up several 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 human, let alone formulated kernel functions. 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 additive and/or sequential compositions of multiple graph processes described by multiple kernels. For example, the evolution of social network might involve not only triangle closing process (i.e., two friends of a person tend to be friends) by triangle-closing kernels, but also could include 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.

3.2 Generalized Graph Kernels

In order to address the above mentioned challenges, we propose a new generalized graph 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 3.1. *Without loss of generalizability, using graph Laplacian L and L' to represent the graph topology of G and G', the spectral graph translation can be explicitly formulated as $F(L) \rightarrow L'$ by an analytic function F, which is learned by the following equation given $[F(\Lambda)]_{kk} = f(\Lambda_{kk})$:*

$$\min_f \sum_k (f(\Lambda_{k,k}) - U_{\cdot,k}^\intercal L' U_{\cdot,k})^2 \tag{1}$$

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

$$\begin{aligned}
\min_F \|F(L) - L'\|_2^2 &= \|F(U\Lambda U^\intercal) - L'\|_2^2 \\
&= \|\sum\nolimits_{k=0}^{\infty} \frac{F^{(k)}(\gamma)}{k!}(U\Lambda U^\intercal)^k - L'\|_2^2 \quad \text{(power expansion)} \\
&= \|U\left(\sum\nolimits_{k=0}^{\infty} \frac{F^{(k)}(\gamma)}{k!}(\Lambda)^k\right)U^\intercal - L'\|_2^2 \\
&= \|UF(\Lambda)U^\intercal - L'\|_2^2 \\
&= \|U \cdot diag([F(\Lambda_{1,1}), \cdots, F(\Lambda_{|V|,|V|})])U^\intercal - L'\|_2^2 \\
&= \|diag([F(\Lambda_{1,1}), \cdots, F(\Lambda_{|V|,|V|})]) - U^\intercal L' U\|_2^2
\end{aligned}$$

Hence, this problem is equivalent to the problem of $\min_f \sum_k (f(\Lambda_{k,k}) - U_{\cdot,k}^\intercal L' U_{\cdot,k})^2$ given $F(\Lambda)_{kk} = f(\Lambda_{kk})$. The proof is completed. □

We propose the following new generalized graph kernel:

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

Here we introduce some important properties of the proposed generalized graph kernel.

Lemma 3.2. *The generalized graph kernel in Equation 2 has the following properties:*

1. *Existing graph kernels based on L,A,N, and Z such as those in Table 1 are its special cases.*
2. *The additive combinations and compositions of the existing graph kernels are special cases of our operation.*

Proof. Now we prove Property 1. Graph kernels are typically functions of four types of variables, namely Laplacian L, adjacency matrix $A = D - L$, normalized Laplacian $Z = D^{-1/2}LD^{-1/2}$, and normalized adjacency matrix $N = I - D^{-1/2}LD^{-1/2}$. 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 by 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 $k = 0, 1, \cdots$.

Here we prove Property 2. Assume there are two generalized graph kernels $F_a(\Lambda)$ and $F_b(\Lambda)$, then it is easy to see that both their sumation and composition can still be fit by another generalized graph

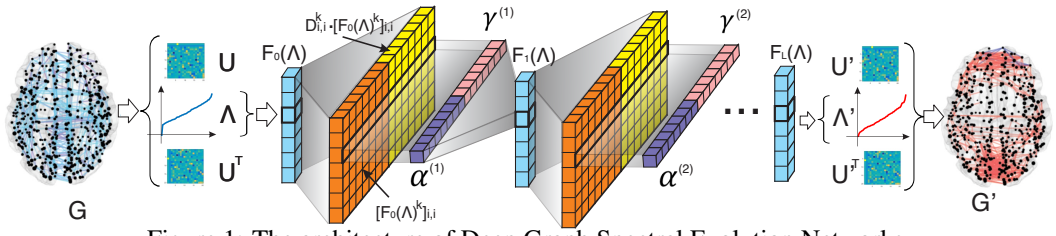


Figure 1: The architecture of Deep Graph Spectral Evolution Networks.

kernel. By leveraging Property 1, the additive and composition of all the various graph kernels such as those in Table 1 can be fit by our generalized graph kernels. □

4 Deep Graph Spectral Evolution Networks

In this section, a new neural network named GSEN, which is based on the proposed generalized graph kernel, is proposed. To achieve this, we reduce the order of the polynomials from infinity to K that is independent of and typically far less than the graph size. Moreover, our GSEN is composed by stacking multiple such generalized graph kernel as a special type of multi-order 1-D convolution operation, as illustrated in Figure 1 and described as follows.

