MODELING AND VARIABILITY ESTIMATION OF NETWORK COLLECTIONS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Dali Guo

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Mario Ventresca, Chair

School of Industrial Engineering, Purdue University

Dr. Joaquín Goñi

School of Industrial Engineering, Purdue University

Dr. Christopher Quinn

Department of Computer Science, Iowa State University

Dr. Thomas Michael Talavage

College of Engineering and Applied Science, University of Cincinnati

Approved by:

Dr. Abhijit Deshmukh

Head of the School Graduate Program

Dedicated to my mom and dad, teaching me critical thinking and giving me unconditional support all the way.

ACKNOWLEDGMENTS

Firstly, I want to thank my PhD advisor and dissertation committee chair, Dr. Mario Ventresca, who always encourages me to choose topic and research questions based on my interests. Without his guidance I would not have made it. The process of thinking and rethinking the philosophy behind the research questions and methodologies makes this interdisciplinary research possible. I want to thank other committee members. Thank you Dr. Joaquín Goñi for bringing me the resource on neuroscience and complex networks. Dr. Christopher Quinn and Dr. Thomas Michael Talavage: your insightful comments and expertise enlighten me about my research with different perspectives.

To all members in Mario's group: it is a great experience to work with you. I learned a lot from you, and our "lunch panel" really contributes many ideas to my work.

I want to thank School of Industrial Engineering - Purdue University which provides a perfect academic and social atmosphere and supports my PhD study for the past years.

To my parents: I am forever grateful for your support and understanding. I know it is a hard decision to let your only child go to a different country for PhD study. I really hope to have the chance of reunion after the pandemic.

Finally, to Janie: it is so lucky to meet you when I started this journey. Thank you so much for your love and support in my life.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABSTRACT

Guo, Dali PhD, Purdue University, December 2020. Modeling and Variability Estimation of Network Collections. Major Professor: Mario Ventresca.

Despite the substantial progress made on the understanding and inference of complex networks, the principles of generative process underlying observed empirical complex systems (e.g. the mechanisms of social network development) remain unclear. Thanks to the rapid development of modern measuring instruments, computational power and storage capacity, massive complex network datasets are now more accessible to researchers, which brings the study of statistical analysis and inference of network collections to the forefront. Most empirical network modeling methods are built on the premise of only a single input network, and the ignorance of inherent variability (e.g. individual difference in brain networks) causes dubious and unreliable models because of the non-deterministic nature of complex system. A clear definition and measure of the inherent variability in a collection of multiple network observations are critical to understand, model, infer and predict the represented complex system. To address the variability on a collection of multiple network observations, this dissertation (1) proposes metrics for measuring variability on a network collection which provides an innovative angle to quantitatively define the distributional properties of a network collection; (2) devises an action-based inferential model on a series of networks that highlights the dynamics of functional connectivity during different states; (3) develops an variational auto-encoder based framework for inferring, synthesizing and predicting a collection of functional brain networks along with the extensibility to other types of network collections.

To measure the variability on a local topological structure level, an information entropy based measure, entropic variability, on graphlets in network collection is

proposed and applied to brain network collections inferred from Human Connectome Project (HCP). The functional brain networks show a higher variability compared to structural brain networks. Moreover, the subject-level variability and temporal variability are measured and analyzed by the entropic variability framework. The emotion processing task causes the most intense reconfiguration (from resting-state) of variability in many functional regions. In addition, an action-based inferential model is devised to model and predict a series of networks. The mechanism of interaction dynamics is hypothesized and inferred under the framework, and the experimental result shows the preference of actions varies according to the belonging community. Limbic system has the most unstable preference while visual and default mode network have very stable preference choice. Furthermore, the variational-autoencoder (VAE) based network generative modeling framework is proposed as a generic model for network collections. The experimental result shows that the basic VAE has the ability to encode a network collection to low dimensional latent variables and reconstruct networks with respect to the inferred latent variables. An extended VAE that utilizes sharable latent variables in common state is capable of disentangling the state-related information and predicting the networks in unseen states.

1. INTRODUCTION

1.1 Empirical Study of Complex Networks

With the rapid increase in computational power, storage capacity and computing devices, the study of large data sets has become ever more popular. Many of these data sets are network representations of complex systems, which are abstract mathematical structures containing vertices and edges, and may include attributes [1]. Network science provides a perspective focusing on the pattern of connections and relationships in complex systems, e.g., communication system [2], human brain [3], transportation [4], chemical reaction and genome expression [5], semantic relationships [6]. The study of real world networks based on disciplines such as graph theory, statistical inference, statistical mechanics, data mining, information theory, and algorithm design sheds light on the understanding and control over the underlying complex system. Moreover, the mathematical and computational tools of network science provide systematic methods of inferring the characteristics and behavior of complex systems, and have been shown to be able to make insightful mathematical predictions about the state and processes on networks. For instance, knowledge of brain network topology can be used to generate predictive models of the spread and functional consequences of brain disease [7, 8]. In a study of the recovery process of stroke patients with attention deficits, it is shown that there exists significant relationship between behavioral improvement and changes in functional brain networks [9]. Moreover, the analysis of structure-function relations at multiple temporal scales shows the potential to predict the cortical dynamics by anatomical connectivities [10].

Although there is no well-established definition for a complex system, according to [11], real-world complex systems tend to be non-deterministic and exhibit non-linear dynamics. However, local stochasticity caused by non-deterministic inter-

actions shows robust organization and memory [11]. Therefore, a single observation of a complex system (and associated network) is a combination of randomness and regularity generated from potentially unknown generative process. From a modeling perspective, the goal is to fit the observed data (networks) for both generalization (measure of randomness) and accuracy (measure of regularity).

Due to the non-trivial structure of large-scale networks [12, 13], most existing statistical methods and tools cannot be directly applied for inference and synthesis. Network scientists have proposed algebraic tools such as spectral graph theory [14, 15] to study the network structure. Its main target is to study the properties of a graph in relationship to the characteristic polynomial, eigenvalues/eigenvectors of matrix representation of the graph such as adjacency matrix and Laplacian matrix [16]. Although the matrix representation is determined by the labeling of nodes, its spectrum and related properties are invariant to different matrix representation. From the perspective of algebraic graph theory, the spectral properties represent the characteristics of network. However, the spectral analysis is irreversible which means the knowledge of graphs' spectrum cannot itself reproduce networks. Hence, it cannot directly answer questions such as "how a network evolves over time" and "what a similar network looks like".

Meanwhile, algorithmic methods with statistical prior (assumption) such as Erdös-Rényi model [17–19], exponential random graph models [20, 21], latent space models [22], etc. derive the network structure by statistical hypothesis, i.e. prior probability distribution of edges, and fit observed data to the proposed structure. Note that the algebraic methods and algorithmic methods are neither isolated nor contradictory, and many network science studies utilize both methods, although the two frameworks are driven by different perspectives. Normally, algorithmic methods are aimed at proposing models to explain the observed networks by inference and synthesizing networks similar to observed networks. However they are struggling to find the balance between generalization and accuracy [23], and most of the time researchers ignore the evaluation of generalization due to a limited number of network

observations (often only a single observation is used as input) [24]. One approach, *dk*-random graphs [23], is proposed to learn the randomness versus regularity by rewiring a network with different constraints. Generally, weaker constraints leads to more randomness and better ability of generalization, and stronger constraints leads to more regularity and higher accuracy to the original network. Although *dk*-random graphs shed some light on the problem of generalization of network models, it is a simplified framework that relies on interdependent degree and subgraph-based properties. When multiple networks are observed, most times *dk*-random graphs are not able to fit the observed networks because the inherent variability in the network collection is not simply based on degree and subgraph-based properties [24]. The algorithmic modeling methods in the past often ignore the variability existing in network collections and as a result infer misleading models that do not match the observed networks [24].

1.1.1 Philosophy

We now have increased ability to observe many networks from a single phenomena and by better understanding the variation in those networks we can create models better suited to synthesizing realistic networks. Therefore, this dissertation advocates that to better study the mechanisms that generate and control a real-world non-deterministic complex system represented as a network, it is necessary to create models from multiple network observations, i.e. a network collection, and its properties. More specifically, quantitatively defined variability of a network collection is a critical statistical property, and generative models that include variability can help us better understand the generative process and mechanism behind the observed networks. A mathematical definition of network collection is as follows:

Definition 1.1.1 *A network collection is a multiset of networks that are generated by the same generative process, i.e., they are independent observations of the same complex system. A network collection of cardinality k is denoted by $\mathcal{G}$ where $\mathcal{G} = \{G_i = (V_i, E_i) : i = 1, 2, \ldots, k\}$.*

The node set and edge set of a network collection can be denoted by $V_{\mathcal{G}} = \{V_i : i = 1, 2, \ldots, k\}$ and $E_{\mathcal{G}} = \{E_i = V_i \times V_i : i = 1, 2, \ldots, k\}$.

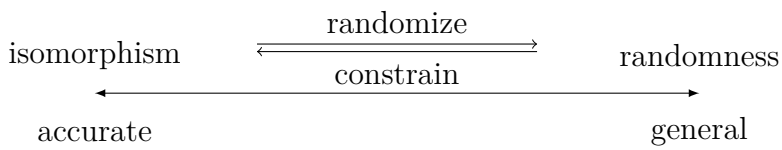


Figure 1.1. Empirical variability measure: accuracy versus generalization in network inferential modeling.

The empirical variability of an observed network collection is a quantitative description of the degree of discrepancy between the networks in the collection[1]. For instance, there are 2 extremes in Figure 1.1. To the left end, we have $G_1 = G_2 = \cdots = G_k$ and there is no variability for the represented system with respect to the observed network collection. To the right end, we have a $\mathcal{G}$ that includes all possible networks with equal chance of getting observed, which has the maximum variability, i.e. ER random graph [19]. However, most real world network collections are in the middle, and quantification of the variability is critical to understand the represented complex system due to the ignorance of variability in past complex network studies and the important role that variability plays in statistics [25, 26].

1.2 Problem Statement

Networks in a collection are high-level objects themselves, which means classic measures of variability such as range, variance, etc. cannot be directly applied to network collection. In order to develop quantify the variability in network collection and potential models that are capable of learning and regenerating the variabilities, we proposed 3 research questions that will be answered in this thesis. Figure 1.2 shows the typical generative modeling procedure from a single network and the extension

[1]Refer to 3 for a formal definition of network collection variability.

to a network collection. Before stating the research questions of this thesis, we would like to first outline the assumptions and scope used throughout this thesis.

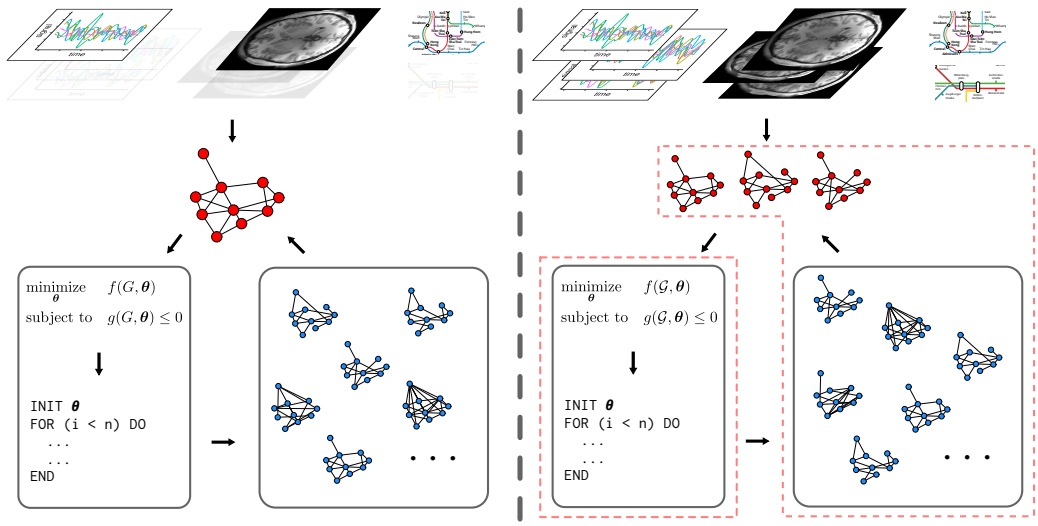


Figure 1.2. Paradigm of inferring generative model for a single network and a network collection. On the left, the connections (e.g. correlation between time series, functional connectivity between brain regions, traffic between stations) are observed and abstracted to a single graph. A model is proposed and fits the network observation. The fitted model can synthesize networks. If the synthesized networks are similar to the observations, we think it is a good model. On the right, when multiple observations of one complex system (e.g. time series in different period, brain connectivity of different subjects, traffic of different transportations between stations) is observed, we get multiple network observations, which implies the distributional properties should be considered for comparing the synthesized network collection and the observed network collection.

1. The research objects (data sets) are network collections, which is composed of multiple networks. Each network is treated as one complete noisy "data point" from the population (See Definition 1.1.1). The networks in a network collection are all simple, unweighted and undirected. Dynamic networks are discrete, synchronous network collections.

2. The networks in a collection are generated by some common generative mechanisms. For instance, the development of brain structure is a product of a

complex series of dynamic and adaptive processes operating within a highly constrained, genetically organized but constantly changing context [27–29].

3. The networks in a collection have the same set of nodes, which means $V_1 = V_2 = \cdots = V_k$ in $V_{\mathcal{G}}$ in Definition 1.1.1. In other words, we are studying networks of fixed set of entities (represented by nodes) and changing interactions (represented by edges).

Based on the assumptions, we are considering the following research questions:

1. *Can we empirically quantify the variability of a network collection?* The observed variability is composed of the known variability caused by individual characteristics and random noise. A quantitative variability measure of networks reveals deeper properties of the network collection rather than the property of individual network independently. Moreover, with the quantified variability, we are more confident on the comparison of synthetic networks (networks in blue) and input networks (networks in red) on the right panel of Figure 1.2.

2. *Can we devise a model that can reproduce the variability of observed network collection?* A model that can reproduce the variability of observed network collection is able to generate networks that have similar properties to the observed network collection in Figure 1.1. On the right panel of Figure 1.2, we want to ask can we devise a generative model that is capable to replicate the property of input network collection.

3. *Can we map the variability of network collection to interpretable features?* To extend the model devised in Question 2, we presume the level of variability is related to observable features of the network collection. For instance, functional brain networks are more diverse in cognition task than in resting-state. Based on this assumption, we could devise model that can synthesize networks with specific feature.

1.2.1 Practical Significance of Research Questions

Variability plays a critical role in analysis, classification and generation of high dimensional data. For instance, principal component analysis (PCA) is wildly used for biomedical images classification [30, 31]. Research question 1 is raised to explore potential quantification of variability of a network collection. The quantified variability is not only an important characteristic in complex network analysis, but also a possible objective in optimization. Generative network models that take a single input network and propose theoretical hypothesis, e.g. it is assumed that every possible edge occurs independently with a fixed probability in ER random graph, have difficulties on explaining the observed non-deterministic complex networks in real world [24]. Since networks are high-level non-Euclidean data objects that cannot be measured by standard and acknowledged measurement [32], it is necessary to formulate a framework for answering questions such as, "what is the median of a network collection?", "is this network collection more diverse than another network collection?", "what feature contributes the most to network diversity?". Among all typical statistical questions, the study of variability has shown its importance by its extensive applications in statistical methods such as t-test, PCA and deep learning [30, 33]. Although these statistical methods cannot be directly applied to networks, the philosophy of utilizing variability to study statistical objects sheds light on the study of network collections. This dissertation is aimed at showing potential ways of quantitatively studying the variability of the networks.

Based on the answer to question 1, research question 2 is raised to devise models that are able to reproduce the variability for a collection of networks. As we discussed in Section 1.1, the importance of the distributional property in network population estimation is emphasized by devising a model that matches the variability. Furthermore, we can synthesize network collection that has similar distributional properties to the observations by the devised model. A well-implemented network model can describe the network population with extracted representation (e.g. parameters). In

addition, the synthesized network samples will have similar characteristics as the real world networks, which saves time and power for collecting data.

Last but not least, if we know the variability is related to the population characteristics, is it possible to isolate the variabilities into interpretable and controllable features? We claim that the bridge between observed features/factors of network collection and parameters in network model 1) contributes to the understanding of the generative mechanism; 2) facilitates the modeling accuracy by taking the extra information of the interpretable features; 3) extends the ability to synthesize networks with specific features.

For instance in functional brain network research, if we know the brain networks have distinct level of variability for different states, is it possible to reproduce accurate network collections for a specified state? Given partial information of the functional brain networks of a subject, is it possible to predict the networks in untested tasks? Unveiling the mapping between variability and interpretable features could give merit to the understanding of the generative mechanisms and synthesizing desired network collections.

1.3 Overview of the Thesis

After the scope of this research is defined and the problems to be answered are stated in Chapter 1, previous researches related to network collections and network inferential modeling are introduced in Chapter 2. In Chapter 3, measures of variability are proposed and tested with various data sets. In Chapter 4, a temporal action-based model for network series is proposed and implemented with human brain network data from HCP. Based on the work in Chapter 3 and 4, a generalized variational auto-encoder model is proposed in Chapter 5 to learn the network collection observation and reproduce the observed network variability.

2. BACKGROUND

The mathematical notation of a network collection in Definition 1.1.1 is similar to that of a multilayer network, and many analysis tools for multilayer networks can be used in network collection modeling as introduced in Section 2.1. The main difference between multilayer networks and a network collection is that for multilayer networks we are interested in the topological properties of connections within and across layers of a single observation, while the networks in network collections are seen as independent samples of a complex system (which could be multilayer networks as well). In Section 2.3, the concept of a network ensemble, a statistical mechanics representation of a random network, is reviewed. It is an concise and elegant framework for studying the statistical properties of network collection but it is oversimplified for networks in the real-world. In Section 2.4, we discuss the relationship between previous research progress on network collection and the gaps we aim to fill.

2.1 Multilayer Networks

The study of multiple networks starts with the study of the heterogeneous features of nodes and edges. In contrast to simple graphs where nodes and edges are homogeneous, networks with multiple types of connections provide more information for studying the real-world system. For example, in [34] a social network containing 14 individuals having 6 different types of connections was observed: participation in horseplay, participation in arguments about open windows, friendship, antagonistic (negative) behavior, helping others with work and the number of times workers traded job assignments. These 6 types of connections are 6 types of edges in the networks therefore giving 6 different layers of edges with same set of nodes.

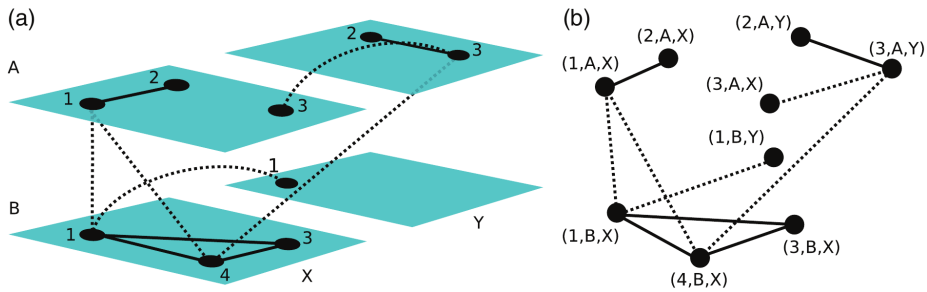


Figure 2.1. (a) An example multilayer network with 4 nodes and 4 layers. There are two dimensions of the layer, $\{A, B\}$ and $\{X, Y\}$, which gives the four layers: $(A, X), (B, X), (A, Y), (B, Y)$. The solid and dashed lines represent inner-layer and inter-layer edges respectively. (b) The network in (a) can be flattened into a single network where the non-isolated nodes are populated with the layers they belong to. The adjacency matrices can be concatenated likewise. This figure is cited from [35].

A general form of a multilayer network is a quadruplet $M = \{V_M, E_M, V, \boldsymbol{L}\}$ where V is the set of nodes (the same as single-layer network), $\boldsymbol{L}$ is the sequence of layers. $V_M = V \times L_1 \times L_2 \times \cdots \times L_d$ and $E_M = V_M \times V_M$ [35]. Self-edges and multi-edges are disallowed for simplicity[1]. A tensor representation of an unweighted multilayer network is a rank $-2(d+1)$ adjacency tensor $\mathcal{A} \in \{0,1\}^{|V|\times|V|\times|\boldsymbol{L_1}|\times|\boldsymbol{L_2}|\times\cdots\times|\boldsymbol{L_d}|}$ and the tensor representation can be flattened into a single-layer network as shown in Figure 2.1 [35]. Multilayer networks of such form are typically studied with tensor-decomposition methods [36, 37] and multiway data analysis [38]. Based on the basic form of multilayer networks, one can categorize common multilayer networks by their properties and constraints. Table 2.1 shows some typical multilayer networks with properties and constraints. Due to the debates and unified term usage, the constraints listed here are for reference only and some constraints are often relaxed for specific methods or context. For instance, the dynamic networks in [39] have non-diagonal couplings for the dynamics between neighbors. The categorization in Table 2.1 is not strict and many networks can be converted between types depending on research

[1]It can be relaxed by constructing new layers to include self-edges and multi-edges but it will bring interpretation issue of layers due to the heterogeneous definition of layers.

purpose. For example, the dynamic networks are often modeled as multiplex networks if the ordinal layers constraint is imposed.

Table 2.1. Examples of multilayer networks with properties and constraints. A multilayer network has *Disjoint layers* if the node sets of layers are disjoint. A multilayer network has *Independent layers* if there is no dependency relations between layers. A multilayer network has *Diagonal couplings* if inter-layer edges only exist between the two nodes represent the same entity in different layers. A multilayer network has ordinal layers if the layers are ordered by some sequence and inter-layer edges only exist between adjacent layers.

Name	**Disjoint layers**	**Independent layers**	**Diagonal couplings**	**Ordinal layers**	**Example references**
Interdependent networks	✓	×	×	×	[40]
Interconnect networks	✓	✓	×	×	[41–44]
Network of networks	✓	✓	×	×	[45]
Multiplex networks	×	×	✓	×	[46–48]
Dynamic networks	×	×	✓	✓	[49–51]

Based on the basic form and constraints, we can implement the multilayer network analysis to real-world complex systems. For instance, [40] shows an interdependent network application of a coupled system of electrical grid and communication network in Italy. In this example, it has a electrical grid layer that has nodes representing power stations and edges representing connection between power stations. Similarly, the other layer, communication layer, has nodes represent servers and edges represent connection between servers. Additionally, every server is considered to be connected to the nearest power station. The robustness of such coupled system is analyzed and the critical point of average degree is identified. An example of interconnect networks is a network of sexually transmitted epidemic in which two layers are heterosexual and homosexual networks of sexual contacts. And there exists bisexual individuals who connect the two layers [42].

Generally, the measurements of topological structure devised for single-layer networks can be applied to multilayer networks by taking each layer as separate network and aggregating the result by the cost of changing layers [44, 52–55]. Similarly, network models such as ER model [18, 19] and configuration model [56] can be easily

adapted to multilayer networks by modeling the layer independently or considering the layer property such as layer joint degree distribution [41]. Moreover, there are successful adaption of ERGM for multilayer networks as well [57, 58].