Specifically, each layer can be expressed as follows:

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

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

As shown in Figure 1, we implement GSEN through an L-layer convolution operations from the source graph to target graph. Specifically, the input, namely $F_0(\Lambda)$, is Λ that is the spectrum of the source graph. 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 L-th layer, which outputs the predicted graph spectrum $F_L(\Lambda)$ for the target graph.

Complexity and Efficiency: Training neural network amounts to solve the optimization problem in Equation 1, which can be handled by backpropagation. Our method largely and effectively reduces the number of parameters, to $2 \cdot K \cdot L$, 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 time complexity of $O(K \cdot N \cdot L)$ while the convolution operations involves another $O(K \cdot N \cdot L)$ so the total time complexity of the neural network is $O(K \cdot N \cdot L)$. Also, notice the generation of the input data involves eigendecomposition of L which could be time-consuming for large graph. To address this issue, we can leverage reduced eigendecomposition to only involve the calculation of lower-rank matrix and hence can largely speed up and scalable to large graph.

5 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. All the experiments are 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.

5.1 Experimental Setup

We evaluate the effectiveness on 11 synthetic datasets, and 4 real-world datasets on brain network prediction and malware confinement in the Internet of Things (IoT) task. The datasets, evaluation methods, and comparison methods are elaborated in turn.

5.1.1 Datasets

• **Synthetic Datasets:** In each of the 11 synthetic datasets, we generate 1000 source-target graph pairs. Specifically, first, 1000 unweighted and undirected random graphs with 50 nodes and 200 edges are generated as the source graphs, then each edge in the source graphs is assigned with a

random weight between 0 and 1. Finally, 1000 target graphs are generated by applying one of the kernels depicted in Table 1.

• **Real-world HCP Dataset:** In these datasets, the source and the target graphs respectively reflect the structural connectivity (SC) and the 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). 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 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. Finally, all the 823 pairs of SC and FC adjacent matrices are normalized as defined in Section 3.

• **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 of IoT datasets.

5.1.2 Comparison Methods

The comparison methods include: **Graph spectral transformation kernels:** We compared our GSEN 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 $\mathcal{F} : \Lambda \to \Lambda'$ 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. **Brain network prediction methods:** We consider four classic brain network prediction methods that use SC to FC Galán (2008); Abdelnour et al. (2014); Meier et al. (2016); Abdelnour et al. (2018). Abdelnour et al. (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 et al. (2018) is a newly proposed general-purpose graph topology translation method based on the graph generative adversarial network. **C-DGT** : node-edge Co-evolving Deep Graph Translator (C-DGT) by Guo et al. (2019) is the state-of-the-art deep graph translation network, which considers both and edge attributes that are regularized in the spectral domain. For the datasets without node and edge attributes in our experiments, we simply assign the attributes as all-ones.

5.1.3 Evaluation Metric

For the effectiveness experiments, the Pearson correlation is computed between the upper triangular values of the normalized adjacent matrix of the real target graph and that of the predicted target graph. For all the comparison and our methods, 5-fold cross-validation is performed.

For the efficiency experiments, as the training time depends on the data and the max number of epochs for the gradient-based optimization algorithms (e.g. SGD, ADAM). We use the per-epoch training time on CPU as the evaluation metric for efficiency study.

5.2 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.

5.2.1 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 GSEN method achieves 0.90 Pearson correlation on average among all 11 synthetic datasets. In contrast, the second-best method, namely the C-DGT, only can achieve 0.61 Pearson correlation. The traditional graph spectral kernel functions can achieve

	Dataset											
Method	$\mathcal{K}_{\text{Com}}(L)$	$\mathcal{K}_{\text{Com}}(Z)$	$\mathcal{K}_{\text{Exp}}(N)$	$\mathcal{K}_{\text{Gen}}(L)$	$\mathcal{K}_{\text{Gen}}(Z)$	$\mathcal{K}_{\text{Heat}}(L)$	$\mathcal{K}_{\text{Heat}}(Z)$	$\mathcal{K}_{\text{Neu}}(N)$	$\mathcal{K}_{\text{Path}}(N)$	$\mathcal{K}_{\text{Reg}}(L)$	$\mathcal{K}_{\text{Reg}}(Z)$	Avg.
$\mathcal{K}_{\text{Com}}(L)$	**1.00** (GS)	**1.00** (GS)	0.28	-0.04	-0.15	-0.26	-0.26	0.23	0.26	-0.02	-0.23	0.16
$\mathcal{K}_{\text{Com}}(Z)$	**1.00** (GS)	**1.00** (GS)	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	**1.00** (GS)	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**	**0.93** (GS)	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	**1.00** (GS)	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	**1.00** (GS)	**1.00** (GS)	-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	**1.00** (GS)	**1.00** (GS)	-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	**1.00** (GS)	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**	**1.00** (GS)	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	**0.97** (GS)	**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	**1.00** (GS)	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	<u>0.61</u>
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	0.99	<u>0.80</u>	**1.00**	0.89	0.99	**1.00**	**1.00**	<u>0.71</u>	0.85	**0.90**

Table 2: Pearson correlation between the predicted graph and empirical graph 11 synthetic datasets. Each column denotes the Pearson correlation of the synthetic dataset generated by the kernel function of the second row. Each row denotes the Pearson correlation of the prediction method of the first column. Some results are "gold standard" ones because the predictor and synthetic data generator use the same graph kernel, and hence are marked as "GS" for those results. The right-most column denotes the average Pearson correlation among all 11 synthetic datasets. The highest Pearson correlation in each column/dataset is highlighted in bold font while the second-highest Pearson correlation is marked with an underline.

near-perfect results on the synthetic datasets generated from the same kernel, so all the diagonal Pearson correlation coefficients from the top left corner in Table 2 are close to 1. But none of these kernels perform well on all 11 synthetic datasets, which means they are not generic methods. Their average performance is thus worse than the deep learning-based methods. The deep learning-based graph translation method C-DGT performs much better than the other deep learning-based GT-GAN and fully-connected baseline methods. This because the C-DGT methods partially consider 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 generic spectral method. GSEN typically performs better on normalized Laplacian matrix Z than original Laplacian matrix L, 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 $\mathcal{F}(\Lambda)$.

5.2.2 Performance on real-world datasets

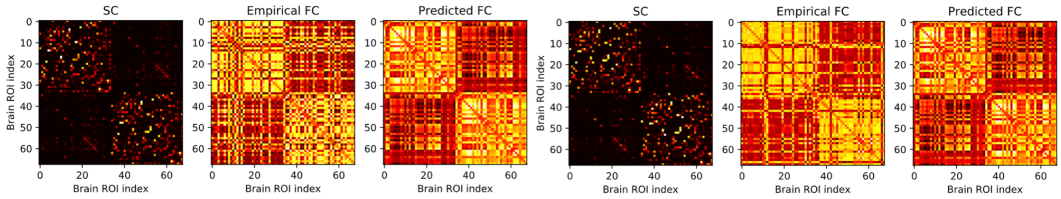
• **Metric-based evaluation:** Table 3 shows the Pearson coefficient by comparing the predicted graphs with the empirical target graphs. Our method achieves the highest Pearson coefficient on 3 out of 4 datasets, and the highest average Pearson correlation among the four real-world datasets. For the malware confinement datasets, namely the IoT datasets, GSEN slightly outperforms the C-DGT method, which is the state-of-the-art method on these datasets. But as we will show in the next section, our method is at least 40 times faster than the C-DGT method. In addition, the C-DGT method receives the lowest Pearson correlation coefficient on the brain network SC-FC translation dataset whose nodes attributes are not available. The GT-GAN method achieves the highest SC-FC dataset, but performs worse than C-DGT and GSEN on other three datasets. The GT-GAN method is also the slowest method in terms of the per epoch training time. None of the methods can achieve the Pearson correlation higher than 0.5 including the top four methods that are exclusively designed for the SC-FC mapping problem in the neural science domain. This might be caused by the noise in the resting-state fMRI data. We will show some insightful reasons through multiple case studies below.