2.2 Information Entropy

In statistics, the variability of a given distribution is often measured by some scale parameters such as standard deviation, range, etc. However, these dimensional variability measures are less legitimate for discrete distribution because discrete variables are dimensionless. Information entropy and other information measures are widely used for measuring the variability of discrete variables. In this section the concept of information entropy is reviewed as background knowledge.

In information theory, an event is more informative when it happens with lower probability because more bits are needed to code the event. Consider a discrete random variable X and x is an observation of X. The information carried by x is defined as

$$I(x) = -\log P(x). \tag{2.1}$$

The expectation of information for X,

$$H(X) = \mathbb{E}[I(x)] = -\sum_{x \in X} p(x) \log p(x), \tag{2.2}$$

is known as the information entropy of X. $H(X)$ measures the degree of variability and uncertainty of the random event X.

2.3 Network Ensembles

A network ensemble is a probability distribution over many possible networks [26]. In other words, network ensemble specifies the probability that one network is observed and the probability can be explicitly expressed. Theoretically most network

ensembles can draw inference from the observed network collections and synthesize network by sampling from the inferred distribution. As we discussed in 1.1, the distribution of a set of networks is one form of network model and most of the models are ensemble models [26]. The study of network ensembles reveals many topological properties by mathematical derivation. For example, the ER model $G(n, p)$, in which edges between any 2 nodes out of the given n nodes are independently and uniformly drawn with probability p, can be expressed as a multivariate Bernoulli distribution. Therefore the degree distribution follows a binomial distribution. The entropy of a network ensemble is

$$H(G) = -\sum_{G \in \mathcal{G}} P(G) \log P(G) \tag{2.3}$$

where $P(G)$ is the probability of a specific G is observed and $\mathcal{G}$ is the set of all possible networks[2]. For instance, the entropy of $G(n, p)$ is the joint entropy of all edges

$$H(G) = \sum_{i<j} H(E_{ij}) = \frac{n(n-1)}{2} H(p) \tag{2.4}$$

where $H(p) = -p \log p - (1-p) \log(1-p)$.

Furthermore, the entropy of other network ensembles such as the configuration model and other constraints (given degree correlation, community structure, distance, etc.) is studied in [59, 60]. The researchers quantify the variability of network ensembles and explain the relationship between network variability and a hypothetical constraint. For instance, [59] proves that ensembles with fixed scale-free degree distribution have lower entropy, i.e. lower variability, than the ensembles with homogeneous degree distribution, which means power law network ensemble is located to the left of the network ensemble of homogeneous degree in Figure 1.1.

In [61,62], the principle of maximum entropy argues that with limited prior knowledge, the distribution that maximizes the entropy is the distribution that best represents the current state of knowledge. When applied to network ensembles, it is

[2]It is replacing the X in 2.2 with G if you consider a network ensemble is a random variable

proved that ERGM is the distribution of networks that maximizes the entropy subject to known constraints [26]. In ERGM the probability of a network is

$$P(G) = \frac{e^{-\boldsymbol{\theta}\boldsymbol{m}(G)}}{Z} \tag{2.5}$$

where $\boldsymbol{m}(G)$ is the vector of constraints to match and $\boldsymbol{\theta}$ is the corresponding parameter (Lagrangian).

2.4 Relevance and Gaps

As stated in Section 1.1 and 1.2, this thesis targets the analysis and modeling on multiple observations of a complex system. There are many similarities between Definition 1.1.1 on network collection and the definition of multilayer network. However, multilayer network framework formulates the problem of the heterogeneity in network observations by layering. It emphasizes how the inter-layer edges impact the topology and dynamics in network. Meanwhile, our study on network collection has no explicit definition of inter-layer edges. Theoretically, a network collection can be categorized into a multilayer network with common set of nodes, interdependent layers and diagonal couplings, although the topology of intra-layer and inter-layer edges are off-topic with our statistical point of view. Layers in multilayer networks are treated as characteristics and features of the system, while samples in network collections are treated as observations of a system with stochasticity.

The variability of network ensembles is successfully studied in [59, 60], thanks to the delicate statistical formulation of network ensembles. The entropy of network ensembles quantifies the variability of statistical network models with constraints. It is possible to compare the randomness of models and constraints in the information theory perspective. However, the network ensembles are developed with hypotheses that are not for inference purpose. For instance, Barabási-Albert (BA) model is based on a strong assumption that new node is connected to existing nodes with a probability that is proportional to degree of the node. Then some theoretical

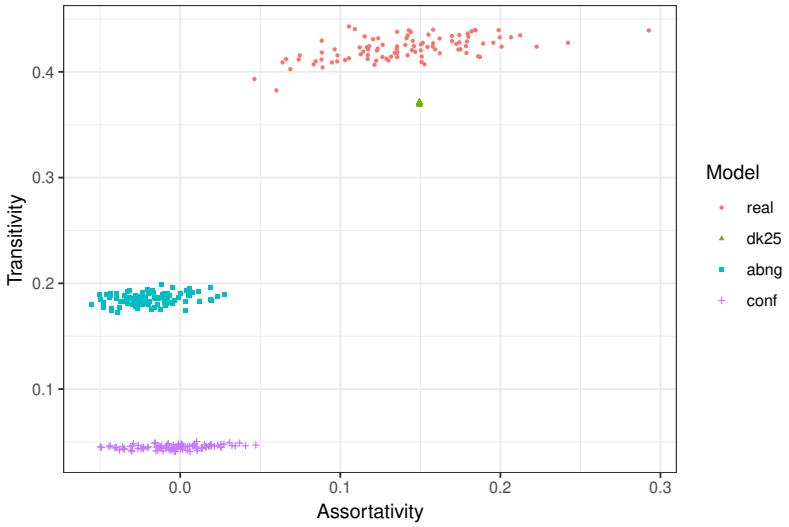


Figure 2.2. Distribution of global network properties assortativity and transitivity of network collection of 100 structural brain networks and synthesized network collections. The inability of network models to replicate the distributional properties of the original process highlights the need for devising better techniques for training and evaluating models using network populations.

properties (e.g. power law distribution of degree) are derived based on the model. Sometimes observed empirical networks follow the theoretical properties sometimes they do not [63]. Therefore the ensembles start with a strong assumption instead of empirical observations, and they struggle with inferring real-world network observations due to limited knowledge on prior parameter assignment [64, 65]. In order to assess the randomness in empirical networks, *dk*-random graphs [23] is proposed with a similar philosophy that constrains on network ensembles can quantify the variability of a set of networks. The experiment of *dk*-random graphs shows higher order of degree distribution constraint shrinks the network space and consequently reduces the variability of network ensembles. However, entropy of network ensembles and *dk*-random graphs are not proposed for empirical network collections. The variability in these statistical models are either a fuzzy hypothetic prior (ERGM) or specified constraints (*dk*-random graph). In either case, when applied to real-world network collections, they struggle to match the theoretical analysis with observations [24].

An example in Figure 2.2 illustrates the performance of some generative network models that are proposed to model one single network: *dk*-2.5 [23], ABNG [66] and configuration model [56, 67]. The 100 structural brain networks (labeled as "real" in Figure 2.2) are processed as described in [68]. As depicted in Figure 1.2, we typically observe a single network from the population and ideally would like a generative model to be capable of generating similar network samples. Consequently, one network from the population was selected at random as input for training all 3 models, which were then used to synthesize populations of 100 networks each. Finally, we compare the distribution of degree assortativity and transitivity in the 4 network collections in Figure 2.2. We observe that the networks synthesized by *dk*-2.5 all have exactly the same global network properties. While there is no variability in these properties, a simple comparison with the observed network might lead to a conclusion that the model aptly describes the generative process, which can prove to be highly misleading. On the other hand, the population of networks synthesized by ABNG and configuration shows more variability, but fails to match the network properties of the original population (red dots). Therefore we need a non-parametric method for measuring the variability of any observed network collection.

There are few investigations that focus on the distributional property of empirical network collection. For instance, in [69], the distribution of degree, clustering coefficient and path lengths of 6 empirical network and synthetic networks simulated by Kronecker model and ERGM are analyzed. Later in [70] an extension of Kronecker model is proposed to fixed the low variance in original Kronecker model. The analysis on network population starts to emphasize the natural variability of network metrics in network collection such as degree distribution, clustering coefficient, etc. and nevertheless show the qualitative comparison instead of a quantification of the variability. Although in [69, 70] the generative network models take a network collection as input, only one network out of all networks in a collection that is closest to the median network is used as input in synthetic process.

2.5 Dataset: Brain Networks

The datasets used in most of the experiments in this dissertation are functional brain networks inferred from fMRI images in Human Connectome Project (HCP, http://www.humanconnectome.org/). More specifically, the images are firstly processed by [71] that includes 100 unrelated subjects from the HCP 900 subjects data release [72]. The resting-state fMRI data for each individual was acquired over 2 separate days with 2 sessions per day and there are 7 task sessions in which task fMRI data was acquired for each subject with 1 session per task. Detail of scan sessions are shown in Table 2.2. The repetition time (TR) and echo time (TE) are 720ms and 33.1ms respectively. The HCP functional preprocessing pipeline was employed for the fMRI data used here [73–78]. In addition, a bandpass first-order Butterworth filter in forward and reverse directions [0.001Hz, 0.08Hz] and [0.001Hz, 0.25Hz] was applied to the resting-state fMRI data and task fMRI data respectively [68, 79]. The processed blood-oxygen-level-dependent (BOLD) signals reflects the changes deoxyhemoglobin driven by localized changes in brain blood flow and blood oxygenation [80]. Most fMRI research based on BOLD imaging takes the advantage of high spatial resolution images of brain which helps identify localized fluctuation and activity [81]. However, unlike electroencephalogram (EEG) which directly measures the electrical activity, BOLD measures the the neural activity indirectly by haemodynamic response which is relative and not individually quantitative. Although fMRI and underlying BOLD signals have limitations, they are believed to be one of the most trustworthy measures of brain activity [82]. The the cerebral cortex was parcellated into 360 ROIs (Figure) with a multi-modal parcellation as proposed by [83]. This parcellation utilizes a machine-learning classifier that can recognize the multi-modal "fingerprint" of individuals which is a precise and robust parcellation map across individuals.

Functional brain networks are inferred from the preprocessed BOLD signals as shown in Figure 2.4. The temporal Pearson correlation[3] between BOLD signals of

[3]Pearson correlation is used for simplicity. Other measures such as partial correlation and Bayes net models are possible alternatives [84].

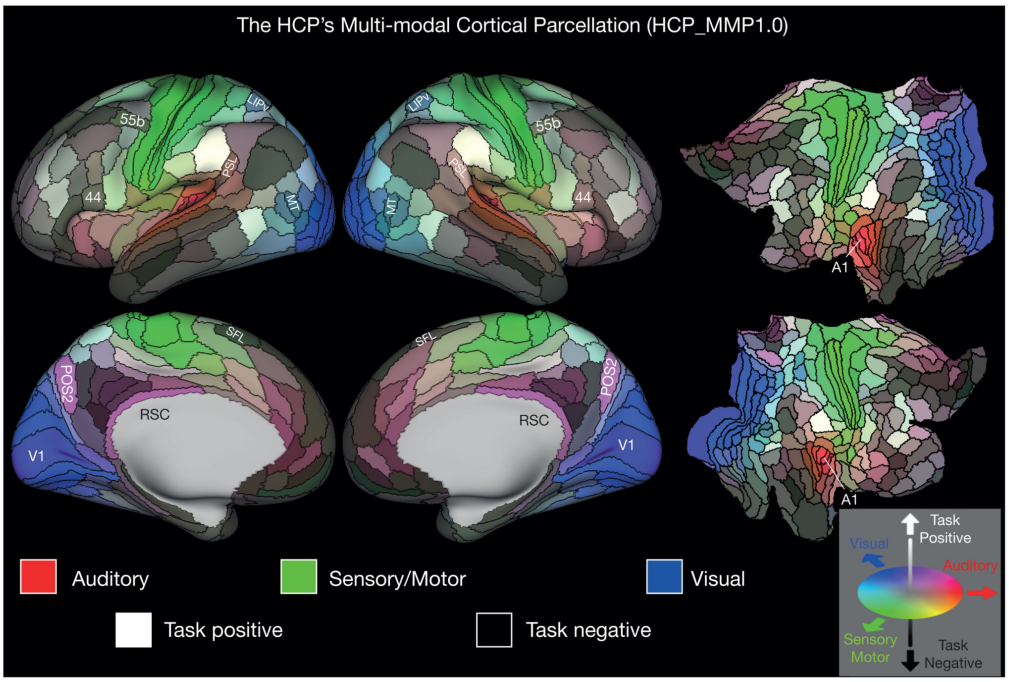


Figure 2.3. The HCP's multi-modal parcellation by [83]. The 180 ROIs in both left (top row) and right (bottom row) hemispheres on inflated (column 1 and 2) and flatten cortical surface (column 3).

Table 2.2. Format of Preprocessed BOLD data. Each task session is replicated twice (resting-state is replicated 4 times).

Condition	Number of runs	Frame per run	Run Duration (min:sec)
Resting-state	4	1200	14:33
Working Memory	2	405	5:01
Gambling	2	253	3:12
Motor	2	284	3:34
Language	2	316	3:57
Social Cognition	2	274	3:27
Relational Processing	2	232	2:56
Emotion Processing	2	176	2:16

different ROIs indicates the functional connectivity between ROIs. Then the correlation matrix is then binarized by fixed threshold[4] as adjacency matrix for functional brain networks in which ROIs are represented by nodes and binarized correlations are represented by edges.

[4]The global fixed threshold is implemented here for simplicity. Other methods such as network fragmentation (e.g. first minimum spanning tree then add edges to desired density) are all possible alternatives.

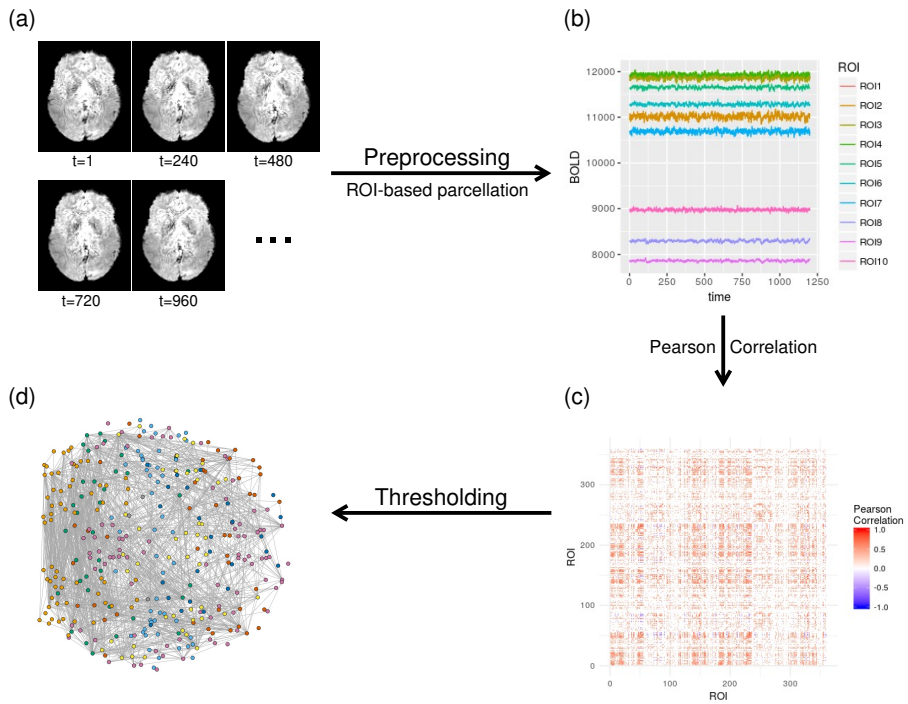


Figure 2.4. The process of functional brain network inference. (a) Preprocessed fMRI images in different time stamp. (b) Original cerebral cortex are parcellated into 360 ROIs with a multi-modal parcellation by [83]. (c) Calculating the pairwise Pearson correlation gives a correlation matrix of the ROIs. (d) The real-valued correlation matrix is binarized into a symmetric binary matrix that serves as the adjacency matrix for an undirected functional brain network.

3. MEASURES OF VARIABILITY ON A NETWORK COLLECTION

In this chapter, a general form of variability on a network collection is defined with proposed conditions. Then the variance of global network metrics and pairwise dissimilarity are reviewed and analyzed. Moreover an information entropy based framework, entropic variability, is introduced as an innovative way of measuring the variability of network collections with respect to local structures. An experiment of comparing all the introduced measures is conducted with the data set of stock price correlation network, structural brain network and functional brain network. In addition, the variability of functional brain networks that caused by individual difference and task reconfiguration is analyzed with the proposed entropic variability.

3.1 Definition of Network Collection Variability

Given a network collection $\mathcal{G} = \{G_i = (V_i, E_i) : i = 1, 2, \ldots, k\}$ and the node and edge sets, $\mathcal{V} = \{V_i : i = 1, 2, \ldots, k\}$ and $\mathcal{E} = \{E_i : i = 1, 2, \ldots, k\}$, a function $\sigma : \mathcal{G} \to \mathbb{R}$ is defined as a measure of variability on $\mathcal{G}$ where the following conditions are satisfied:

1. $\sigma(\mathcal{G}) \geq 0$.
2. $\sigma(\mathcal{G}) = 0 \;\Leftrightarrow\;$ There exists $\bar{G}$ such that $G_i = \bar{G}\ \forall\ i \in \{1, \ldots, k\}$.
3. Suppose a mapping that coerces all node sets into one node set $f : \mathcal{V} \to \bar{V}$ and the edge coercion mapping $h|f : E \to E'$, the variability $\sigma(\mathcal{G}) = \sigma(G'_i = \{(\bar{V}, E'_i) : i = 1, 2, \ldots, k\}) + \phi(\mathcal{V})$ where $\bar{V} = f(\mathcal{V})$, $E'_i = h(E_i)|f(\mathcal{V})$ and $\phi(\mathcal{V})$ is the variability of node set $\mathcal{V}$.

Condition 1 indicates that variability is non-negative. Condition 2 indicates that the variability of a network collection equals zero if and only if all networks in the collection are identical. In condition 3, the variability of a network collection is divided into two components, the variability of node sets $\sigma(\mathcal{V})$ and the variability of the transformed networks that are regularized by f and $h|f$. For instance, if the networks in the collection have different number of nodes and nodes are unlabeled, one can merge similar nodes by clustering methods [85, 86] to compress all networks to same size and then label the nodes by nodal attributes such as the sorted degree sequence.

For simplicity, we assume that all networks in a collection have the same set of nodes, i.e., $V_1 = V_2 = \ldots, = V_k = V$. Therefore the network collection is defined as $\mathcal{G} = \{G_i = (V, E_i) : i = 1, 2, \ldots, k\}$. This assumption applies when the real-world complex system has identical nodes but varied edges. For instance, the brain of any human can be regularized into a common template where brain regions are consistent for different individuals [87]. However, the pattern of brain physical and functional connections likely vary per individual [88, 89]. The difference could also result from temporal factors, for example the series of coauthorship networks [90] of the same authors are changing over time.

In the following sections, variability of network collections are examined from three perspectives: variance of network metrics, in-collection dissimilarity and the information entropy carried by network components.

3.2 Variance in Network Morphospace

Complex networks are high dimensional data with non-trivial topological structure. To study the variability in a network collection, one can map networks in a collection into a metric space of topological properties of interest, e.g., assortativity, transitivity, modularity, etc [91, 92]. That metric space is a "network mophospace" whose axes represent specific network traits and each point in the space represents

one network [91]. Networks that are close in the morphospace share some common "morphological" characteristics. Therefore variance of the metrics could indicate the morphological variability of network collection. For instance, networks generated by simulation are plotted in a morphospace of communication efficiency in Figure 3.1. If networks in the network morphospace have lower variance then they are considered to have low variability. As a result the characteristics of networks are extracted independently. This method measures $\sigma(\mathcal{G})$ by measuring the variance of samples $\{\boldsymbol{m}(G)|G \in \mathcal{G}\}$ where $\boldsymbol{m}(G)$ is a vector of real-valued metrics of graph G (e.g. $||E_{\text{diff}}||$ and $||E_{\text{route}}$ in Figure 3.1.) Since variance is greater than 0 and equal to 0 when all samples are identical by definition, condition 1 and 2 are satisfied.

However, measuring the variability of a network collection by the variance in this morphospace has limitations. First, this measure depends on the choice of network metrics. Without prior knowledge of the studied object, there is no legitimate way to determine the metrics, and arbitrary selection of metrics could lead to misleading conclusions [93, 94]. Secondly, the selected metrics are often used as constraints in a network generative model, which will constrain the distribution of metrics and consequently lose the model generalization ability. For example, networks generated from dK-random graphs enforce exact joint degree distribution as the input network and therefore the assortativity of degree of generated networks is identical to that of the input network and has zero variance. Lastly, the changes in local structure are ignored because each network is measured independently and the distributional properties are based on independent measures. For instance, in the network collection, edges in one community rewire to another community but the density does not change. In the space of network density, the networks have zero variance but there does exist density variation in local structure.

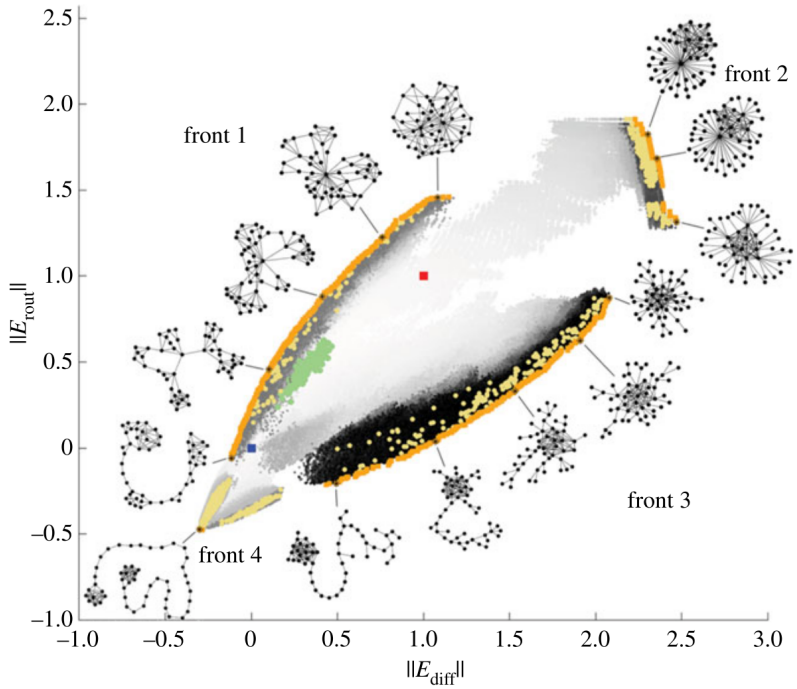


Figure 3.1. Communication-efficiency morphospace from [91]. Each point represents a network generated by optimization algorithms. The $||E_{\text{diff}}||$ and $||E_{\text{route}}||$ represent the efficiency of communication through random walk and shortest path respectively. Green points indicate the location of the initial network collection; blue and red points indicate the location of the lattice and random networks with respect to the initial network collection, respectively; orange points show the location of the final collection of networks generated by simulations with different objectives. Figure is cited from [91].

3.3 Pairwise Dissimilarity

Quantifying the topological distance or dissimilarity between a pair of networks is a popular approach in the network science community due to its importance in many applications such as protein ligand docking, video indexing and computer vision [95,96]. A straight-forward idea is to measure the distance between adjacency matrices by methods such as dissimilarity cross entropy, Jensen-Shannon distance (JSD) [97], mean square error (MSE), etc. Note that this method relies on the assumption that networks have the same set of nodes. A more general method for networks with varied node sets is to measure the distance between distributional properties such as degree distribution, centrality distribution, etc. For instance, the JSD between two degree distributions measures the discrepancy of degree between the pair of networks. Besides, there are similarity measures on networks based on graph matching such as subgraph isomorphism [98]. Therefore one can measure the variability of the network collection by the pairwise dissimilarity between networks. Generally, higher overall dissimilarity indicates larger variability. The idea is to measure $\sigma(\mathcal{G})$ by measuring the pairwise dissimilarity $\{d(G_i, G_j) : i, j = 1, 2, \ldots, k\}$ where $d(G_i, G_j)$ is a dissimilarity measure between two networks G_i, G_j which satisfies $(G_i, G_j) \geq 0$ and $d(G_i, G_j) = 0$ if $G_i = G_j$.

With the pairwise dissimilarity one can embed the networks into lower dimensional space with ordination techniques such as multidimensional scaling (MDS), and get the distribution of network as well as the variability of network collection. Pairwise dissimilarity exploits pairwise dependence between networks while methods of statistics of global measures ignores the dependence. However, the computational cost of pairwise dissimilarity increases quadratically with the grow of network collection size[1], while making a single comparison is very computational costly [99].

[1]For a network collection consists n networks, we need to make $\frac{n(n-1)}{2}$ comparisons.

3.4 Information Entropies

Since network models are probability distributions as well, the entropic measure of variability of network models can be calculated as Equation 2.2 where x is an network observation and X is the random variable for all possible network observations [26]. With real-world network data, the probability $P(X = x)$ can be approximated by the empirical probability. However, the number of distinct networks increases exponentially when number of nodes increases and therefore the observations are too sparse in the network space to approximate empirical probability well. One solution is to decompose the network into localized structures and study the entropy of local structure over all observations such as graph motifs and graphlets. The "decompose and measure" is applied to single network [100–102] but has not been applied to network collection. In this section, two specific localized entropy measures are proposed and analyzed.

3.4.1 Localized Entropy Measures

For a network collection with a fixed set of nodes, the change of local structure leads to the change of topological structure. Then we can measure the entropy of the local structure by computing the empirical distribution of specific local structure over all samples. In this section, two localized entropy measures, edge existence entropy and geodesic entropy, that measures the entropy of edge and geodesic distance respectively are introduced.

As Definition 1.1.1, consider a network collection $\mathcal{G} = \{G_1, \ldots, G_k\}$ that have the same set of nodes. Suppose $A^{(k)}$ represents the adjacency matrix of G_k. The empirical distribution of edge e_{ij} is $P(e_{ij} = 1) = \frac{1}{N}\sum_{k=1}^{N} A_{ij}^{(k)}$ and $P(e_{ij} = 0) = \frac{1}{N}\sum_{k=1}^{N}(1 - A_{ij}^{(k)})$. Then we propose *edge existence entropy* which indicates the variability of connection e_{ij} among all samples can then be defined as

$$H_e(e_{ij}) = -(p_1 \log p_1 + p_0 \log p_0) \tag{3.1}$$

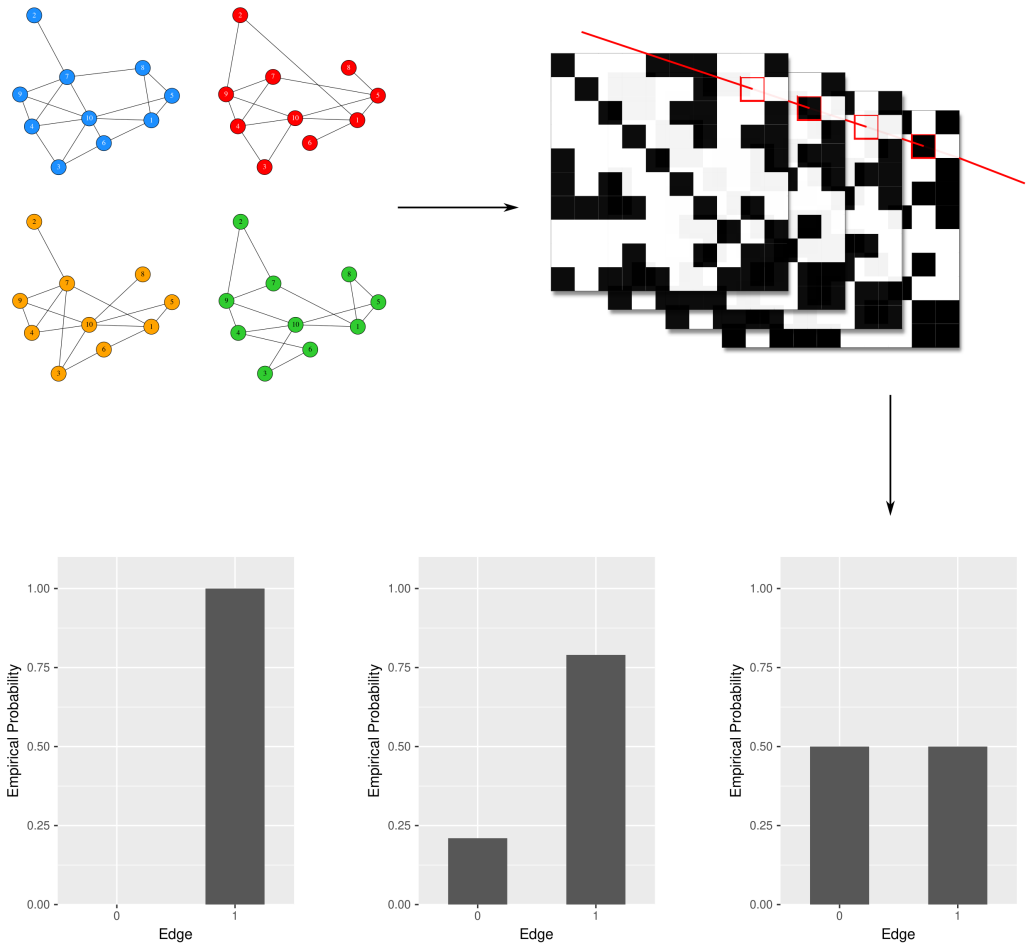


Figure 3.2. An example of edge existence entropy in a network collection of 4 binary undirected network samples. Consider the edge between node 2 and node 9. Among the 4 samples we have 2 samples with an edge between 2 and 9, and 2 samples without an edge. That means the probability of 1 (exists edge) and 0 (no edge) are 0.5 and 0.5 respectively. Therefore the edge existence entropy for this pair is 1. Combining the edge existence entropy of all pairs we get the edge existence entropy of the network collection.

where $p_1 = P(e_{ij} = 1)$ and $p_0 = P(e_{ij} = 0)$. For a pair of nodes (v_i, v_j), if they are connected in all network samples, there is no information provided per sample, as $H_e = 0$, because there is only one possible outcome $e_{ij} = 1$. Edge existence entropy reaches its maximum when $p_1 = p_0 = 0.5$ where the connection is most unstable. Figure 3.2 shows an example of calculating the edge existence entropy of a network collection of 4 undirected binary networks. An edge existence entropy can be computed for each potential edge in the network and therefore study the overall variability by the statistics of the distribution of edge existence entropy. For instance, the mean of entropies indicates the average level of variability and the variance of entropies indicates the deviation of variabilities.

Edge existence entropy sheds light on the variability of connection density of complex network collections. Meanwhile, the communication efficiency is another critical property that helps explain the dynamical process operating on a network. For instance, the distance between nodes, which is the length of shortest path between them, implies the communication efficiency. Hereby the *geodesic entropy* is proposed as:

$$H_d(d_{ij}) = -\sum_{d=0}^{\infty} p_d \log p_d \tag{3.2}$$

where p_d is the proportion of network samples where the distance between node v_i and v_j is d among all N network samples.

3.4.2 Mutual Information in Clustering Effect

Edge existence entropy measures the variability of edges over all samples independently. The transitivity of variability in network collection can be studied with information entropy as well. In this section, we are going to discuss a basic application of mutual information in studying the clustering effect in network collections.

In information theory, mutual information of two random variables measures the amount of uncertainty decrease of one random variable after considering another random variable. For example, if two random variable are independent, knowing one

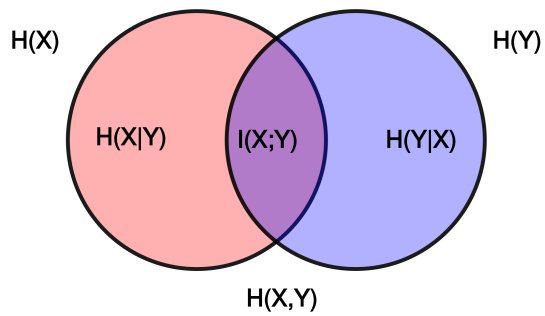


Figure 3.3. Venn diagram showing the relationships of different information with respect to variable X and Y. Left $I(X;Y)$ (purple area) represents the mutual information of X and Y.

random variable does not help reduce the uncertainty of the other random variable. Therefore the mutual information between the two random variables are 0. As shown in Figure 3.3, the mutual information $I(X;Y)$ between random variables X and Y can be calculated as:

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{3.3}$$

where $H(X)$ and $H(Y)$ are the entropy of X and Y, and $H(X,Y)$ is the joint entropy of X and Y.

Consider the local clustering effect in graphs with random variable X representing node A has connections to both B and C, and random variable Y representing node B and C are connected. With multiple networks observation we can calculate $H(X)$ and $H(Y)$ indicating the variability of the connection to any two neighbors and the connection between the two neighbors, respectively. Then the transitive mutual information of a triad (A, B, C) thus can be calculated as Equation 3.3 and $I(X;Y)$ is the information reduction in uncertainty of the connection between a pair of nodes due to the knowledge of the connection of a common neighbor. Higher $I(X;Y)$ indicates higher influence of variability of the transitive relation, and the connection between a pair of nodes and their common neighbor are independent if $I(X;Y) = 0$.

3.5 Experimental Result

In this section, measures introduced in Section 3.2, 3.3, and 3.4 are applied to three network collections for comparison. A set of structural brain networks (networks inferred from physical structure that vary by individuals), resting-state functional brain networks (networks inferred from correlation that vary by individuals) and the correlation networks of SP500 stock price (networks inferred from correlation that vary by time) are tested. The structural and functional brain networks of 100 subjects are processed based on the Human Connectome Project (HCP) 900 dataset [72]. The processing of the diffusion weighted imaging (DWI) data to get the structural brain

networks are described in [71]. The functional brain networks are inferred by thresholding correlation matrix of functional MRI blood-oxygen-level-dependent (BOLD) signal between regions of interest (ROIs) in resting-state [71]. The correlation networks of SP500 stock price is generated by inferring the Pearson correlation matrices between SP500 stocks from 1997 to 2017 with time window of 3 months[2]. A basic description of the network collections are shown in Table 3.1.

Table 3.1. Description of Three Network Collections. Sample size indicates the number of networks in each network collection. The density, transitivity and diameter are reported with mean $\pm$ standard deviation.

Name	**Sample Size**	**# of Nodes**	**Density**	**Transitivity**	**Diameter**
structural brain	100	360	0.032 ± 0.001	0.422 ± 0.011	8.11 ± 0.65
functional brain	100	360	0.050 ± 0.014	0.580 ± 0.051	9.22 ± 1.37
stock price	80	193	0.132 ± 0.110	0.674 ± 0.074	10.23 ± 2.51

3.5.1 Statistics of Global Measures

This experiment shows the variance of transitivity, assortativity of degree and modularity distribution of the three network collections for comparison. Transitivity measures the frequency of triangles and then the strength of transitive relation between nodes. Assortativity of degree indicates the similarity of node degrees. Modularity measures the strength of community structure and high modularity implies a dense intra-community connection and sparse inter-community structure. The modularity is based on predefined community structure. For structural and functional brain networks from HCP, the community is 7 resting-state networks (RSN) [87]. For stock price correlation networks, the community structure is the 11 industries contained in the consumer discretionary sector. and the results are shown in Figure 3.4.

The variances of transitivity σ^2_{trans} for structural brain networks, functional brain networks and stock price correlation networks are 0.00012, 0.00264 and 0.00554 re-

[2]We only include the stocks that have records no later than 2017 which gives the 193 qualified stocks.

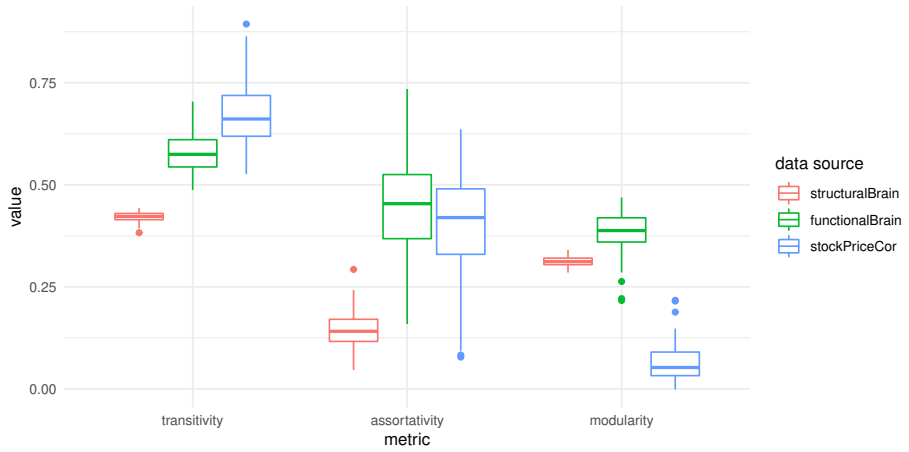


Figure 3.4. The transitivity, assortativity and modularity of three network collections.

spectively. The variances of assortativity σ^2_{assort} are 0.0017, 0.0137 and 0.0154. And the variances of modularity σ^2_{mod} are 0.00014, 0.00231 and 0.00215. By an one side F-test for two sample variance test with 5% confidence level, we can conclude that $\sigma^2_{trans}(struturalBrain) < \sigma^2_{trans}(functionalBrain) < \sigma^2_{trans}(stockPriceCor)$, $\sigma^2_{assort}(struturalBrain) < \sigma^2_{assort}(functionalBrain) \approx \sigma^2_{assort}(stockPriceCor)$ and $\sigma^2_{mod}(struturalBrain) < \sigma^2_{mod}(functionalBrain) \approx \sigma^2_{mod}(stockPriceCor)$. That means in all 3 graph metrics, structural brain networks have the lowest variability (most consistent) than functional brain networks and stock price correlation networks.

3.5.2 Pairwise Dissimilarity

In this experiment, the pairwise dissimilarity of degree distribution is analyzed. The JSD of degree distribution between each pair of networks in network collections is computed and the histogram of it are shown on the left in Figure 3.5. High JSD indicates greater discrepancy of degree distribution within the network collection. In the right panel, an abstract space of networks is constructed by 2 dimension MDS on the JSD matrices as distance matrices. From the experimental result, the network collection of stock price correlation has higher internal degree distribution discrepancy.

Structural brain networks have the lowest variability among the three. This result is consistent with the variance in network morphospace computed in Section 3.5.1.

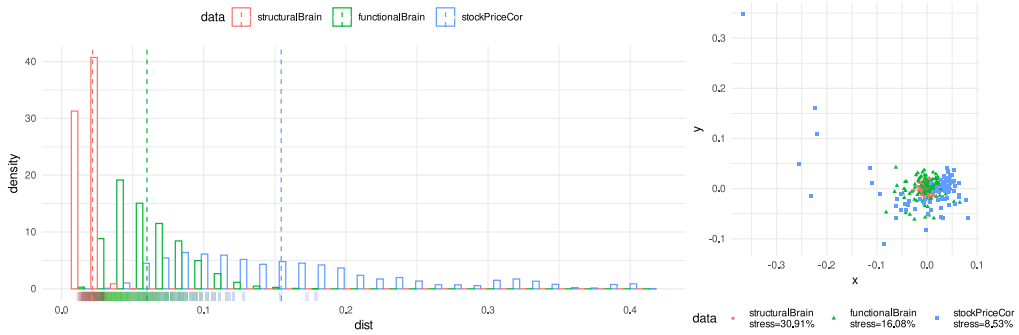


Figure 3.5. Left: histogram of the distribution of pairwise JSD of degree distributions for three network collections. Right: Recover the JSD to 2 dimensional space by multidimensional scaling (MDS). The stress value measures the goodness of fit of the recovery. Lower stress value indicates better fit.

3.5.3 Information Entropy

As described in Section 3.4, the information entropy of network collections are computed. Figure 3.6 shows the edge existence entropy and geodesic entropy of the three network collections. The structural brain network collection has most edge existence entropies close to 0 implying the topological structure is more consistent compared to functional brain networks and stock price correlation networks. That implies the individual difference of brain structure is lower than the functional connectivity with respect to this measure. As a comparison, the stock price correlation networks have the largest variability among the three.

Figure 3.7 shows the distribution of transitive mutual information for all triads. Here $H(X)$, $H(Y)$ and $H(X, Y)$ are computed with base of 2. The structural brain networks have 0 mutual information for most triads and functional brain networks have higher mutual information, which implies the knowledge of common neighbors does not reduce the uncertainty of connection in structural and functional brain networks in most cases. As shown in Figure 3.4 where stock price correlation networks

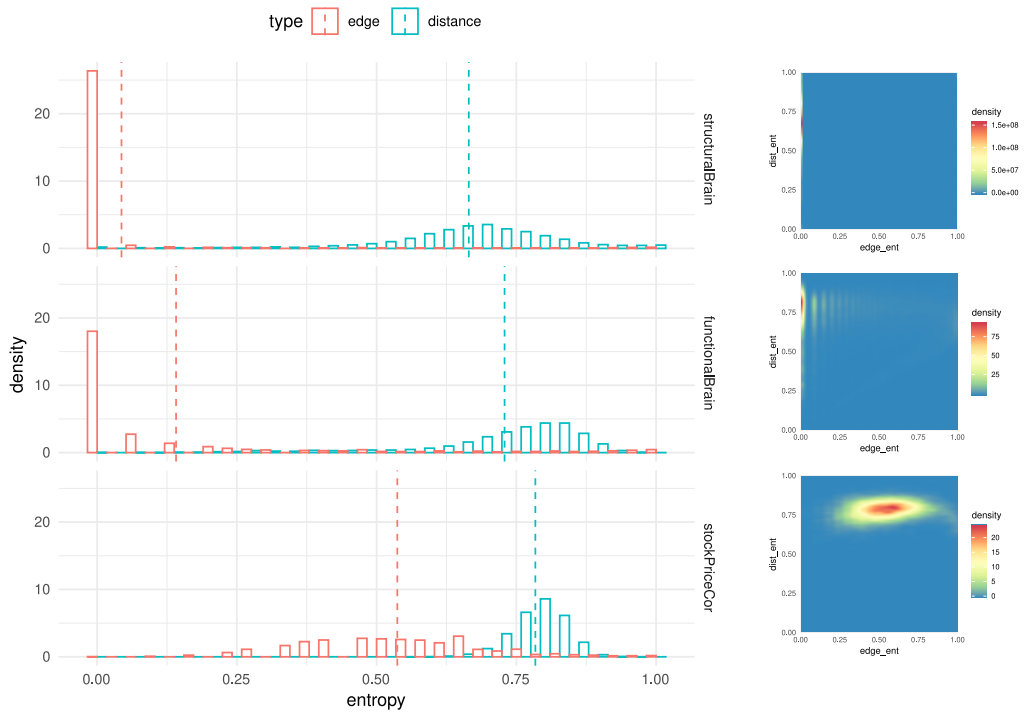


Figure 3.6. The histogram and 2D density plot (with kernel density estimation) of edge existence entropy versus geodesic entropy of the structural brain network collections, functional brain networks and stock price correlation networks. The left panel shows the histogram of edge existence entropy and geodesic entropy and the vertical dashed lines represent the mean. Right panel shows the distribution in edge existence entropy versus geodesic entropy space.

have the highest variance of transitivity among the three, they are expected to have higher transitive mutual information than structural and functional brain networks as shown in Figure 3.6.

3.6 Entropic Variability of Structural Brain Networks[3]

In this section, the edge existence entropy and geodesic entropy of structural brain networks and synthesized networks by *dk*-2.5 [23], ABNG [66] and configuration model [56, 67] are tested. The 100 structural brain networks are processed by

[3]A version of this section has been previously published in *Network Science* (DOI: https://doi.org/10.1017/nws.2019.63) [24].

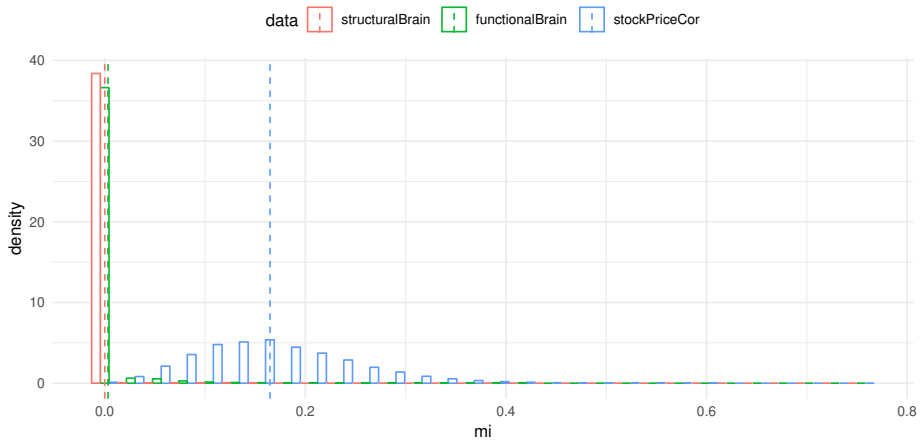


Figure 3.7. Histogram of transitive mutual information. Structural brain networks have lowest transitive mutual information which means variability of the connections to the common neighbors does not provide as much information to variability of the connection between the common neighbors compared to functional brain networks and stock price correlation networks.

diffusion weighted imaging (DWI) data as described in [71]. Instead of randomly picking one network from the network collection as input, we parameterize each model using multiple input networks assuming that the additional information will enhance the ability of models to synthesize collections that are representative of the input distribution. It is known that sample size will impact the accuracy of entropy measure [103]. In addition to the model training and synthesis on the complete dataset, partial input is considered by bootstrapping to verify the modeling performance and the effect of sample size on the entropy measures.