	Dataset				
Method	IoT-20	IoT-40	IoT-60	SC-FC	Avg.
Galan2008	0.74	0.79	0.81	0.23	0.64
Abdelnour2014	0.73	0.76	0.81	0.23	0.63
Meier2016	0.74	0.78	0.81	0.26	0.65
Abdelnour2018	0.73	0.76	0.81	0.23	0.63
GT-GAN	0.80	0.74	0.64	**0.45**	<u>0.66</u>
C-DGT	<u>0.81</u>	<u>0.82</u>	**0.84**	0.14	0.65
Baseline	0.70	0.72	0.74	0.33	0.62
GSEN	**0.82**	**0.84**	**0.84**	<u>0.35</u>	**0.71**

Table 3: Pearson correlation between the predicted graph and empirical graph on real-world datasets

• **Case study on the brain network SC-FC prediction dataset:**

Figure 2 plots two subject's 1) structural connectivity (i.e., the adjacent matrix of the source graph shown on the left column), 2) empirical functional connectivity (i.e., the adjacent matrix of target graph shown on the middle column), 3) predicted functional connectivity (i.e., the adjacent matrix of target graph shown on the right column.



(a) Case study on Subject 121. (b) Case study on Subject 88.

Figure 2: Case studies of the source, real, and predicted target graph topologies on SC-FC dataset.

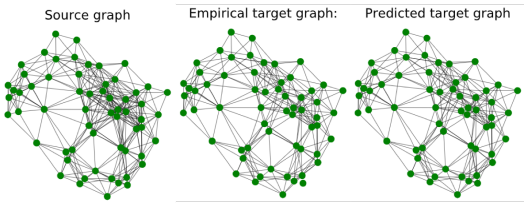


Figure 3: Case studies of the source, real, and predicted target graph topologies on IoT dataset.

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 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 may affect the performance of all prediction methods. The additional cases are provided in our supplementary material due to the space limitation.

• **Case study on the malware confinement (IoT) dataset:** Figure 3 demonstrates the cases of the source graphs, empirical target graph and the predicted target graphs by our method from all three malware confinement datasets. The leftmost source graph in each dataset denotes the original network connections. 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. The rightmost graph is the target network predicted by GSEN using 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 malware propagation.

5.2.3 Efficiency evaluation

To validate the efficiency as well as 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 our GSEN is remarkable, which can be trained in 4.75 seconds per epoch on the graphs with 1000 nodes.

	GSEN		GT-GAN		C-DGT	
Dataset	time	speed up	time	speed up	time	speed up
IoT-20	0.06s	$\times$ 1	31s	$\times$ 517	2.44s	$\times$ 41
IoT-40	0.09s	$\times$ 1	66s	$\times$ 733	5.86s	$\times$ 65
IoT-60	0.13s	$\times$ 1	108s	$\times$ 831	12.10s	$\times$ 93
IoT-200	0.21s	$\times$ 1	174s	$\times$ 829	40s	$\times$ 190
IoT-400	0.72s	$\times$ 1	692s	$\times$ 961	-	-
IoT-600	1.60s	$\times$ 1	1611s	$\times$1007	-	-
IoT-800	2.89s	$\times$ 1	2964s	$\times$1026	-	-
IoT-1000	4.75s	$\times$ 1	4112s	$\times$866	-	-

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

6 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. The proposed GSEN solves 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 demonstrated the outstanding expressiveness and efficiency accuracy in terms of the graph topology prediction accuracy and runtime, as well as qualitative analyses on the predicted graph topologies.

REFERENCES

Farras Abdelnour, Henning U Voss, and Ashish Raj. Network diffusion accurately models the relationship between structural and functional brain connectivity networks. *Neuroimage*, 90:335–347, 2014.

Farras Abdelnour, Michael Dayan, Orrin Devinsky, Thomas Thesen, and Ashish Raj. Functional brain connectivity is predictable from anatomic network's laplacian eigen-structure. *Neuroimage*, 172:728–739, 2018.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pp. 3844–3852, 2016.

Kien Do, Truyen Tran, and Svetha Venkatesh. Graph transformation policy network for chemical reaction prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 750–760. ACM, 2019.

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

Xiaojie Guo, Lingfei Wu, and Liang Zhao. Deep graph translation. *CoRR*, abs/1805.09980, 2018. URL `http://arxiv.org/abs/1805.09980`.

Xiaojie Guo, Liang Zhao, Cameron Nowzari, Setareh Rafatirad, Houman Homayoun, and Sai Manoj Pudukotai Dinakarrao. Deep multi-attributed graph translation with node-edge co-evolution. In *he 19th International Conference on Data Mining (ICDM 2019)*, pp. to appear, 2019.

Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

Ann M Hermundstad, Danielle S Bassett, Kevin S Brown, Elissa M Aminoff, David Clewett, Scott Freeman, Amy Frithsen, Arianne Johnson, Christine M Tipper, Michael B Miller, et al. 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, 2013.

Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.

Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecular optimization. *arXiv preprint arXiv:1812.01070*, 2018.

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

Jérôme Kunegis, Damien Fay, and Christian Bauckhage. Network growth and the spectral evolution model. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 739–748. ACM, 2010.

Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 462–470. ACM, 2008.

Rong-Hua Li, Jeffrey Xu Yu, and Jianquan Liu. Link prediction: the power of maximal entropy random walk. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1147–1156. ACM, 2011.

Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Jil Meier, Prejaas Tewarie, Arjan Hillebrand, Linda Douw, Bob W van Dijk, Steven M Stufflebeam, and Piet Van Mieghem. A mapping between structural and functional brain networks. *Brain connectivity*, 6(4):298–311, 2016.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pp. 412–422. Springer, 2018.

Stephan Spiegel, Jan Clausen, Sahin Albayrak, and Jérôme Kunegis. Link prediction on evolving data using tensor factorization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 100–110. Springer, 2011.

Panagiotis Symeonidis, Nantia Iakovidou, Nikolaos Mantas, and Yannis Manolopoulos. From biological to social networks: Link prediction based on multi-way spectral clustering. *Data & Knowledge Engineering*, 87:226–242, 2013.

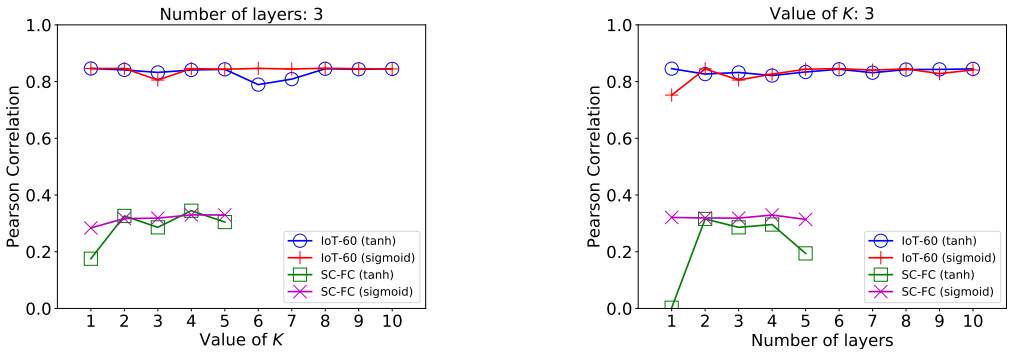
David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

Peng Wang, Ru Kong, Xiaolu Kong, Raphaël Liégeois, Csaba Orban, Gustavo Deco, Martijn P van den Heuvel, and BT Thomas Yeo. Inversion of a large-scale circuit model reveals a cortical hierarchy in the dynamic resting human brain. *Science advances*, 5(1):eaat7854, 2019.

A SUPPLEMENTARY MATERIAL

This is the supplementary material for deep graph spectral evolution networks for graph topological evolution. In this supplementary material, we provide additional experiments on the sensitivity analyses and case study results on the brain network datasets and IoT dataset.

A.1 HYPER-PARAMETER SENSITIVITY TEST



(a) Sensitivity test to the power degree K

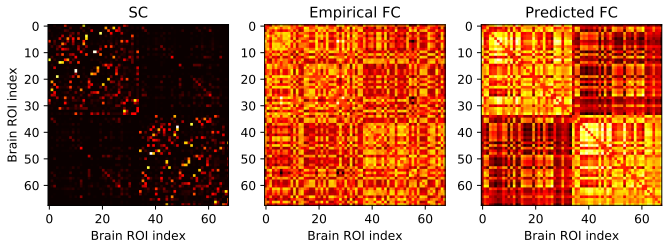
(b) Sensitivity test to the number of layers