To verify the effect of sample size to the entropy methods, 20 replications are done for each sample size $s_i \in S = \{2, 3, 5, 10, 20, 50, 80, 100\}$. In each replication, s_i networks are randomly drawn from the 100 networks without replacement, thus creating a network collection with sample size s_i, which is then used to compute edge existence entropy and geodesic entropy. Given a sampled network collection, a mean degree sequence is estimated by averaging the degree sequences of the s_i networks in the collection. This mean degree sequence is then used by the configuration model to synthesize a network collection. Similarly, for ABNG, mean value of the user-defined

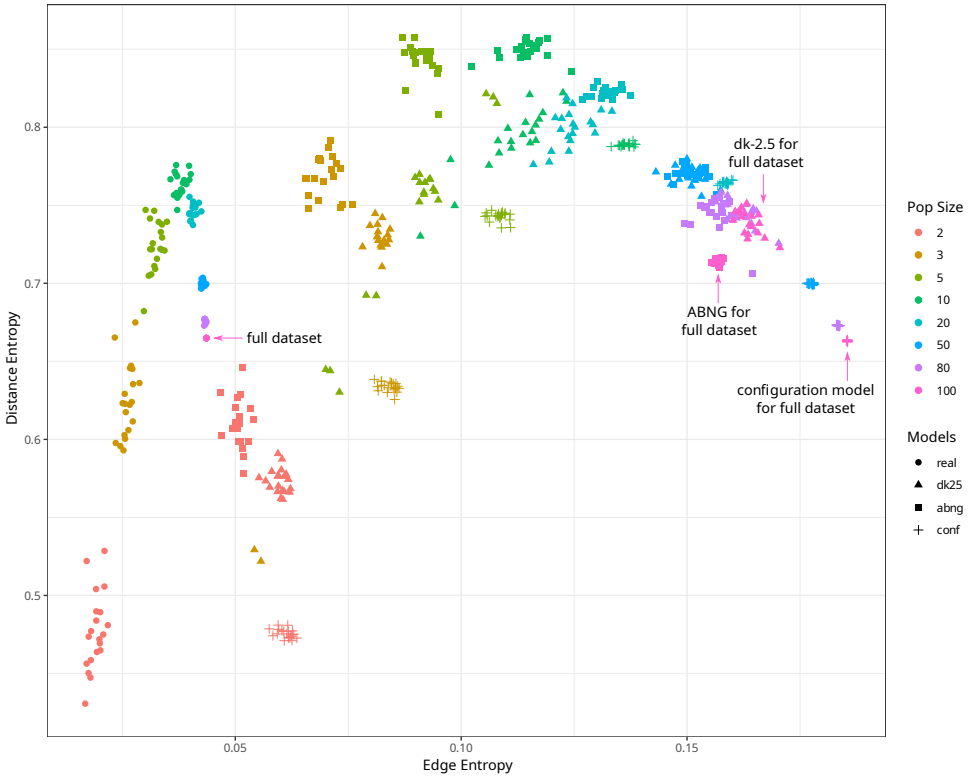


Figure 3.8. Distribution of edge existence entropy versus geodesic entropy for brain networks and synthetic networks with different sample sizes. Each point represents the edge and geodesic entropy of one network collection containing multiple networks. With increasing sample size, the geodesic entropy increases and gets peaked at around sample size of 10 and then decreases. That is caused by insufficient sample size and network sparsity. Models that are better on matching the variability will have scatters closer to "full dataset".

characteristics in the network collection can be used for learning action matrices for the collection. For the dk-random graph model, a probability matrix is created by averaging the adjacency matrices of the s_i networks in the collection. A network is then sampled from the probability matrix as representative network, which is repeatedly rewired using the $2.5k$-rewiring scheme to synthesize network collections.

Figure 3.8 shows the edge and geodesic entropies of structural brain networks and networks synthesized using the configuration model, dk-random graphs, and ABNG. The real network collections lie on the left side of the space, and the synthesized network collections of three models are positioned right of them. That implies syn-

thesized networks have higher edge existence entropy compared to real networks. The edge and geodesic entropy of the full dataset (100 structural brain networks) and collection synthesized by different models are annotated in the plot. Network collections with bootstrapping are colored by their sample sizes. For the three models and input, the edge existence entropy and geodesic entropy both increase with increasing sample size before 10 samples despite the geodesic entropy decreases when the sample size is larger than 10. Both entropy measures converge for more than 50 samples. Generally, limited sample size will reduce the accuracy of entropy measures.

The performance of network generative models on local variability can be evaluated by comparing the entropy of input network collection and model synthesized network collections. All three models overestimate edge existence entropies which means in this scenario, true structural brain networks have lower variability in local connectivity than the models express overall. ABNG gets a better estimate than the other two models with respect to edge existence entropy. As for geodesic entropy, both ABNG and *dk*-random graph models overestimate it but the configuration model estimates the geodesic entropy accurately. In conclusion, none of the three models can match the variability of input network collection across all measures. They either overestimate the edge existence entropy or overestimate geodesic entropy.

Figure 3.9 shows the assortativity-transitivity space, for the 100 structural brain networks and network collections synthesized in the results presented in Figure 3.8. Instead of 20 replications for each sample size setting, Figure 3.9 only shows one replication. Each point in the plot represents the assortativity and transitivity of one network. Among the three models, configuration model, *dk*-random graphs and ABNG, *dk*-random graph model performs the best on matching the centroid which represents the average metrics. However, it has the lowest generalized variance implying it fails at matching the variability of the group. ABNG outperforms the other two models at matching the generalized variance implying its capability of capturing the variability of input networks. Furthermore, there is no significant relation

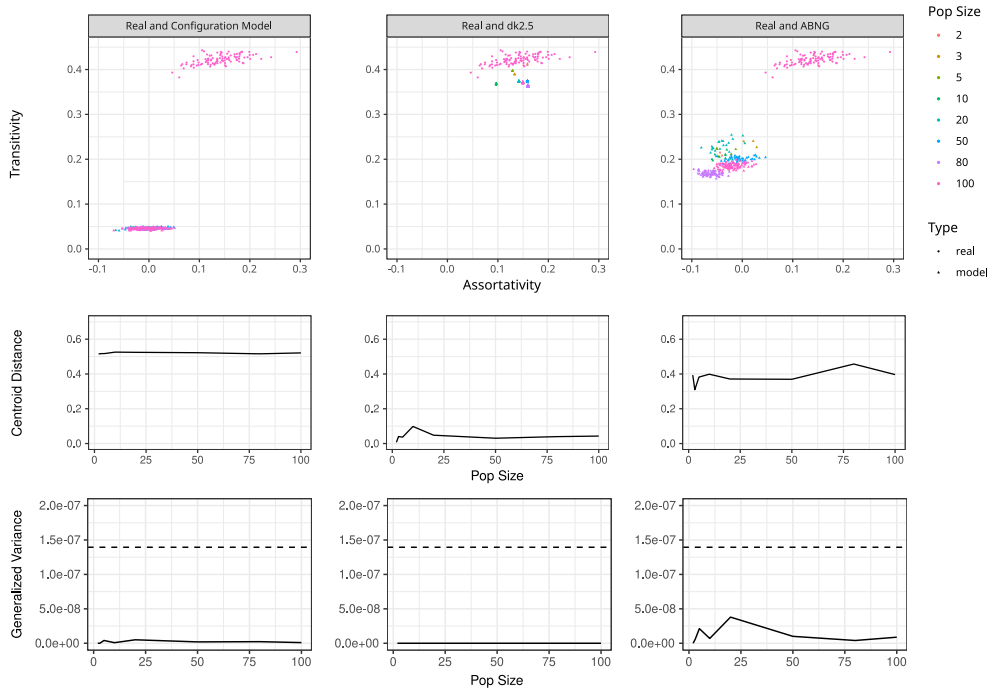


Figure 3.9. First row shows distribution of assortativity of vertex degree and transitivity in network collections of structural brain networks and networks synthesized from them by configuration model, *dk*-random graphs and ABNG. Second row shows the centroid distance between synthesized networks and input networks. Lower distance means better estimate on mean value. Last row shows the generalized variance of assortativity and transitivity of synthesized networks while the dashed line represents the level of input networks. Being closer to dashed line indicates better estimate on the variance on both metrics.

between sample size (annotated by color) and the goodness-of-fit in the space for all three models.

3.7 Entropic Variability of Functional Brain Networks

3.7.1 Subject-level variability

In order to measure the variability caused by individual difference among subjects, we compute the edge existence entropy and geodesic entropy of functional brain networks over the 100 randomly selected subjects in resting-state and performing tasks

respectively. The 2D distribution of edge existence entropy versus geodesic entropy of different regions in different tasks are shown accordingly. The functional brain networks are generated from the Pearson correlation between 360 ROIs in pairs. Any correlation higher than 0.2 is kept as an edge in networks therefore (100 subjects $\times$ 8 sessions $\times$ 2 encoding phases =) 1900 networks are generated. The 1900 networks are divided into 19 groups by tasks and phases. Each group has 100 networks which is defined as a network collection, and the edge existence entropy and geodesic entropy are computed with respect to Equation 3.1 and 3.2 for each group. The edge existence entropy and geodesic entropy of each pair of nodes are computed according to the group setting, which implies the variability of connection and communication efficiency over the 100 subjects in different states.

Considering the resting state network (RSN) partition [87] of the 360 ROIs, there are intra-RSN connections representing the interaction within same functional community and inter-RSN connections representing the interaction between different functional communities. Since the networks are sparser for inter-RSN connections and denser for intra-RSN connections, the entropies for inter-community connections are ignored for the following analysis to focus on the variability of connections within functional areas. For each group (task and phase), the pairs within RSNs are divided into 8 different RSNs that the pair belongs to.

Figure 3.10 shows the 2D density plot of the edge existence entropy versus geodesic entropy in different states by RSNs. Figure 3.10 shows the distribution of entropic variability changes for different states. That implies different tasks require different levels of entropic variability change. If the distribution in resting-state is considered as baseline, the distributional distance of entropic variabilities from resting-state to task indicates the reconfiguration of intra-RSN variability for each task. Kernel maximum mean discrepancy is used to measure the distance between pair of bivariate distribution estimated by observed samples.

Figure 3.11 shows that except for subcortical area, all regions have consistent variability reconfiguration between test (resting-state 1 as baseline) and retest (resting-

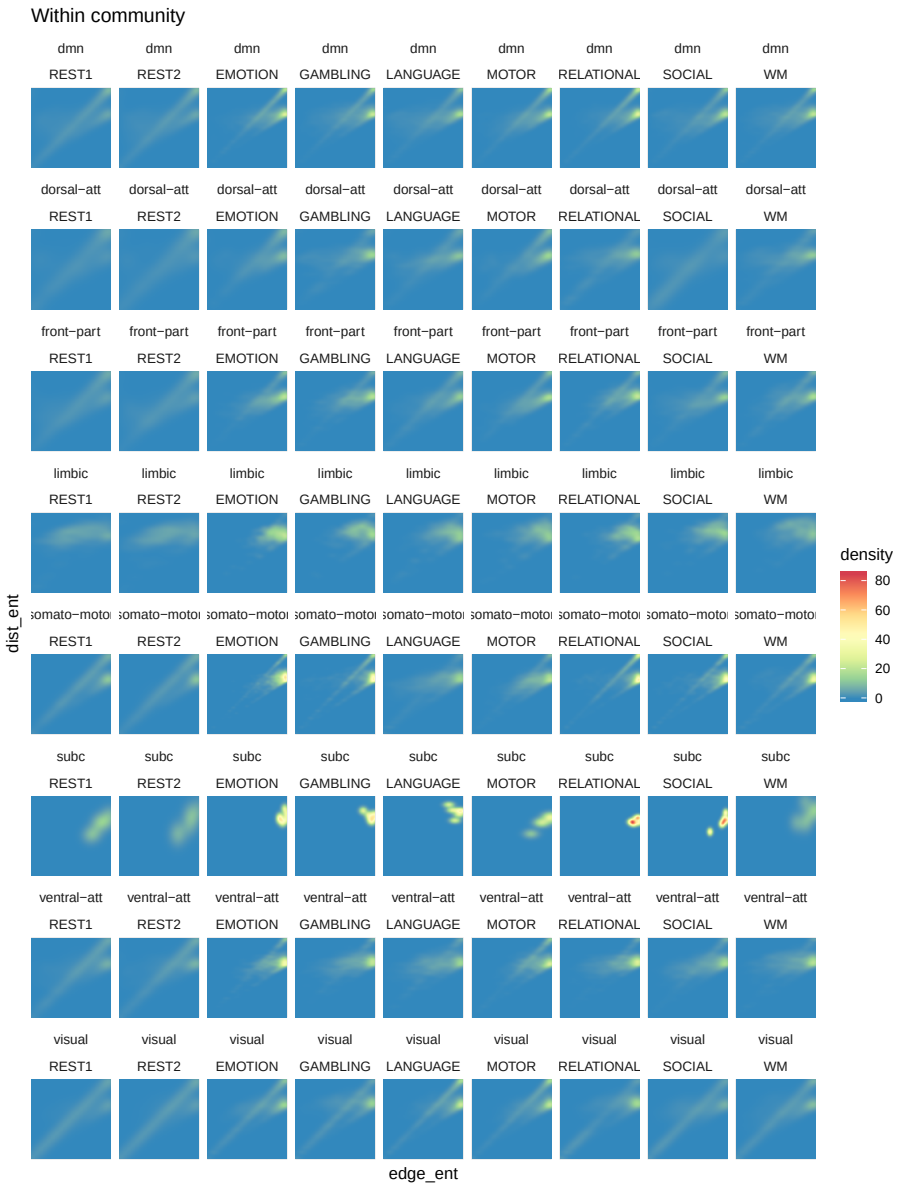


Figure 3.10. 2D kernel density estimation of edge existence entropy and geodesic entropy distribution grouped by RSNs. X axis and Y axis represents edge existence entropy (ranged from 0 to 1) and geodesic entropy (ranged from 0 to 1) respectively. Red indicates more points (denser distributed) in area. Blue indicates less (sparser distributed) points in area.

state 2 as baseline). Limbic system requires the most intense reconfiguration for all 7 tasks because it has the largest area of polygon.

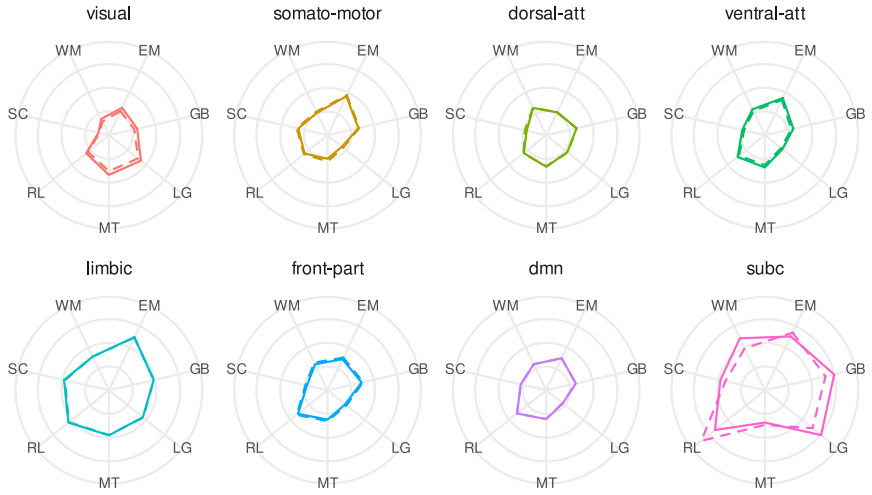


Figure 3.11. Kernel maximum mean discrepancy between resting-state (baseline) and different tasks grouped by RSNs. The 8 tasks performed are emotion (EM), gambling (GB), language (LG), motor (MT), relation (RL), social (SC) and working memory (WM). Solid line and dashed line indicate comparison to resting-state session 1 and seesion 2 respectively. Lower value in the plot implies more similar to resting-state. Larger area implies more intense reconfiguration from resting-state in general.

3.7.2 Temporal variability

In Section 3.7.1, the variability caused by individual difference is analyzed by deriving a network for each subject. In order to study the variability caused by time during the scan session, the temporal network variability is studied by taking a series of temporal networks as network collection. First, we randomly select one fraction of time series of the same length for all tasks to exclude the impact of length of session. Then the multidimensional time series is decomposed into multiple snapshots by sliding window method with step size of one. One network is generated for each snapshot therefore a series of networks is generated. In this section, the total length selected is 150 frames and the window size is 90. Therefore one series contains 61 networks as samples. Since the step size is one, the change of the network topological structure is very small and the geodesic entropy is close to 0. In this section, we only study the edge existence entropy with respect to time.

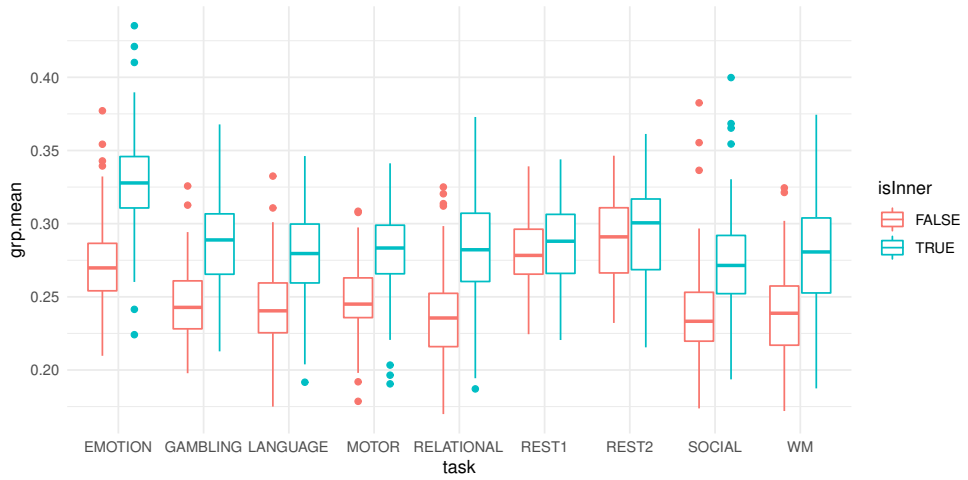


Figure 3.12. Box plot of edge existence entropy. Y axis indicates the mean edge existence entropy of different subjects. Higher edge existence entropy implies higher variability over time. The edge existence entropy is divided into two group: edges within community (in blue) and edges across communities (in red).

Figure 3.12 shows the distribution of mean edge existence entropy of different subjects for different tasks grouped by if the connection is intra-RSN or inter-RSN. Resting-state shows a homogeneity of edge variability for both intra-RSN and inter-RSN connections compared to all other tasks. For the 2 resting-state sessions, the p-values of t-test between mean edge existence entropy among intra-RSN connection and inter-RSN connection are 0.0774 and 0.1717 respectively, which implies there are no significant difference in mean between the two group of data for both sessions. Moreover, the intra-RSN edge existence entropy stays at the same level for both resting-state and task-state (except for emotion task), while inter-state connectivity variability decreases significantly for all task-states.

3.8 Conclusion

In this chapter, a general form of network variability is defined. The specific case that all networks in collection share the same set of nodes is discussed in the following sections. Based on the concept of network morphospace [91], the variance

of global metrics are firstly introduced as a variability measure. Then a pairwise dissimilarity method is introduced by taking account of the dependence between network observations. In addition, an entropic variability is proposed based on the concept that the variability of local structure (i.e. subgraphs and their topological properties) contributes to the overall variability.

These three methods are applied to three network collection data sets with different data source and format (including structural brain networks, functional brain networks and stock price correlation networks) for comparison. These three methods give the consistent conclusion that structural brain networks have the lowest variability, functional brain networks has higher variability and stock price correlation networks have the highest variability. Although these three methods have distinct concepts and calculation, they share the consensus on the variability of our network collection data sets.

Similar to the comparison of classic variability measures such as variance, interquartile range, median absolute deviation, etc., there is no conclusion that which measure is superior than other measures. These proposed network variability measures are defined with different conditions and concepts, and the measured variability should be interpreted according to the definition and scenario of application.

4. INFERENTIAL MODELING OF A NETWORK SERIES[1]

In this chapter, a network generative model is proposed to tackle the Question 2 "Can we devise a model that can reproduce the variability of observed network collection" with a specific case: variability in network collection introduced by time. Given a series of networks that have fixed set of nodes, the change of edges indicates the temporal variability. For instance, if the network does not change for the whole time series (i.e. all networks in series are identical), this network series has zero variability. To model the change and hence reproduce the network series, we assume that the changes of edges are related to some local structure (e.g. nodes with less neighbors are less likely to change associated edges.).

In this section, we applied the proposed model to the temporal functional brain networks in resting-state of 100 unrelated subjects. The experimental results show that the proposed model is able to reproduce similar network series, and the community of node has impact on the change of associated edges over time. In Section 4.1 and 4.2, an action-based network method is proposed for modeling the network changes. Performance evaluation and experimental results are shown in Section 4.3 and 4.4.

4.1 Action-based Network Generator

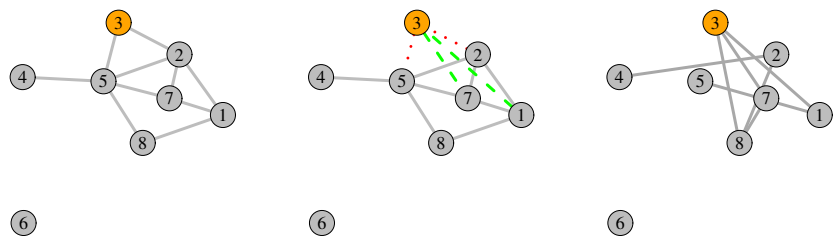
Action-based networks (ABN) [66] are based on the premise that nodes create, rewire or delete edges by probabilistically choosing from a set of predefined actions that locally modify network topology. Since interactions such as creation, rewiring or

[1]A version of this chapter has been previously published in *International Conference on Complex Networks and their Applications* (DOI: https://doi.org/10.1007/978-3-319-72150-7_103) [104].

deletion of edges occurs many times in a complex system and we do not have complete information at node level to state with high accuracy why changes of networks are made, it is natural to represent the actions as a probability distribution. Hence we can infer the proposed probability distribution from the observations via estimation methods such as maximum likelihood.