Figure 4: Sensitivity analysis

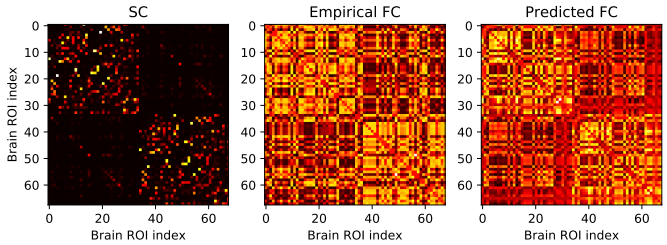
We use two real-world datasets, namely the IoT-60 dataset and the SC-FC dataset to test the sensitivity of the proposed method to its hyper-parameters. The proposed method includes three hyper-parameters: 1) the number of layers, 2) the degree of power K, and 3). the activation function. For the choice of activation function, we exclusively select the sigmoid$(\cdot)$ and tanh$(\cdot)$ function because they can easily constrain the eigenvalues to a valid range (e.g. [0,2) for the normalized Laplacian matrix). The results of the sensitivity test to the power degree K are plotted in Figure 4a, which fix the number of layers to 3. Our method is quite robust for $K > 1$ on both datasets using both activation functions. When $K = 1$, the Pearson correlations are dropped on the SC-FC datasets because of the inaccurate estimation of the Taylor expansion. On the other hand, the sensitivity tests for the number of layers are plotted in Figure 4b. The proposed method is also insensitive to the number of layers as long as the number of layers is greater than one. Recall when there is only one layer, our method is reduced to a single graph spectral kernel that may not be able to handle sophisticated graph translation problems (e.g. the SC-FC prediction problem). Therefore, this result has validated the necessity of the "deep" proposed in this paper.

A.2 ADDITIONAL CASE STUDY RESULTS ON SC-FC DATASETS AND IOT DATASET

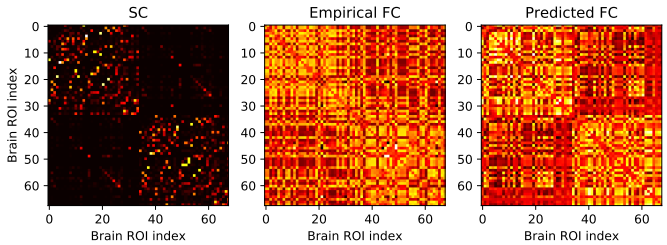
The additional case study results for SC-FC datasets are shown in Figure 5, and the case study results for IoT datasets are shown in Figure 6.



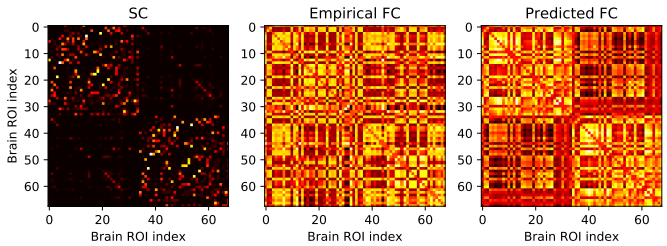
(a) Case study on Subject 6.



(b) Case study on Subject 66.

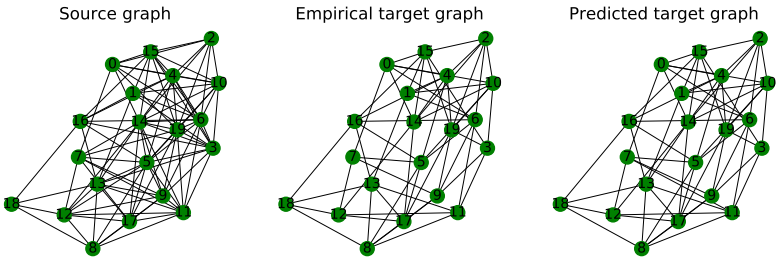


(c) Case study on Subject 72.

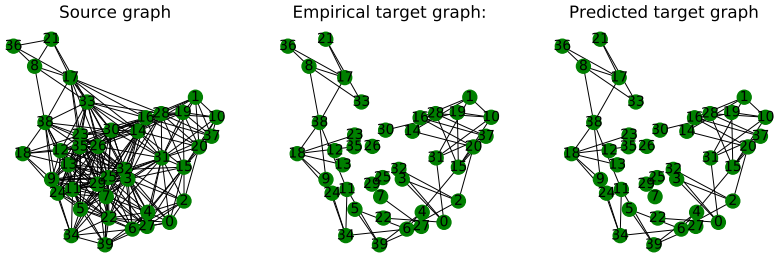


(d) Case study on Subject 123.

Figure 5: Additional case study of brain functional connectivity predicted by our method.



(a) Case study on IoT-20 dataset.



(b) Case study on IoT-40 dataset.

Figure 6: Additional case study of malware confinement datasets with our method.