Suppose we are given a starting network $\mathcal{G}_0 = \{V, E_0\}$ with $n = |V|$ and $m_0 = |E_0|$ edges at time $t = 0$ and an action matrix $\vec{M}_{n\times k}$ with k actions. In action matrix $\vec{M}_{n\times k}$, each column represents an action that can be chosen by node v_i where $\sum_j \vec{M}_{ij} = 1\ \forall i = 1, \cdots, n$. Each row represents the probability distribution over all actions, which means $\vec{M}_{n\times k}$ is an uncompressed representation of the action preference. At time point t, every edge with endpoint v_i probabilistically chooses an action according to $\vec{M}_{i\cdot}$ and then rewires it to some other node according to the rule(s) of the action. Note that "do not rewire" is included as an action that would keep the edge for completeness of actions. Actions can be deterministic such as rewiring to the node with highest degree, or probabilistic such as rewiring with a probability proportional to node degree. $\mathcal{G}_{t+1}$ is the network that after all nodes complete such rewiring of edges in $\mathcal{G}_t$. Then repeat the process until some stopping criteria. Instead of synthesizing a series of networks independently, this algorithm generates the edge changes over time based on node actions and network structure at previous time point. In this way, we can synthesize a series of networks $< \mathcal{G}_0, \mathcal{G}_1, \cdots, \mathcal{G}_T >$ by sampling from $\vec{M}_{n\times k}$. An synthetic example of the algorithm is shown in Figure. 4.1.

Assume that we have a starting graph $\mathcal{G}_0$ as shown in Figure. 4.1(a) and 4 actions are preferential attachment on *degree*, *PageRank*, *closeness* and *no rewiring*. There are 8 nodes and 10 edges. If we only consider node v_3, it has two neighbors v_2, v_5 implying that the nodes to be rewired to are v_1, v_4, v_6, v_7, v_8. Consider an action matrix with 3rd row $\vec{M}_{3\cdot} = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \end{bmatrix}$, by which (v_3, v_2) and (v_3, v_5) are choosing their actions. With such probability distribution $\vec{M}_{3\cdot}$, (v_3, v_2) chooses degree and (v_3, v_5) chooses PageRank. Among the unconnected nodes, (v_3, v_2) would be rewired to v_1, v_4, v_6, v_7, v_8 with probability of $\begin{bmatrix} \frac{1}{3} & \frac{1}{9} & 0 & \frac{1}{3} & \frac{2}{9} \end{bmatrix}$



(a) (b) (c)

Figure 4.1. A toy example explaining the synthetic algorithm. (a) Starting network generated with Erdős-Rényi model $G_{nm}(n = 8, m = 10)$. (b) According to the action vector $\vec{M}_3$, v_3 rewired the two red dotted edges to the two green dashed edges. (c) After all nodes rewire based on their own action vector, the network in next time stamp.

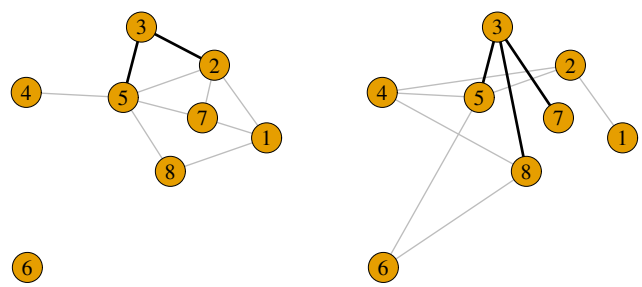
since $degre(v_1, v_4, v_6, v_7, v_8) = (3, 1, 0, 3, 2)$. Similarly, (v_3, v_5) would be rewired to v_1, v_4, v_6, v_7, v_8 with probability of $\begin{bmatrix} 0.23 & 0.21 & 0.08 & 0.25 & 0.23 \end{bmatrix}$ as their closeness. With a trial, (v_3, v_2) and (v_3, v_5) are rewired to (v_3, v_1) and (v_3, v_7) respectively. The result is shown in Figure. 4.1 (b). After a rewiring for all existed edges in $\mathcal{G}_0$ with the same action vector $\begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \end{bmatrix}$, a synthetic $\mathcal{G}_1$ is shown in Figure. 4.1 (c).

With a series of observed networks, maximum likelihood was applied to estimate the action matrix $\vec{M}$. Based on network $\mathcal{G}_t$, all edges rewired at $t+1$ can be represented by a graph $\Delta\mathcal{G}_t = \mathcal{G}_t - \mathcal{G}_{t-1}$ where $t = 1, \cdots T$. In other words, $\Delta\mathcal{G}_t$ gives the difference in edge set of $\mathcal{G}_t$ and $\mathcal{G}_{t+1}$. The posterior can be expressed by:

$$P(\vec{M}_{i\cdot}|\{\mathcal{G}_0, \mathcal{G}_1, \cdots, \mathcal{G}_T\}) = \prod_{t=1}^{T} \left(\prod_{v_j \in \mathcal{N}_i(\Delta\mathcal{G}_t)} \vec{M}_{i\cdot} \cdot \vec{p}_{ij}^{(t-1)} \right) \tag{4.1}$$

where $\mathcal{N}_i(\Delta\mathcal{G}_t)$ is the set of all first order neighbors (neighbors within one step away) of v_i in $\Delta\mathcal{G}_t$ (which are all rewired edges for node v_i), $\vec{p}_{ij}^{(t-1)}$ is the decision vector between node v_i and v_j at time point t, which determines the other endpoint for rewired edge. For instance, if action k is preferential attachment on closeness, the kth element of $\vec{p}_{ij}^{(t-1)}$ can be expressed by $\frac{\text{closeness}(v_j)}{\sum_{v \in V \setminus \{\mathcal{N}_i, v_i\}} \text{closeness}(v)}$, which is the proportion of closeness of node v_j to the sum of closeness of all non-neighbor nodes.

An example of $\vec{p}_{ijk}$ is shown in Figure. 4.2. Consider the two graphs as a series of networks and 4 actions: preferential attachment on *degree*, *closeness*, *PageRank*, and *no rewiring*. For node v_3, it rewired edge (v_3, v_2) to (v_3, v_7) and (v_3, v_8) at $t = 1$. In addition, (v_3, v_5) are not rewired at $t = 1$. At time $t = 0$, the degree of v_7 and v_8 takes 0.33 and 0.22 among degree of all potential nodes $(v_1, v_4, v_6, v_7, v_8)$, which means $p_{3,7,1}^{(0)} = 0.33$ and $p_{3,8,1}^{(0)} = 0.22$. Likewise, calculating the proportions of PageRank and closeness, we have $\vec{p}_{3,7}^{(0)} = \begin{bmatrix} 0.33 & 0.25 & 0.30 & 0 \end{bmatrix}$ and $\vec{p}_{3,8}^{(0)} = \begin{bmatrix} 0.22 & 0.23 & 0.22 & 0 \end{bmatrix}$. For the unchanged edge (v_3, v_5), $\vec{p}_{3,5}^{(0)} = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$.



$t = 0$ $t = 1$

Figure 4.2. An example showing observed networks in $t = 0$ and $t = 1$. The probability of this rewiring can be calculated based on the action matrix and corresponding graph metrics.

With the posterior probability, we can estimate each $\vec{M}_{i\cdot}$ by solving the constrained optimization problem:

$$\begin{aligned} \underset{\vec{M}_{i\cdot}}{\text{maximize}} \quad & \log(P(\vec{M}_{i\cdot}|\{\mathcal{G}_0, \mathcal{G}_1, \cdots, \mathcal{G}_T\})) \\ \text{subject to} \quad & \vec{1} \cdot \vec{M}_{i\cdot} = 1 \\ & \vec{M}_{i\cdot} \succeq \vec{0} \end{aligned} \tag{4.2}$$

where $\vec{M}_{i\cdot} \succeq \vec{0}$ means $\vec{M}_{i\cdot}$ is componentwisely greater or equal to 0.

4.2 Choice of Actions

The choice of actions is critical since the fundamental assumption is that nodes rewire edges based on actions related to local structure. For the data set of functional brain networks, the measures are classified into 6 main categories: basic concepts and measures, measures of integration, measures of segregation, measures of centrality, network motifs and measures of resilience [89]. Since network motifs and measures of resilience only have global measures instead of local measures on each node, we only consider the first 4 categories. Among the 4 categories, considering representativeness and computational cost, we chose 6 measures to be the basis of action set as shown in Table 4.1.

After the measures are determined, we need to devise the action based on these measures. Preferential attachment is a common strategy, however there is no prior information about the nodes behavior in brain networks. In other words, we do not know whether a node prefers or disfavors to rewire to nodes with higher metric value. Note that the dynamic action based network is different from the original ABNG in [66], which is the mechanism generating desired networks from sparse network. Therefore, we need to test whether a node prefers to connect nodes with higher statistic or inversely. To test, we estimate the action matrix on a sample of 16 subjects with both measurements and their inverse by maximizing the likelihood as

Table 4.1. List of actions allowed by nodes, not including "no rewiring"

Category	**Measurement**	**Calculation**
basic measures	degree	$k_i = \sum_{v_j \in \mathcal{N}_i} a_{ij}$, where $a_{ij} = 1$ if v_i and v_j are connected, and $a_{ij} = 0$ otherwise.
integration	local efficiency	$E_i = \frac{\sum_{v_j, v_h \in N, j \neq i} a_{ij} a_{ih} [d_{jh}(N_i)]^{-1}}{k_i(k_j - 1)}$, where $d_{jh}(N_i)$ is length of the shortest path between v_j and v_h, that contains only neighbors of v_i.
	modular similarity	$S_{ij} = \delta(m_i, m_j)$, where m_i is the module (RSN7[a]) containing v_i, and $\delta(m_i, m_j)$ if $m_i = m_j$, and 0 otherwise.
segregation	local clustering coefficient	$C_i = \frac{2t_i}{k_i(k_i - 1)}$, where t_i is the number of links between neighbors of v_i.
centrality	closeness	$c_i = \frac{1}{\sum_{i \neq j} d_{ij}}$, where d_{ij} is the shortest path between v_i and v_j.
	betweenness	$b_i = \sum_{h,j \in N, h \neq j, h \neq i, i \neq j} \frac{g_{hj}^{(i)}}{g_{hj}}$ where g_{hj} is the number of shortest paths between v_h and v_j and $g_{hj}^{(i)}$ is the number of shortest paths between v_h and v_j that pass through v_i.

[a] RSN7 refers to 7 resting-state cortical sub-networks (RSNs) as proposed by [87].

Equation 4.2. The 16 subjects are randomly selected from the 100 subjects and such sampling is due to the computational cost.

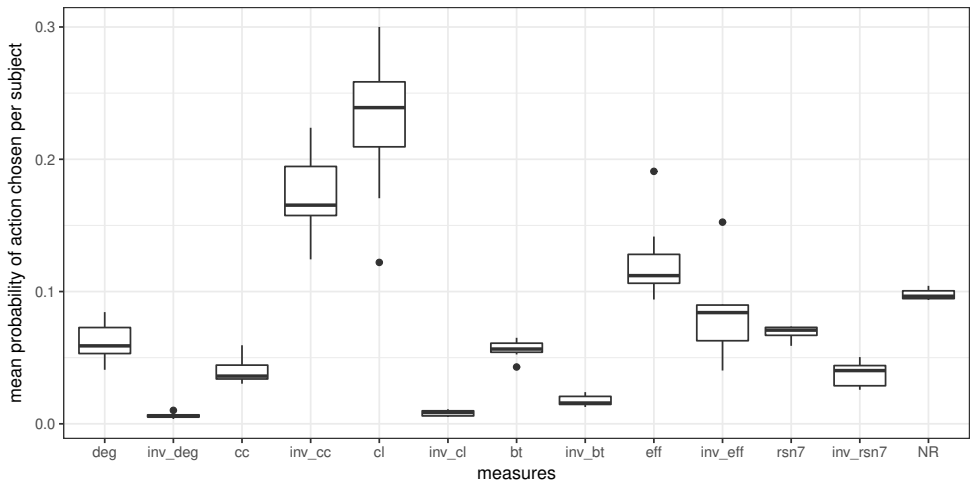


Figure 4.3. The mean probability of actions being chosen by subject. The actions in x-axis from left to right are *degree, inverse of degree, clustering coefficient, inverse of clustering coefficient, closeness, inverse of closeness, betweenness, inverse of betweenness, local efficiency, inverse of local efficiency, modular similarity, inverse of modular similarity and no rewiring.* An action matrix is estimated for each subject and there are 16 random subjects in this experiment. Lower mean probability indicates the action is less favored.

As shown in Figure. 4.3, the probability of preference on all measures are significantly higher than their inverse except for clustering coefficient. That implies nodes are less likely to rewire to a node with higher clustering coefficient over time. Thus, the actions will be used in the experiment are preferential attachment on degree, inverse of clustering coefficient, closeness, betweenness, local efficiency, RSN7 and no rewiring.

4.3 Evaluating Generator Goodness-of-fit

Our goal is to find a model that can synthesize networks with a similar global structure to the observed sequence of networks. Note that the likelihood of prior is served as an objective in our optimization. Past research used energy as an objective

[105,106] that is related to the global structure. The choice of measures to be included in such energy introduced bias into the optimization. Since our objective is the likelihood, our model is free from such bias introduced by predefined objective.

To evaluate the fitted model, synthetic networks are generated from the action matrix. Then we measure the pairwise similarity of some measures (such as degree distribution or closeness) between the raw network series and its synthetic networks in each time step. The similarity can be measured by the D-statistic in a two-sample Kolmogorov-Smirnov test (KS test). A lower D-statistic means two samples are drawn from sufficiently similar distribution. In this paper, KS test is applied to degree, clustering coefficient, closeness, betweenness and local efficiency distribution. The reason for selecting these metrics is the same as we chose actions in Section 4.2.

4.4 Experiments and Result

The resting-state fMRI data used in this experiment is preprocessed as Section 2.5. To divide the time series, we chose a window length of 100 time points (about 1.2 mins) since such a window endures long enough to allow adequate estimation of correlations over the frequencies that are present in the wavelet band of interest [0.001 Hz, 0.08 Hz]. Each scan session is then composed of 12 windows in total. For each pair of windows, a Pearson correlation coefficient on BOLD time series is computed, which generates a series of symmetric matrices overtime. A false discovery rate correction at level of $q < 0.05$ is applied to adjust p-values for multiple comparisons. These matrices can be transformed into networks if we assume that higher absolute value of correlation coefficient implies a stronger connection. To binarize the matrix, global thresholding is not a good choice since it tends to generate an either too dense network or network with too many isolated nodes and components. The isolated nodes and components impair some graph metrics such as shortest path length. In order to avoid this issue, we compute a correlation cutoff that can make all nodes connected as a giant component for each network. More specifically, starting with an empty

network, edges are added to the network sequentially as correlation increase until all nodes are connected.

As stated before, we fit the ABN model to brain networks generated from fMRI data of 100 unrelated subjects. To compare the mean and standard deviation of network structure metrics, each series of networks yields one estimated action matrix and the synthetic series of networks are generated from that action matrix for evaluation.

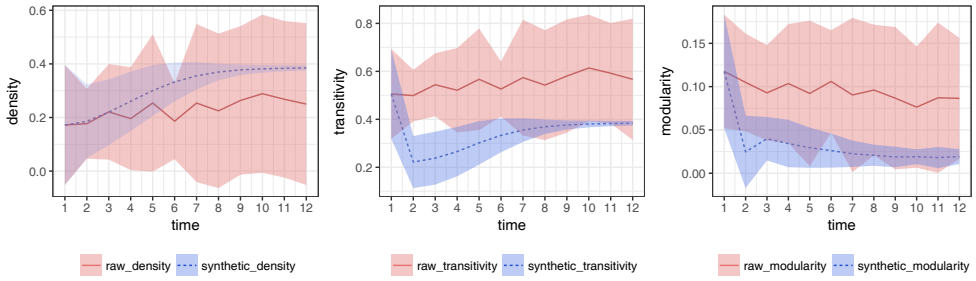


Figure 4.4. The density, transitivity and modularity of network series synthesized by our algorithm (in red) and raw observed network series (in blue). 20 replications are done for calculating the statistics. Mean value of the metrics is represented by lines and standard deviation is represented by ribbon.

Figure. 4.4 shows the result of fitted model. We find that the action-based model can mimic the dynamics of brain networks in some global structures, though the accuracy of prediction is decreasing with time. For instance, the degree is fitted in the first two steps but a smooth increase is observed in the next few steps. This can be partly explained by the cumulative error during the process. Since the raw series of networks is assumed to be a consistent process and our model cannot correct the bias when generating the series, some bias caused by random factor may lead the network away from the original network.

Another issue is the variance of the network. Compared to the raw data, the density, modularity [107] (with community of RSN7) and transitivity of synthetic networks all tend to converge in synthetic networks, however the raw data does not show such convergence. This may be caused by the synthetic algorithm we designed and actions we used. As a preliminary experiment, this model is not able to synthesize

most structures but since raw and synthetic networks have same trend at the first 3 time steps, the result seems nonetheless promising.

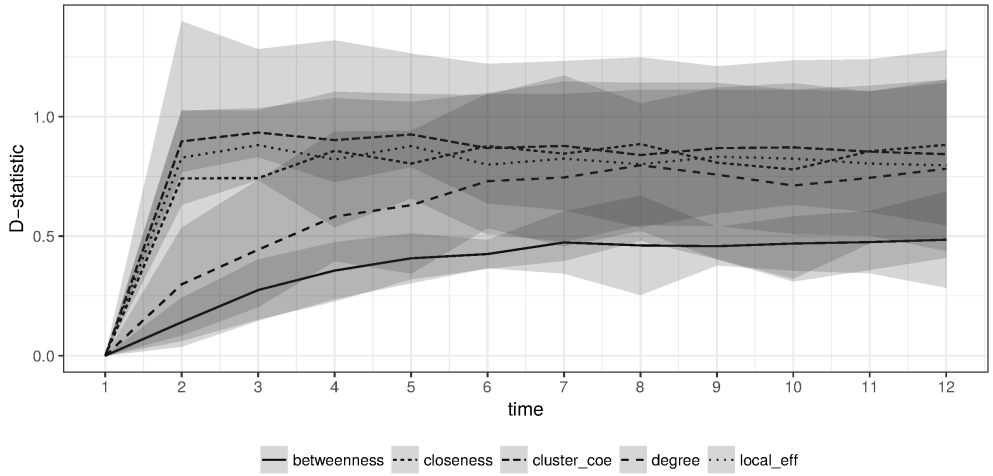


Figure 4.5. *D*-statistics of two sample KS test of betweenness, closeness, clustering coefficient, degree and local efficiency between raw network series and synthetic network series. Experiment is replicated 20 times for calculating the statistics. Line represents the mean value and ribbon indicates the standard deviation. Lower value means better fits.

Figure. 4.5 shows the result of two sample KS test on degree, closeness, betweenness, closeness, clustering coefficient and local efficiency distribution between target networks and synthetic networks. Betweenness distribution gets the best fit to target networks among all measures but closeness, clustering coefficient and local efficiency distribution exceed 0.5 at the second step. Degree distribution, as an important marker of network development and resilience, also exceeds 0.5 after the fourth step.

To investigate the discovered node-based action matrix grouped by node communities (RSN7), a nonmetric multidimensional scaling (MDS) is applied. MDS can represent the distances among the objects in a parsimonious way by reducing the dimension of the variables. We reduce the 7 variables (actions) to 2 by MDS and the corresponding result is shown in Figure. 4.6. The example shown in Figure. 4.6 is representative for all subjects and the results of other subjects are not shown due to page limits.

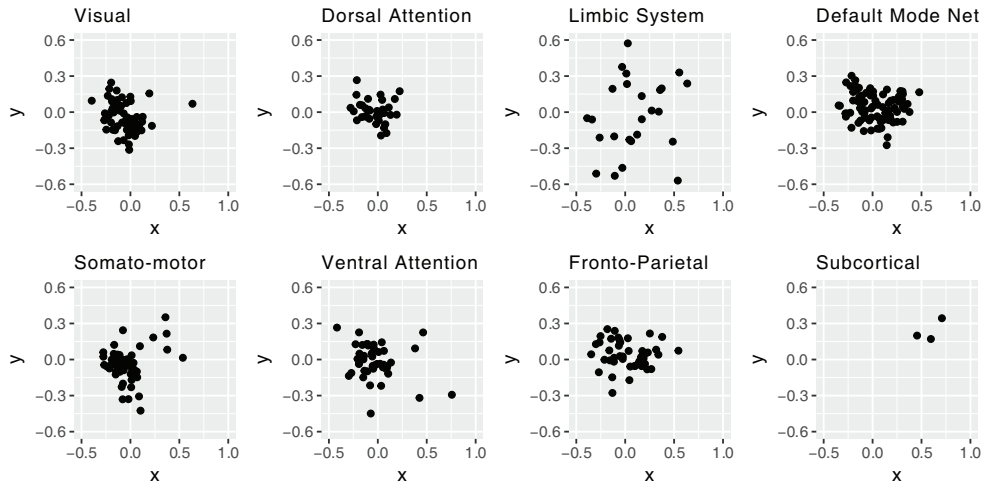


Figure 4.6. The action vectors of all ROIs of subject 100307 in session resting 1 with LR phase after nonmetric multidimensional scaling (MDS) grouped by RSN7 scheme as an example. The stress value of MDS is 17.82% indicating a fair dimension reduction. The action vector with length of 7 is reduced to vector in 2 abstract dimensions for visualization. One point represents the reduced action for one ROI, and denser distribution implies more consistent action preference within RSN. Note that the subcortical cerebellum is included for completeness.

We can find nodes in different resting-state networks have different preference on actions. For instance, nodes in RSN5 spread more sparsely in the plot which can be explained by a flexible action preference within modular and nodes in RSN3 are more unanimous on choosing actions. Although we only have 3 nodes in RSN8 which belongs to cerebellum, the actions of them are very different from nodes in cerebrum. Such dissimilarity of action preference across modular may be caused by brain functionality isolation.

4.5 Conclusion

In this chapter, we designed an action-based network model for dynamic networks and presented preliminary results. Our study makes some novel contribution of modeling a time series of networks. For the data set of functional brain networks, by estimating the action matrix of nodes, we can synthesize series of network with our

rewiring algorithm. The choice of actions shows nodes of brain networks prefer to rewire to nodes has higher degree, closeness, betweenness, local efficiency and modular similarity,but lower clustering coefficient. The investigation of action matrix shows nodes of different RSN tends to have different action settings. Although the synthetic networks can mimic the dynamics of raw data to some degree, our action based model is not as consistent as the raw networks. This may be caused by inaccurate synthetic algorithm and insufficient information of nodes. This action-based network model based on rewiring is a preliminary experimental generative model of brains dynamics which need further improvement. In addition, the decision of nodes rewiring should not only depend on the graph topological metrics. There should be some more attributes of brain ROIs included for completeness, such as the coordination of ROIs and functional relations between ROIs.

5. NETWORK COLLECTION MODEL BASED ON VARIATIONAL AUTO-ENCODER

In Chapter 4 we introduced an action-based model for synthesizing a series of network and implemented the model for dynamic functional brain networks. The result implies that the network series can be modeled by an action matrix over the rewiring between snapshots, which is equivalent to lag-1 difference in time series analysis. It hypothesizes that the network series is stationary, and the difference between consecutive observations dominates the variability. However, we cannot implement differencing method (such as computing edge switches in Chapter 4) with network collection that are not time series due to the lack of temporal dependence. To learn and reproduce the attributes of a collection of networks, we need tool that can process high-dimensional relational data. Variational autoencoder, a neural network-based generative model, has shown capability of learning the variability of high dimensional data such as images [108]. In what follows, we will discuss the implementation of VAE on graphs and some experimental results.

5.1 Variational Autoencoder

The recent burst of deep learning methods has shown great success on classification tasks and especially for recognizing patterns in images, audio, video and text [108–110]. Research questions on graph, as a non-Euclidean data structure, such as node classification, link prediction can utilize the deep learning framework for its high extensibility. The application of neural networks on graph data starts with recursive neural networks on tree structured data [111]. Then we have graph neural networks (GNN) and neural network for graphs (NN4G) that are able to handle general graph

data [112,113]. Based on the idea of NN4G, the graph convolutional network (GCN) is proposed and shows prominent performance on tasks such as node classification [114]. Besides the deep learning methods targeting at single networks, there are network models that learn the whole network as one object to complete tasks such as graph classification. One intuitive way of learning networks is to aggregate the learned nodal representations with deep sets [115]. Another way is to decompose the networks into paths generated from random walk and learn the path with RNN/LSTM [116]. However, most of these methods are implemented with small-scale networks such as molecules. The implementation of most current graph deep learning methods will incur computation issues on large-scale complex networks.

Besides the deep learning methods mentioned above, variational autoencoder (VAE) shows its preeminence on modeling and synthesizing high-dimensional data. An autoencoder is a category of unsupervised learning methods that try to regenerate high dimensional input data by a set of encoder and decoder functions [117]. Generally, the encoder compresses the input data into latent variables in lower dimensional space (feature space) and the decoder recovers the latent variables in feature space back to the space of input data [108, 117–119]. Normally the encoder and decoder are neural networks. In recent years, the implementation of VAE on networks has shown great potential on learning network structure. For instance, [120] proposed GraphVAE that takes adjacency matrix, edge attributes, nodal attributes and label of graphs as input X to generate valid chemical molecules. [121] has shown an implementation of VAE on citation network and achieved competitive results on edge prediction task compared to other deep learning methods. Although VAE achieves success on many graph related tasks, the computational cost is high especially for large networks (with more than 100 nodes).

Variational autoencoder is introduced by [108] based on the concept of variational inference on compressed latent variables in feature space. In Figure 5.1, X is the high dimensional input data, and encoder Q is a neural network that outputs two

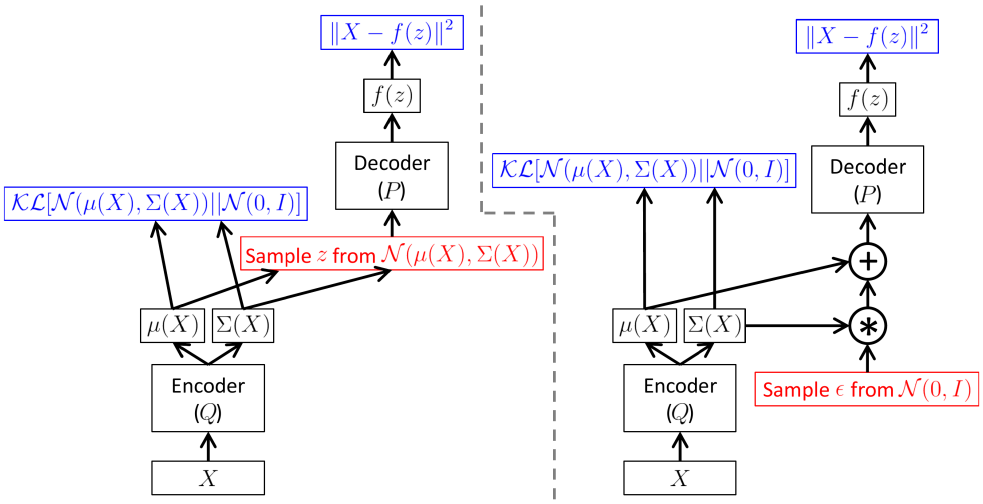


Figure 5.1. Examples of VAE implementation [119]. Left is without the 'reparameterization trick', and right is with it.

vectors $\mu(X)$ and ΣX, which are latent mean vector and latent variance vector[1] of X, respectively. z is the latent variable that follows standard normal distribution. For scheme without reparameterization, $\Sigma(X)$ is the covariance matrix and z is sampled from $\mathcal{N}(\mu(X), \Sigma(X))$ for specific X. Reparameterization trick replaces the sampling step with first sampling from $\epsilon \sim \mathcal{N}(0, I)$, and then computing $z = \mu(X) + \Sigma^{\frac{1}{2}}(X)$ as sample. The decoder P is a neural network as well. VAE has two objectives: generating $f(z)$ that is similar to input X, and encoding X into z that is similar to $\mathcal{N}(0, I)$. The two objectives can optimized by minimizing the L2 norm of $X - f(z)^2$ and the KL divergence between $\mathcal{N}(\mu(X), \Sigma(X))$ and $\mathcal{N}(0, I)$:

$$||X - f(z)|| + \mathcal{KL}[\mathcal{N}(\mu(X), \Sigma(X))||\mathcal{N}(0, I)$$

[1]Variational inference assumes independent Gaussian (zero covariance) for its simplicity.
[2]Depending on the objective and input type, this reconstruction loss can be replaced by other objective such as cross entropy between reconstructed X^{rec} and input X

5.2 Learn Functional Brain Networks with VAE

In this chapter, the implementation of VAE on functional brain networks are shown as an example of learning the distributional properties underlying complex network generative process. Brain networks modeling is a current frontier neuroscience for complementing simple descriptions with mechanistic predictions [82, 122]. For brain network activation prediction that predicts the activated area in brain in different states, it has shown that the task activation in pre-surgical populations can be predicted by resting connectivity [123]. Machine learning methods, such as neural networks, can also be implemented for predicting brain states based on fMRI [124, 125]. Although, prediction on brain network states has been studied for decades, there is a lack of models that are capable of predicting the whole brain connectivity [122]. Based on the high adaptability of VAE framework as generative model for high dimensional data and its successful application in different scenario, we believe VAE can be implemented as a generative network model for network collections.

The functional brain network data is processed as described in Section 2.5. The input networks are resting-state and 7 tasks functional brain networks of 100 subjects encoded in both LR and RL phase. Because the length of fMRI scan sessions varies for different tasks, the length of session will impact the prediction as an attribute. To eliminate the impact of the length of time series for task comparison, for each network sample we randomly select 150 consecutive frames[3] and compute the corresponding correlation matrices. The encoder contains one Multilayer Perceptron (MLP) with 20 neurons for $\mu(X)$ and one simple MLP with 20 neurons for $\Sigma(X)$.

Let $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ be a collection of n network observations that share the same set of nodes. For example, network G_i represents the functional connectivity, i.e., the network derived from binarized correlation matrix of BOLD signal between ROIs, of subject i in resting-state. A generative model $f(\mathcal{G})$ (VAE) learns the distri-

[3]The longest session is in *resting-state* including 1200 frames, and the shortest session is *emotion processing* including 176 frames.

bution of the network collection $\mathcal{G}$ and is capable of generating networks with similar distributional properties to $\mathcal{G}$.

The implementation of VAE is shown in Figure 5.2. A basic VAE consists of an encoder $q(z|X)$ and a decoder $p(X|z)$. The encoder $q(z|X)$ is a neural network that embeds X into continuous representation $z = (\mu_x, \sigma_x)$. For simplicity, the vectorized upper/lower triangular of the adjacency matrix represents graph G, which means for a graph G of n nodes, the input $X \in \mathbb{R}^{\frac{n(n-1)}{2}}$. In practice, latent variable z is in a lower dimensional space compared to input X. In addition, VAE has a strong assumption that the latent representation follows a standard Gaussian distribution, $z \sim \mathcal{N}(0, I)$. The decoder $p(X|z)$ is a neural network that recovers z to its original space to which X belongs.

For a graph represented by its vectorized adjacency matrix X, the loss function for the basic VAE is:

$$\mathcal{L}(X) = -\mathbb{E}_{z\sim q(z|X)}[\log p(X|z)] + D_{KL}(q(z|X)||p(z)) \tag{5.1}$$

The first term $-\mathbb{E}_{z\sim q(z|X)}[\log p(X|z)]$ is the reconstruction loss measured by cross entropy. It measures the distance between the input and reconstructed sample. Lower reconstruction loss indicates less information loss after the encoding and decoding process. The second term is the regularization loss measured by Kullback-Leibler divergence. It measures the information lost if we represent z with our regularizer (e.g. $\mathcal{N}(0, I)$ in this case). Lower regularization loss indicates better representation of the variance and diversity. In order to implement backpropagation for the loss function with random variable z, a reparameterization trick is applied by sampling from $q(z|x)$ to get a deterministic $z = \mu + \sigma \cdot \epsilon$. It's proved that the derivative of loss function by this reparameterization trick converges to the gradient of Equation 5.1 [108].

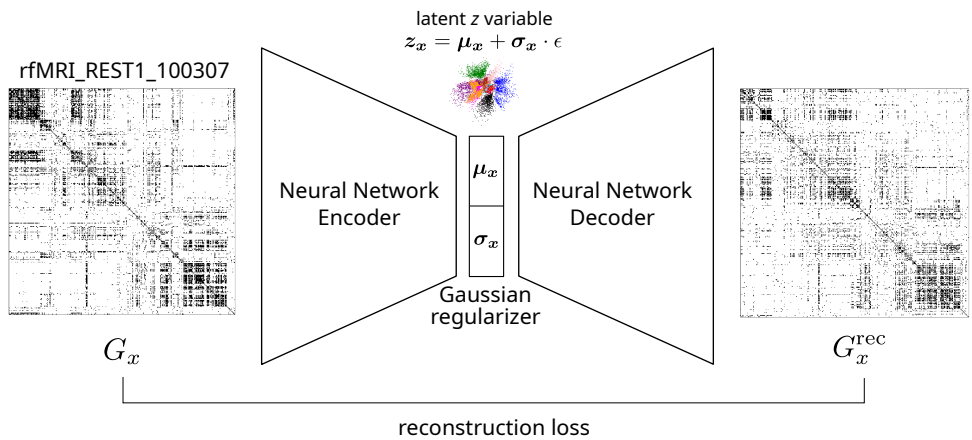


Figure 5.2. The implementation of VAE for brain networks represented by adjacency matrix. G_x on the left is the adjacency matrix of input network. G_x is then fed into the encoder (a neural network) which compresses the input matrix into lower dimensional latent variable z_x that follows $\mathcal{N}(\mu_x, \sigma_x)$. Then a decoder (a neural network) recovers the latent variable back to the form of G_x, which is G_x^{rec}. There are two objectives, Gaussian regularization loss that measures the difference between learned z and Gaussian distribution, and reconstruction loss that measures the difference between regenerated graph and real input graph, respectively.

5.2.1 Unsupervised Learning: Clustering Brain States

The functional brain networks are derived from the BOLD signal series of the 360 ROIs preprocessed as Section 2.5. For simplicity, all correlation matrices are binarized with a threshold of 0.5, i.e., ROI i and ROI j are connected if and only if the Pearson correlation between them $r_{ij} >= 0.5$.

The encoder should be capable of extracting the information of brain states when taking the scan. Since the length of scan sessions varies according to tasks/state, the derived correlation matrices are impacted by the difference in precision and effectiveness introduced by different sample size[4]. Therefore we randomly truncate 150 consecutive frames from original sessions as our data (the longest session, emotion processing, has 176 frames) to rule out the sample size factor. With the 360 dimensional time series of length 150, a 360×360 correlation matrix is derived for each

[4]The length of scan session are difference for different tasks but are fixed for the same task. The information of session length should not be identified as the effective factor for prediction.

run of each all conditions according to Table 2.2. With this dataset, the capability of feature extraction of encoder is verified. The networks that are derived from sessions of the same task are expected to be clustered in the latent space.

In this experiment, the encoder has one fully connected layers. The input vector is a $(\frac{n(n-1)}{2} =)$64620 dimension binary vector. The hidden layer has 400 dimension with rectified linear unit (ReLU) activation function. The output layer has 2 identical densely connected networks of 20 dimensions that output means and variances of latent z respectively. The value of z computed with given means and variances is then fed into the decoder. The decoder has 1 densely connected layers of dimension $(400, 64620)$ to recover the adjacency matrix.

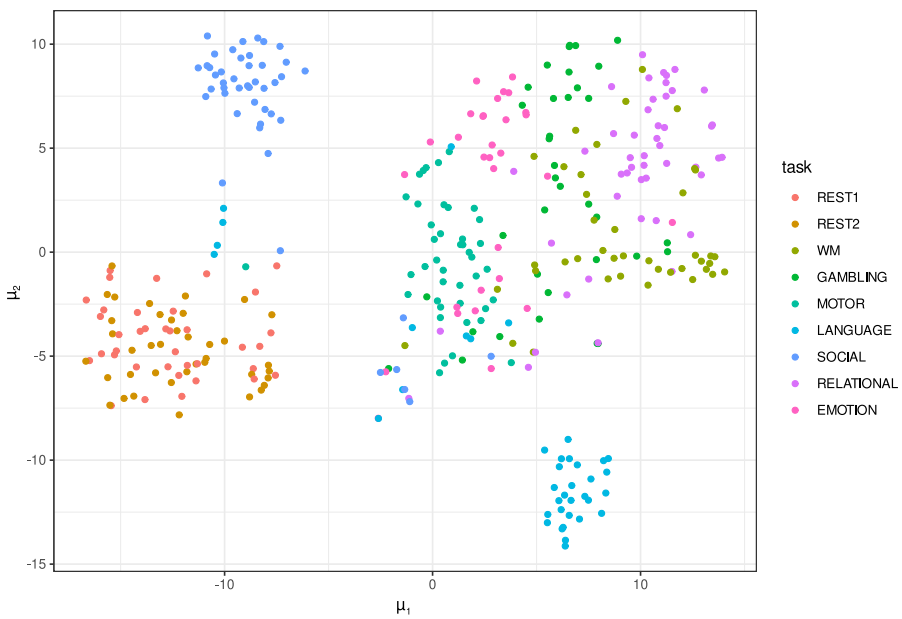


Figure 5.3. The distribution of functional brain networks of 40 subjects in encoded latent space of means by brain states. x and y axes are 2 abstract dimensions of the 20-dimension variable after t-SNE.

The trained z latent variable of the functional brain networks of 40 random subjects in 9 states (2 resting-state plus 7 tasks) are visualized in two dimensional ab-

stract space by t-Distributed Stochastic Neighbor Embedding (t-SNE) technique[5] in Figure 5.3. Networks in different states are clustered without the explicit information of state as input in training. This implies the encoder is capable of extract the state-related information of input functional brain networks. A comparison of the clustering in trained latent space and the original adjacency matrix space can be found in supplementary material in Section 7.2.

5.2.2 VAE as Network Collection Generative Model

The generalization capability of decoder can be verified by checking the distribution of synthetic networks as discussed in Section 1.1. In this experiment, the training networks are functional brain networks in resting-state for simply learning the variability caused by individual difference. Since the latent variable z is assumed to follow standard Gaussian distribution, the networks recovered from Gaussian samples by trained decoder are expected to have similar distribution with respective to training networks.

The architecture of the VAE is same as in Section 5.2.1. To compare the distributional properties of real resting-state functional brain networks and the synthetic networks, we choose three basic graph metrics: density, degree assortativity and transitivity as metrics.

Figure 5.4 shows the result of this experiment. The input network collection is fitted well with VAE in density and assortativity however the model underestimate the transitivity. There exist outliers in density metric are not captured by the model as well because of the long tailed distribution of input networks.

The underestimate of transitivity is potentially caused by the incapability of VAE embedding to reproduce the dependent relationship in low-degreed local structure in sparse complex networks [126]. The low dimensional embeddings destroyed the local

[5]t-SNE constructs a probability distribution over the distance between all pairs of samples in high dimension that the pair of similar (closer) samples has high probability. Then it constructs a lower dimensional space consists of all samples which are close to their location in original space.

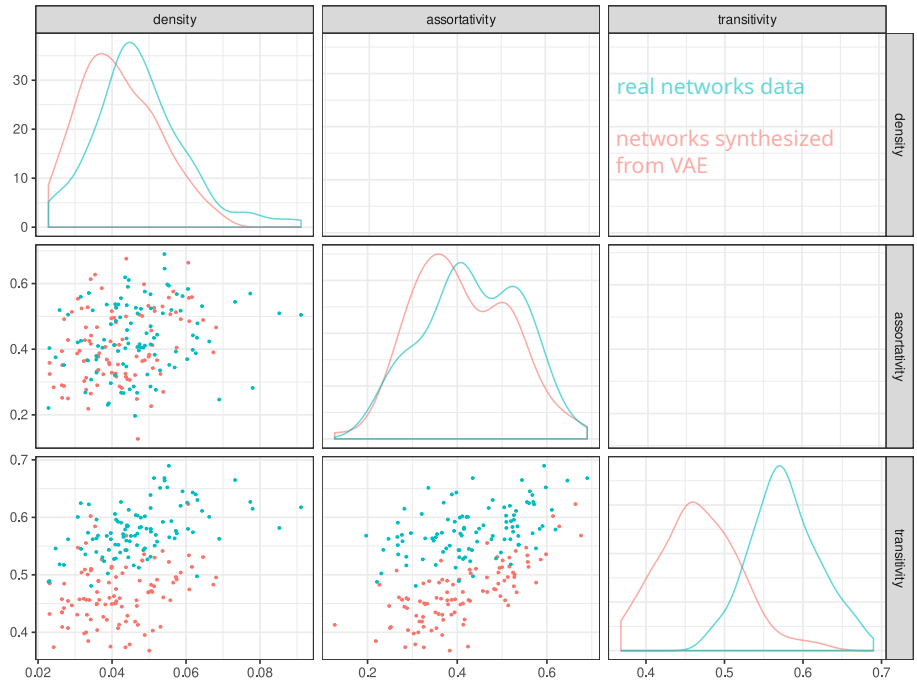


Figure 5.4. The density, assortativity and transitivity of 100 resting-state networks versus 100 synthetic networks synthesized from standard Gaussian samples z. In lower triangle, blue dots are networks of real data and red dots are networks synthesized from VAE. Diagonal plots are the kernel density estimate of the samples in corresponding metric. If red dots (curves) are closer to blue dots (curves), the synthesized networks are similar to input networks which means a good model with respect to the chosen metrics.

triangles. Although the distribution is shifted left due to the lost of triangles, the variance is captured pretty well due to the Gaussian assumption.

In addition to the synthetic results of resting-state brain networks, the results of synthetic networks in other task states are shown in Section 7.1. The result shows that the VAE generative model can also be applied to other non-resting states.

5.3 Task and Subject Disentanglement

VAE in previous sections has latent variable z as an abstract representation of the characteristics of input networks. Although we could find the relationship between z and network features by dimensional reduction methods such as Figure 5.3,

the features are entangled and we cannot directly specify the feature we want. For instance, if we want to extract the feature of tasks as task-related latent variable, z_t, then $z_t = 1$ represents functional brain networks in resting-state, $z_t = 2$ represents networks with motor task, etc. With respect to z_t and other dimensions of z we could infer the state of input networks and synthesize brain networks in other states.

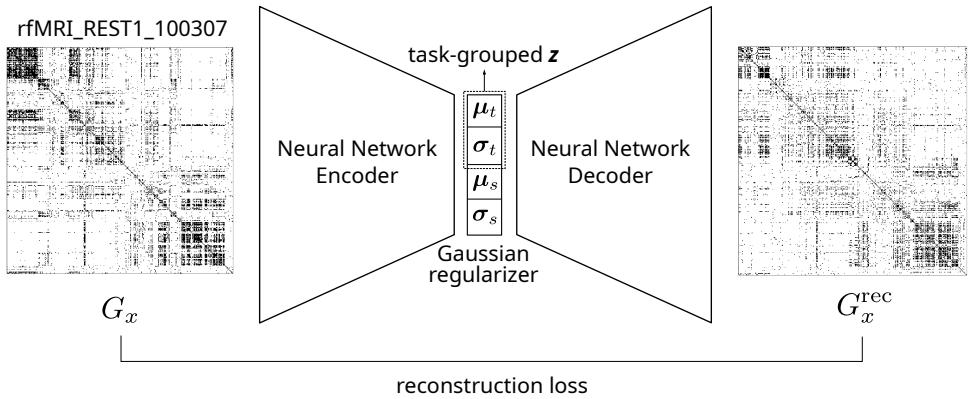


Figure 5.5. The implementation of task-disentangled multi-level VAE for brain networks represented by adjacency matrix. G_x on the left is the adjacency matrix of input network. G_x is then fed into the encoder (a neural network) which compresses the input matrix into two lower dimensional latent variable: task-related z_t that follows $\mathcal{N}(\mu_t, \sigma_t)$, and subject-related z_s that follows $\mathcal{N}(\mu_s, \sigma_s)$. Input networks in same states share the same z_t while all networks generate their own z_s as before. Then a decoder (a neural network) recovers the latent variables back to the form of G_x, which is G_x^{rec}. There are two objectives, Gaussian regularization loss that measures the difference between learned z and Gaussian distribution, and reconstruction loss that measures the difference between regenerated graph and real input graph, respectively.

In order to learn a disentangled representation of task and subject features, we implement the multi-level variational autoencoder [127]. The networks are grouped by tasks, i.e. group 1 is the subset of networks in resting-state, group 2 is the subset of networks in motor task, etc. Then we separate the latent variable into 2 parts $z = \{z_t, z_s\}$ with task latent variable z_t representing the feature of task and individual latent variable z_s representing the feature of individual subject. Figure 5.5 shows an diagram of the task-disentangled VAE for functional brain networks. The

regularization loss for z_s is calculated for each individual independently (same as the regular VAE regularization loss we did in last section). However for the task-grouped loss for z_t, the grouped regularization loss is calculated by multiplying the normal density functions:

$$q(t_x = t) \propto \prod_{i \in G} q(t_i = t|x_i). \tag{5.2}$$

5.3.1 Task Disentanglement: Multi-level VAE

In this experiment, we take the functional brain networks generated with full scan session as input for a multi-level VAE (MLVAE). The information of task is grouped as z_t. We train the ML-VAE model with functional brain networks of all tasks of the randomly selected 80 subjects. Based on the trained encoder, decoder and grouped task latent variable, the resting-state functional brain networks of the 20 subjects in test set are fed into the encoder to get an individual z_s. With respect to the z_s concluded from individual resting-state brain networks and the z_t concluded from the collection of functional brain networks in all 7 states, the model can predict the functional networks of this individual in all 7 states. Both encoder and decoder have one hidden layer including 400 neurons. Both task-disentangled latent variable z_t and individual latent variable z_s have dimension of 20.

In addition, an arithmetic model is implemented in this experiment as a baseline. It has been shown that group-averaging of functional brain networks gives valuable insight into typical organization of a population [128]. Therefore we assume that the reconfiguration between task and resting-state conditions [125] of individuals can be represented by the group-average reconfiguration. More specifically, assume we have the training set of adjacency matrices of N subjects, $\{A_t^{(i)}|i = 1, 2, \cdots, N\}$ where t represents the state of the subject (e.g. resting-state, gambling test, etc.). Then we have a grouped-average adjacency matrix for state t where $\overline{A_t} = \frac{1}{N}\sum_{i=1}^{N} A_t^{(i)}$. We can estimate the the difference of adjacency matrices in different states by corresponding difference of grouped-average matrices. That means with a set of trained $\overline{A_t}$ and

the individual adjacency matrix in resting-state, $A^{(i)}_{\text{rest}}$, we can predict the individual adjacency matrix in any state t that is included in the training set by $A^{(i)}_t = A^{(i)}_{\text{rest}} - (\overline{A_{\text{rest}}} - \overline{A_t})$.

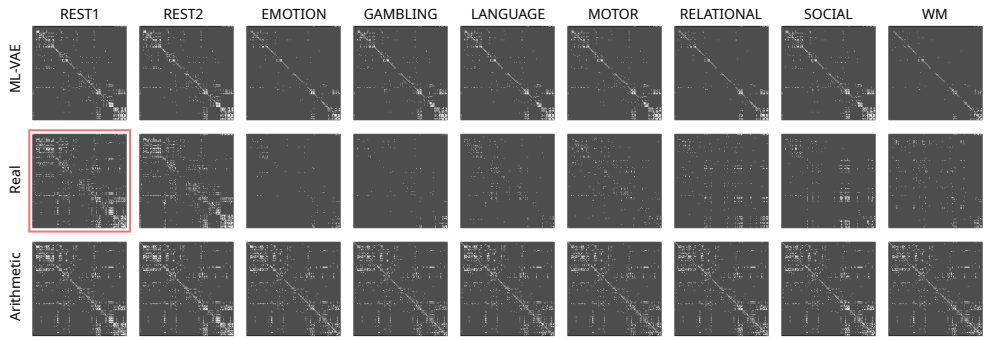


Figure 5.6. The comparison of real networks (row 2), ML-VAE reconstructed networks (row 1) and arithmetic method reconstructed networks (row 3). Both top row and bottom row are predicted by the individual adjacency matrix in resting-state (the one in red square). Closer to real networks in vertical means better prediction performance.

Figure 5.6 shows the prediction result of one individual. The brain networks in middle row are the objective networks for prediction. Our ML-VAE method is better than the baseline arithmetic method because it is able to reconstruct the networks in unseen states and predict the local structure. On the other side, the arithmetic method is closer to replications of resting-state networks because the networks in task state are sparse and information on task states are evened out by averaging.

Based on the prediction on 7 tasks of 20 subject in test set, we compare the difference between the predicted networks and objective real networks. Figure 5.7 shows the Euclidean distance and difference of graph metrics between predicted and objective adjacency matrices for ML-VAE and baseline arithmetic method. The ML-VAE outperforms baseline on all tasks in this comparison. The ML-VAE outperforms baseline on all metrics and tasks except for social task on assortativity and transitivity, and working memory task on transitivity. The performance of baseline on reconstructing resting-state 2 (REST2) is expected to be no worse than ML-VAE because REST2 is just a replication of REST1.

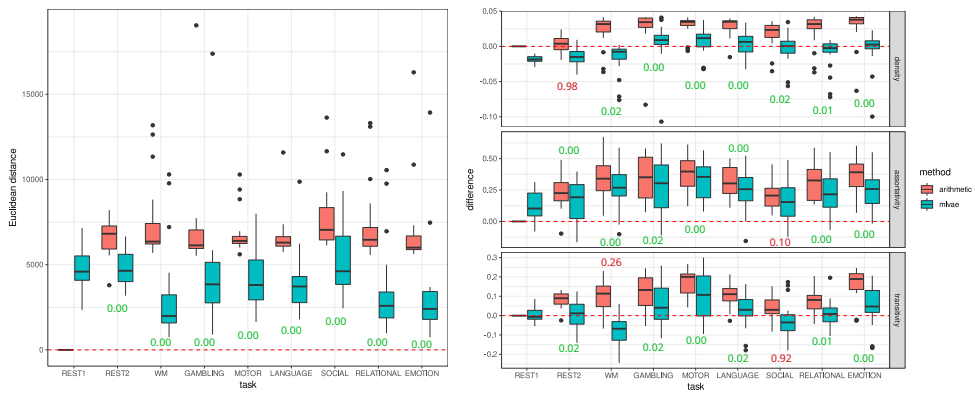


Figure 5.7. Euclidean distance of adjacency matrix and difference of density, assortativity of degree and transitivity comparison of ML-VAE and arithmetic method among 20 subjects. Closer to 0 (red dotted line) means better prediction on respective metric. The value above or below the boxes are p-value of t test between 2 methods. Values lower than 0.05 are shown on green otherwise red. Lower value means higher confidence that ML-VAE outperforms arithmetic method.

In addition, the prediction result varies for different input state for z_s. For instance, in the experiment above, the resting-state functional brain networks are taken as input network for extracting z_s since resting-state scan session is the longest session which makes more samples for correlation estimation. Also the resting-state connectivity is considered a good biomarker identification tool due to its reliability and reproducibility [129]. In the following experiment, the same experiment is repeated for cases that different states are served as individual latent variable z_s input. Then we compute the p-value as in Figure 5.7 to see if ML-VAE outperforms the baseline.

Figure 5.8 shows the result for the test. The null hypothesis that "ML-VAE is no better than arithmetic method (baseline) with respect to Euclidean distance between predicted and real adjacency matrix" is rejected for all tests, which means our method outperform baseline with all states as z_s encoding. [6]. Note that p-value is not a metric for evaluating the performance of prediction. We can conclude that, with

[6]The results for "self-prediction" are removed because baseline method gives exact input adjacency matrix. Mathematically, when we predict the resting-state with resting-state, $t = \text{rest}$, $A_t^{(i)} = A_{\text{rest}}^{(i)} - (\overline{A_{\text{rest}}} - \overline{A_t}) = A_{\text{rest}}^{(i)}$. Moreover, the result for predicting one resting-state network with the other resting-state network (e.g. predict "REST2" with "REST1") is removed for the same reason.

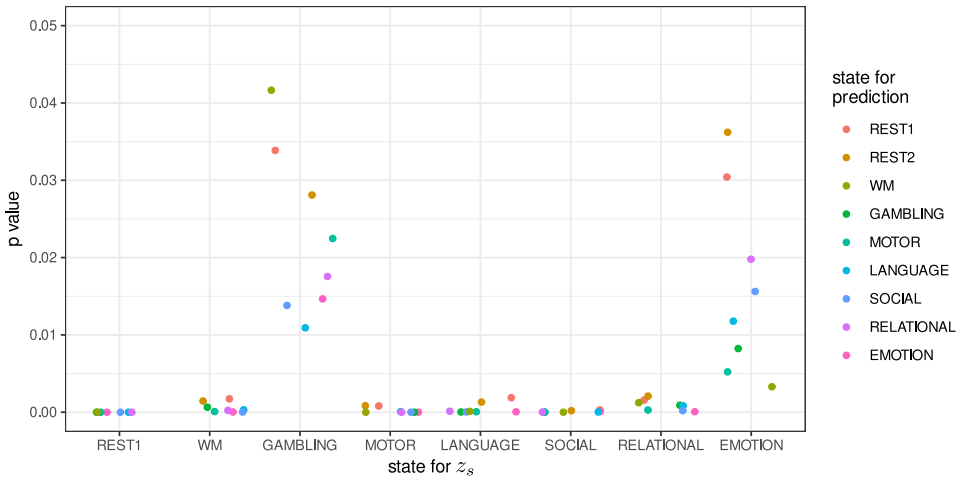


Figure 5.8. The p-value of t-test for comparing ML-VAE and baseline (arithmetic method) with different z_s encoding states. X axis represents 8 different states from which the ML-VAE encodes for z_s. Y axis is the p-value of the one sided t test that has null hypothesis "the prediction by baseline method is better than ML-VAE".

the same sample size, the null hypothesis can be rejected with the lowest threshold for resting-state functional brain network served as z_s.

5.4 Discussion

We have fitted a basic variational autoencoder (VAE) and a task-disentangled multi-level variational autoendoer (ML-VAE) on the functional brain networks obtained from the Pearson correlation matrices of 100 unrelated patients sampled from HCP data release in 8 different states. The basic VAE we implemented here has a simple one-layer encoder and decoder with 400 neurons in the hidden layer. It is shown that VAE can reproduce the distributional properties of functional brain networks, and ML-VAE can disentangle the task-related information in latent variable therefore predict the functional brain network in unseen states based on the network observed in resting-states.

For the basic VAE, we see in Figure 5.3 that the latent variable is closely related to the state of subject. Resting state, social task, language task are clustered separately

and the rest tasks are clustered in one group. A classification of target states can be done with the extracted latent variables as in [125]. Based on the prediction results in Figure 5.4 we can see the potential of this model on modeling the variability of network collections on different scale. In other words, this proposed model can predict better than baseline that (1) which area is activated under specific state; (2) what is the global/local topological structure of the activation in the network.

The task-disentangled prediction model predicts the connectivity in unseen states on the activation scale and topological scale as well. The proposed method outperforms baseline in most of the tasks. We choose the arithmetic method as baseline, although arithmetic method has an known issue on predicting networks in sparse networks with high variability. Future study includes finding a better baseline and include more topological metrics for evaluation. In our ML-VAE model, the topological patterns are not explicitly considered although the latent variable implicitly learned the features. To get better understanding of the topological structure, in the future we could include the topological information in our input explicitly as well.

6. CONCLUSION

In this thesis, we firstly emphasized the importance of variability in complex network generative process. The network collection is defined and analyzed in Chapter 3. The an action-based temporal model is proposed in Chapter 4, and a VAE network generative framework that is applied to general network collection is proposed in Chapter 5. The research findings solved the research questions we proposed in Section 1.2 along with the paradigm in Figure 6.1:

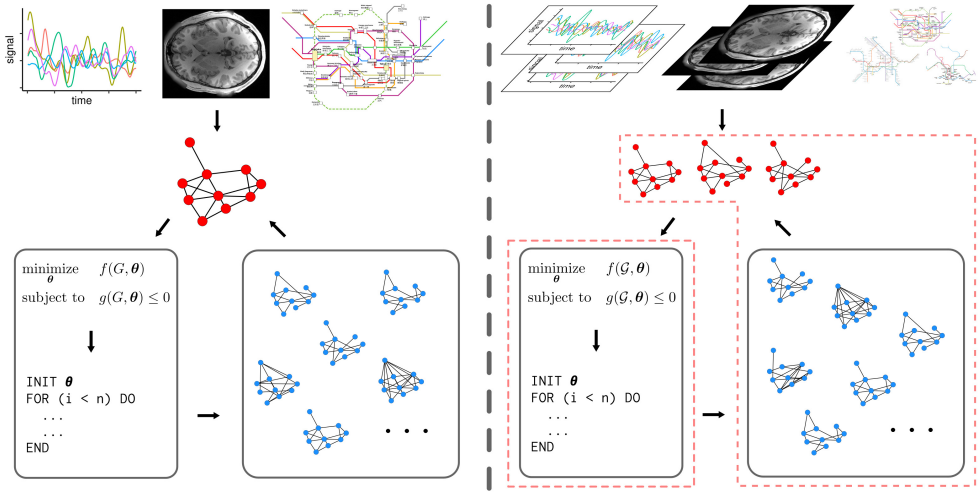


Figure 6.1. Paradigm of inferring generative model for a single network (left) and a network collection (right).

1. *Can we empirically quantify the variability of a network collection?*

   In Chapter 3 we defined a general form of network variability. From global topological measuring to local structure, 3 different types of measures are proposed and implemented on 3 sets of network collections. The 3 different measures have consistent results on the experimental datasets. A more general and extensible measuring framework on graphlets, entropic variability, is proposed and ana-

lyzed. It has shown that the entropic variability plays a crucial role in network generative models based on deep learning methods.

2. *Can we devise a model that matches the variability?*

   In Chapter 4, a specific case of network collection, temporal network, is discussed. An action-based model is implemented to the temporal functional brain network by inferring the edge switchings with its local topological structures. The synthesized networks are similar to the input network series, and it is shown that the preference on local structures are related to the functionality of the brain regions.

   In Chapter 5, a variational autoencoder (VAE) is implemented for functional brain networks as a general network generative models for network collection. The encoder in VAE has the ability to compress the input network into lower dimension latent variable. The trained decoder can reconstruct the latent variable to the network that keeps the distributional properties including variability of input network.

3. *What can we do to map the variability of network collection to interpretable features?*

   In Chapter 5, a multi-level VAE (ML-VAE) is proposed to disentangle the separate latent variables that are mapped to known network features. The state of subjects when taking fMRI scans are set as interpretable feature in the experiments. In ML-VAE, a specific latent variable that is shared by all networks under the same state stores the information of subjects' state. The experimental results show that ML-VAE can learn the variability of task/resting states and infer the functional brain networks of a subject in unknown states.

6.1 Limitation and Future Work

The variability studied in this dissertation is based on the assumption in Section 1.2 that *All networks in collection have same set of nodes.* This is realistic assumption for many scenario for example brain networks. However it is not true when we consider the node-evolving networks such as social network. At the beginning of Section 3 we know it is possible to relax this assumption to more general case that networks have changing nodes which is a potential future work for this dissertation. It requires a regularization step for networks with different set of nodes. In other words, some algorithm is able to cluster nodes or create nodes for individual graphs where all networks share the same set of nodes after the clustering or creation of nodes. At the same time, individual networks should keep the similar topological structure. Although there is no established way of regularizing networks into the same set, we do have tools to cluster nodes such as weighted graph cuts [85], network embedding with exponential family [86] and graph convolutional networks [114]. The main challenges include choosing a general regularization method and defining the variability change after the regularization of networks.

The temporal model introduced in Chapter 4 shows the change of an edge between two nodes over time is related to the local topological properties of both nodes. An action based method is implemented in the experiments. The advantage of the introduced action based model is intuitive however there is still room for improvements on the prediction accuracy. A potential future work is to utilize some neural network method to predict the edge dynamics. Artificial neural networks show the ability to improve the prediction accuracy with the cost of degree of freedom [130].

7. SUPPLEMENTARY MATERIALS

7.1 VAE on All States

In Section 5.2.2, the comparison of synthesized networks and input resting-state functional brain networks is shown and analyzed. The resting-state is chosen because 1) the scan session of resting-state is the longest session; 2) the resting-state is the most consistent session because in other tasks there exist activation and deactivation of functional brain regions due to the process of tasks. However, the application of VAE is not restricted to resting-state networks. The synthetic results of other tasks are shown in Figure 7.1

The graph density is fitted well for all 7 tasks. However, for most tasks we can find several outliers in real network collection that are not synthesized from VAE. Especially in task Working Memory (WM), there exist a group of outliers (for density metric) that are far away from the main group. The distribution of density in WM task is long-tailed and a potential explanation for the distribution is that people are varied on the memory ability and such distinction are not Gaussian distributed. VAE failed at capturing the long tail distribution due to its Gaussian distribution assumption on latent variable.

Overall the assortativity of degree is fitted well except for some degree of underestimation. Working memory task is the most underestimated one again compared to other tasks.

The transitivity are not fitted very well as the observation of resting-state in main text. Although the mean/median is not captured, we can see the variance of transitivity is captured very well. For most tasks, the distribution is "shifted" to the left due to the inherent drawbacks of low-dimensional embeddings of complex network [126].

Figure 7.1. The comparison of graph density, assortativity of degree and transitivity between input functional brain networks in 7 task states (notated on top left) and respective synthetic networks by VAE.

Figure 7.1. continued

Figure 7.1. continued

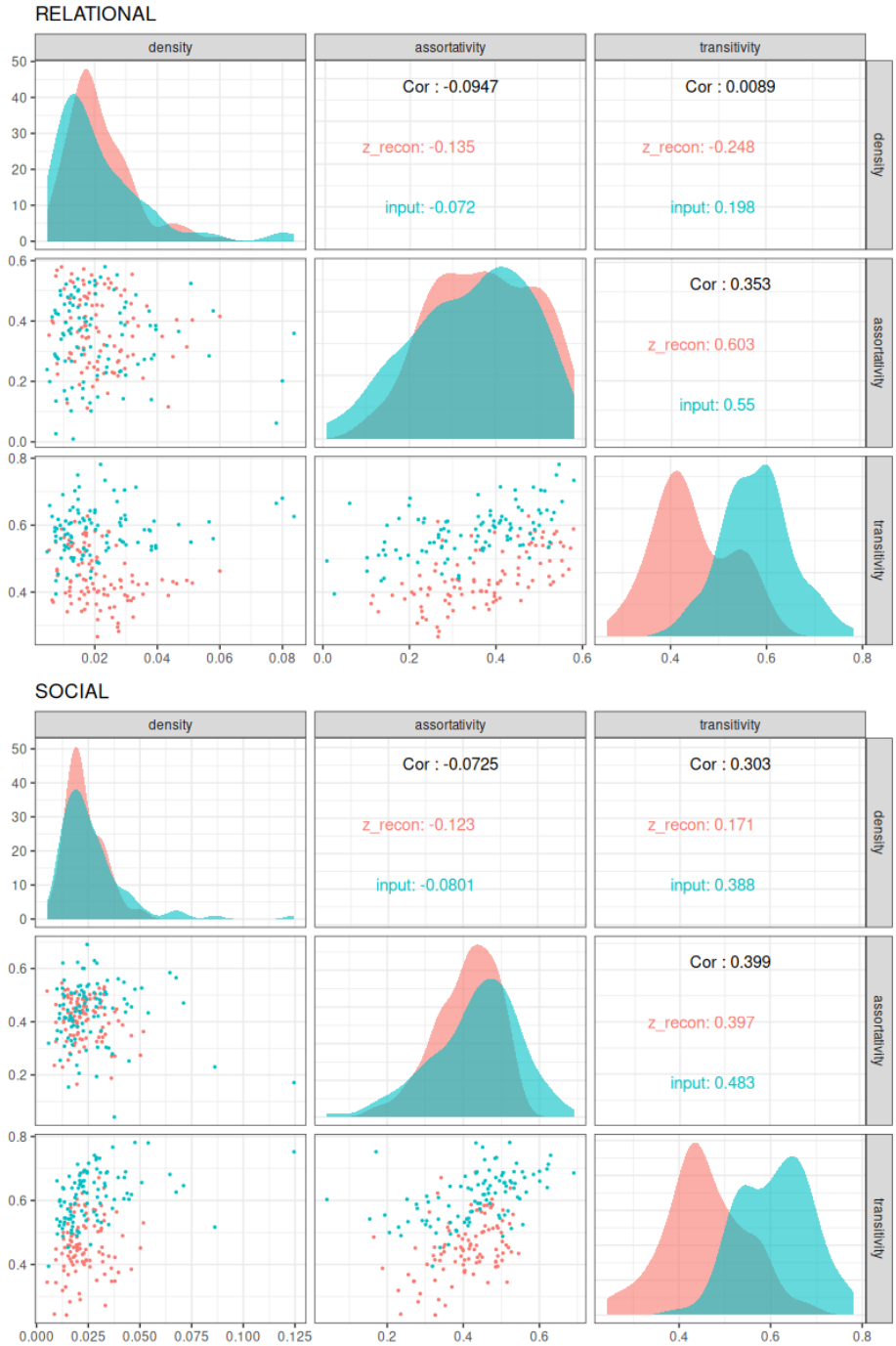
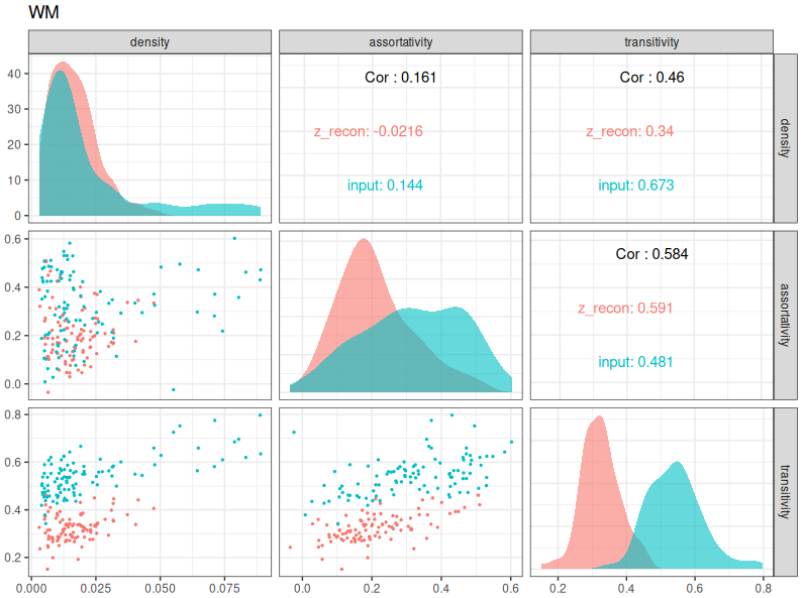


Figure 7.1. continued



7.2 Performance of VAE Encoder on Unsupervised Clustering

In Section 5.2.1, we found that the encoded z latent variable in trained VAE is a good indicator of the state during scan. In this section, the z latent variable clustered by its state is compared to a clustering by original adjacency matrix. If the z latent variable is able to extract the information of the input network, the represented networks should be more separable with respect to its state compared to adjacency matrix.

A simple VAE of 2-layer encoder and 2-layer decoder with a 20 dimension latent variable is trained for the functional brain networks of 100 subjects in 9 states[1]. With the trained VAE, 40 subjects in 9 different states are selected for the experiment in Section 5.2.1. For each session, 150 random consecutive frames are selected for all 360 ROIs, and the correlation matrix of the 360 ROIs are computed[2]. For the processed data set that consists of functional brain networks of 40 subjects in 9 states, we compressed both the encoded z latent variable and vectorized lower triangular adjacency matrix with t-SNE into 2 dimension for visualization.

Figure 7.2 shows the comparison result. The original vectorized adjacency matrix contains information of the states, and network in different states are separated to different areas. While encoded z shows a better separability which indicates the information of state is extracted and enlarged during the encoding process. The networks in social recognition task are most distinct in both methods, while VAE shows its ability to separate resting-state from the large group.

[1]Refer to Section 5.2.2 for details on VAE architecture.
[2]Check Section 5.2.1 for explanation and experiment setup details

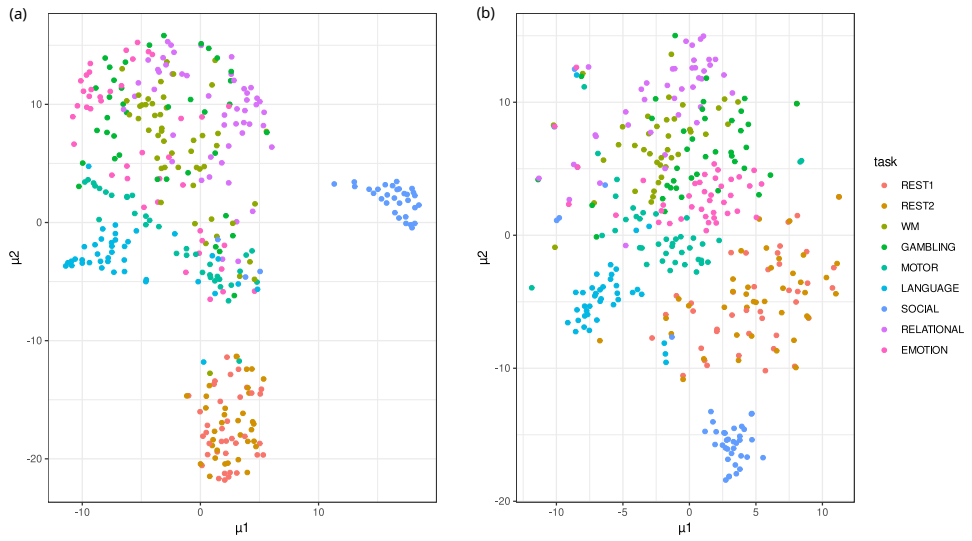


Figure 7.2. The comparison of encoded z latent variable and vectorized adjacency matrix after t-SNE. (a) encoded z after t-SNE; (b) vectorized lower triangular matrix after t-SNE. In both (a) and (b) there are 360 points representing 360 networks (40 subjects in 9 states). 9 states are represented by 9 colors where REST1 and REST2 are same resting state in replicated sessions. More distance between clusters implies better separability.

7.3 Network Prediction Based on Task State Networks by MLVAE

In Section 5.3.1, the functional brain network of individuals under unseen states are predicted based on the individual latent variable extracted from resting-state network. The reason for choosing resting-state networks as input is that resting-state has longest scan session and most stable scan condition (subjects are asked to perform specific task and rest during task sessions). In this section, the predicted functional brain networks in unseen states based on other task sessions are supplemented and analyzed.

Figure 7.3 and Table 7.1 shows the Euclidean distance by input states. The results with resting-state as input outperformed all other results. This result indicates that resting-state is the best state for extract the individual latent variable. One potential reason is resting-state has the longest scan time therefore the correlation is the most consistent and stable. Second potential reason is the individual discrepancy in functional connectivity is higher when people conducting some task than people in resting state. In other words, people are more similar when stay still than doing task.

Table 7.1. p value of one-side t-test for checking how much ML-VAE outperforms baseline. Lower value means better prediction on Euclidean distance compared to baseline. States in rows are input states for z_t and states in columns are prediction states.

	rest1	rest2	emotion	gambling	language	motor	relational	social	wm
rest1	1.0000	0.0042	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
emotion	0.0133	0.0181	0.9979	0.0084	0.0112	0.0048	0.0189	0.0134	0.0027
gambling	0.0162	0.0145	0.0206	0.9965	0.0116	0.0227	0.0178	0.0123	0.0354
language	0.0007	0.0005	0.0001	0.0000	1.0000	0.0001	0.0001	0.0000	0.0002
motor	0.0003	0.0003	0.0000	0.0000	0.0001	1.0000	0.0000	0.0000	0.0000
relational	0.0004	0.0006	0.0001	0.0010	0.0010	0.0003	0.9999	0.0002	0.0015
social	0.0003	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000
wm	0.0006	0.0005	0.0001	0.0007	0.0004	0.0001	0.0003	0.0000	0.9999

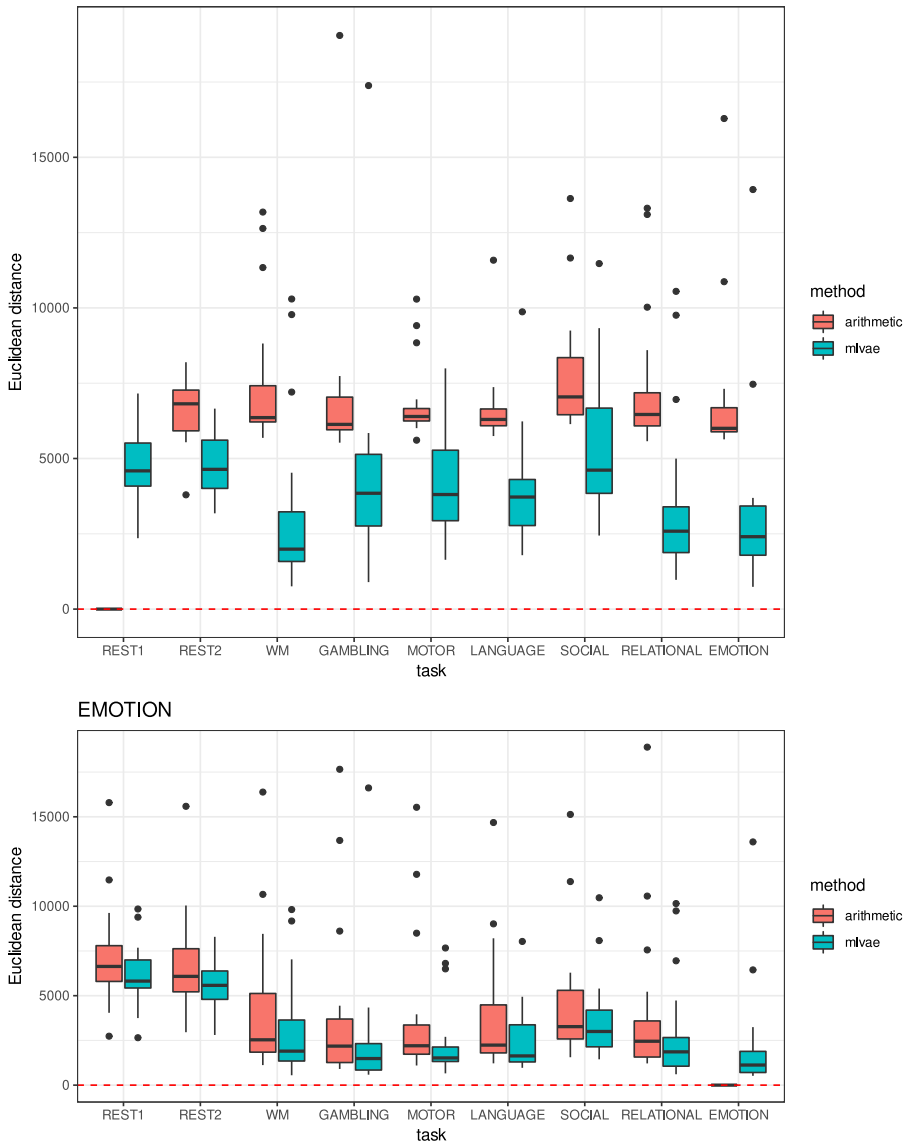


Figure 7.3. The Euclidean distance between synthesized networks and real networks by input states. The one on the top is result with resting-state as input. The results for tasks follow below with input states notated on the top left.

Figure 7.3. continued

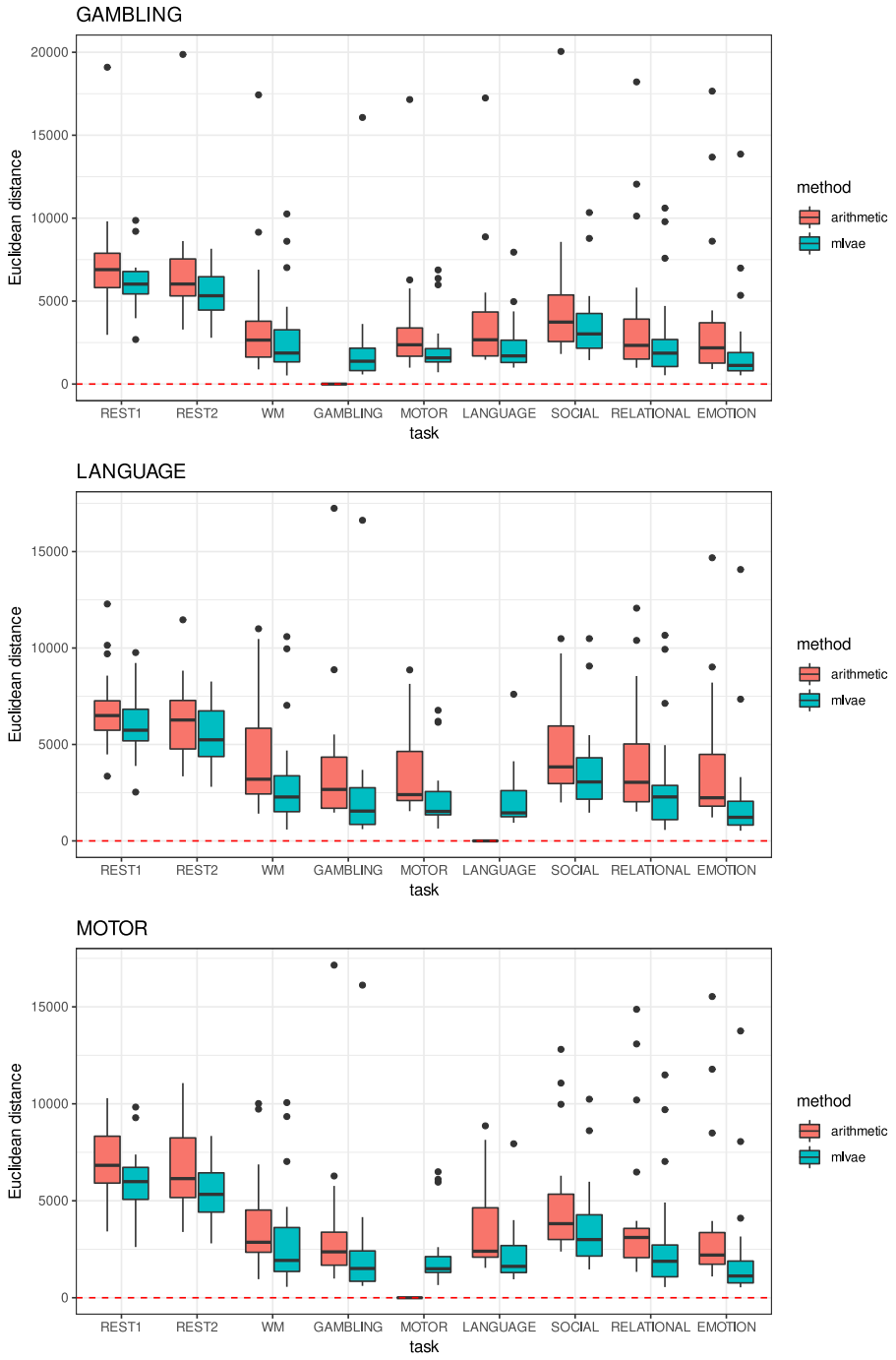
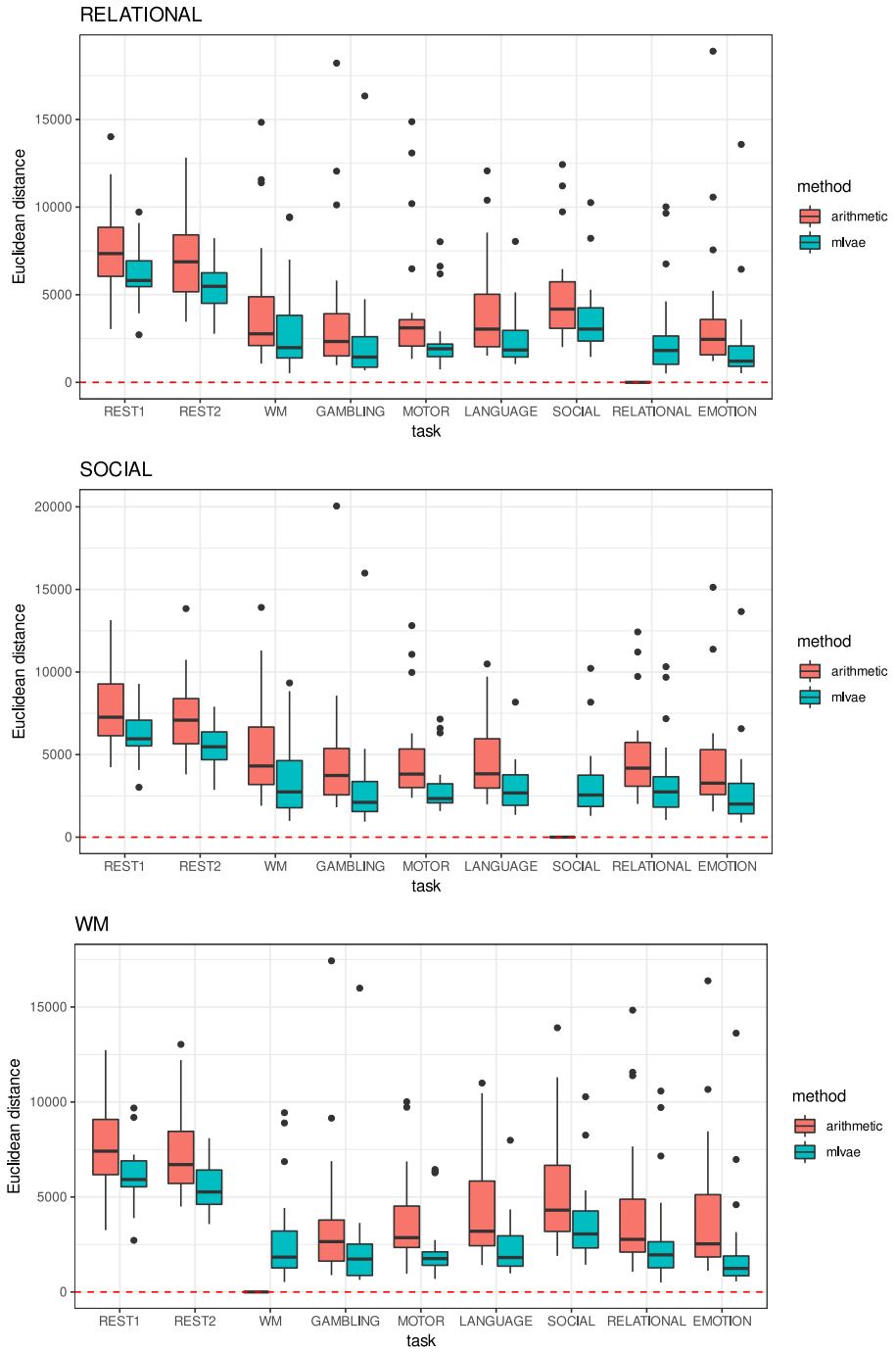


Figure 7.3. continued



REFERENCES

REFERENCES

[1] Mark Newman. *Networks*. Oxford University Press.

[2] John Scott and Peter J. Carrington. *The SAGE Handbook of Social Network Analysis*. SAGE.

[3] Alex Fornito, Andrew Zalesky, and Edward Bullmore. *Fundamentals of Brain Network Analysis*. Academic Press.

[4] K. J Kansky. Structure of transportation networks : Relationships between network geometry and regional characteristics.

[5] James M. Bower and Hamid Bolouri. *Computational Modeling of Genetic and Biochemical Networks*. MIT Press.

[6] William A. Woods. WHAT'S IN A LINK: Foundations for Semantic Networks. In DANIEL G. Bobrow and ALLAN Collins, editors, *Representation and Understanding*, pages 35–82. Morgan Kaufmann.

[7] Alex Fornito, Andrew Zalesky, and Michael Breakspear. The connectomics of brain disorders. 16(3):159–172.

[8] Fabrice Wendling, Pascal Benquet, Fabrice Bartolomei, and Viktor Jirsa. Computational models of epileptiform activity. 260:233–251.

[9] Lenny E. Ramsey, Joshua S. Siegel, Antonello Baldassarre, Nicholas V. Metcalf, Kristina Zinn, Gordon L. Shulman, and Maurizio Corbetta. Normalization of network connectivity in hemispatial neglect recovery. 80(1):127–141.

[10] Christopher J. Honey, Rolf Kötter, Michael Breakspear, and Olaf Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. 104(24):10240–10245.

[11] James Ladyman, James Lambert, and Karoline Wiesner. What is a complex system? 3(1):33–67.

[12] Steven H. Strogatz. Exploring complex networks. 410(6825):268–276.

[13] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. 424(4):175–308.

[14] Fan R. K. Chung and Fan Chung Graham. *Spectral Graph Theory*. American Mathematical Soc.

[15] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856.

[16] Chris Godsil and Gordon F. Royle. *Algebraic Graph Theory*. Springer Science & Business Media.

[17] Ray Solomonoff and Anatol Rapoport. Connectivity of random nets. 13(2):107–117.

[18] E. N. Gilbert. Random Graphs. 30(4):1141–1144.

[19] P. Erdös and A. Rényi. On random graphs I. 6:290.

[20] Ove Frank and David Strauss. Markov Graphs. 81(395):832–842.

[21] Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph (p*) models for social networks. 29(2):173–191.

[22] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent Space Approaches to Social Network Analysis. 97(460):1090–1098.

[23] Chiara Orsini, Marija M. Dankulov, Pol Colomer-de Simón, Almerima Jamakovic, Priya Mahadevan, Amin Vahdat, Kevin E. Bassler, Zoltán Toroczkai, Marián Boguñá, Guido Caldarelli, Santo Fortunato, and Dmitri Krioukov. Quantifying randomness in real networks. 6:8627.

[24] Viplove Arora, Dali Guo, Katherine D. Dunbar, and Mario Ventresca. Examining the variability in network populations and its role in generative models. pages 1–22.

[25] David Chandler. Introduction to Modern Statistical Mechanics.

[26] Juyong Park and M. E. J. Newman. Statistical mechanics of networks. 70(6):066117.

[27] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The Human Connectome: A Structural Description of the Human Brain. 1(4):e42.

[28] Joan Stiles and Terry L. Jernigan. The Basics of Brain Development. 20(4):327–348.

[29] RHOSHEL K. LENROOT and JAY N. GIEDD. The changing impact of genes and environment on brain development during childhood and adolescence: Initial findings from a neuroimaging study of pediatric twins. 20(4):1161–1175.

[30] Andreas Daffertshofer, Claudine J. C. Lamoth, Onno G. Meijer, and Peter J. Beek. PCA in studying coordination and variability: A tutorial. 19(4):415–428.

[31] Abdulhamit Subasi and M. Ismail Gursoy. EEG signal classification using PCA, ICA, LDA and support vector machines. 37(12):8659–8666.

[32] Eric D. Kolaczyk. *Topics at the Frontier of Statistics and Network Analysis: (Re)Visiting the Foundations*. SemStat Elements. Cambridge University Press.

[33] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

[34] F. J. Roethlisberger and William J. Dickson. *Management and the Worker*. Psychology Press.

[35] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. 2(3):203–271.

[36] D. Dunlavy, T. Kolda, and W. Kegelmeyer. 7. Multilinear Algebra for Analyzing Data with Multiple Linkages. In *Graph Algorithms in the Language of Linear Algebra*, Software, Environments and Tools, pages 85–114. Society for Industrial and Applied Mathematics.

[37] T. Kolda and B. Bader. Tensor Decompositions and Applications. 51(3):455–500.

[38] E. Acar and B. Yener. Unsupervised Multiway Data Analysis: A Literature Survey. 21(1):6–20.

[39] Eugenio Valdano, Luca Ferreri, Chiara Poletto, and Vittoria Colizza. Analytical Computation of the Epidemic Threshold on Temporal Networks. 5(2):021005.

[40] Sergey V. Buldyrev, Roni Parshani, Gerald Paul, H. Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. 464(7291):1025–1028.

[41] Sebastian Funk and Vincent A. A. Jansen. Interacting epidemics on overlay networks. 81(3):036118.

[42] Anna Saumell-Mendiola, M. Ángeles Serrano, and Marián Boguñá. Epidemic spreading on interconnected networks. 86(2):026106.

[43] F. D. Sahneh, C. Scoglio, and F. N. Chowdhury. Effect of coupling on the epidemic threshold in interconnected complex networks: A spectral analysis. In *2013 American Control Conference*, pages 2307–2312.

[44] Albert Solé-Ribalta, Manlio De Domenico, Sergio Gómez, and Alex Arenas. Random walk centrality in interconnected multilayer networks. 323-324:73–79.

[45] Jianxi Gao, Sergey V. Buldyrev, Shlomo Havlin, and H. Eugene Stanley. Robustness of a Network of Networks. 107(19):195701.

[46] V. Nicosia, G. Bianconi, V. Latora, and M. Barthelemy. Growing Multiplex Networks. 111(5):058701.

[47] Ginestra Bianconi. Statistical mechanics of multiplex networks: Entropy and overlap. 87(6):062806.

[48] Federico Battiston, Vincenzo Nicosia, and Vito Latora. Structural measures for multiplex networks. 89(3):032804.

[49] Petter Holme and Jari Saramäki. Temporal networks. 519(3):97–125.

[50] Danielle S. Bassett, Mason A. Porter, Nicholas F. Wymbs, Scott T. Grafton, Jean M. Carlson, and Peter J. Mucha. Robust detection of dynamic community structure in networks. 23(1):013142.

[51] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. 328(5980):876–878.

[52] Albert Solé-Ribalta, Manlio De Domenico, Sergio Gómez, and Alex Arenas. Centrality Rankings in Multiplex Networks. In *Proceedings of the 2014 ACM Conference on Web Science*, WebSci '14, pages 149–155. ACM.

[53] Manlio De Domenico, Albert Solé-Ribalta, Sergio Gómez, and Alex Arenas. Navigability of interconnected networks under random failures. 111(23):8351–8356.

[54] Emanuele Cozzo, Mikko Kivelä, Manlio De Domenico, Albert Solé-Ribalta, Alexandre Arenas, Sergio Gómez, Mason A. Porter, and Yamir Moreno. Clustering Coefficients in Multiplex Networks. abs/1307.6780.

[55] E. Estrada and J. Gómez-Gardeñes. Communicability reveals a transition to coordinated behavior in multiplex networks. 89(4):042819–042819.

[56] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. 6(2-3):161–180.

[57] Peng Wang, Garry Robins, Philippa Pattison, and Emmanuel Lazega. Exponential random graph models for multilevel networks. 35(1):96–115.

[58] Michael T. Heaney. Multiplex networks and interest group influence reputation: An exponential random graph model. 36:66–81.

[59] Ginestra Bianconi. The entropy of randomized network ensembles. 81(2):28005.

[60] Ginestra Bianconi. Entropy of network ensembles. 79(3):036114.

[61] E. T. Jaynes. Information Theory and Statistical Mechanics. II. 108(2):171–190.

[62] E. T. Jaynes. Information Theory and Statistical Mechanics. 106(4):620–630.

[63] I. Artico, I. Smolyarenko, V. Vinciotti, and E. C. Wit. How rare are power-law networks really? 476(2241):20190742.

[64] Mark S Handcock, Garry Robins, Tom Snijders, Jim Moody, and Julian Besag. Assessing degeneracy in statistical models of social networks.

[65] Martina Morris, Mark S. Handcock, and David R. Hunter. Specification of Exponential-Family Random Graph Models: Terms and Computational Aspects. 24(4):1548–7660.

[66] Viplove Arora and Mario Ventresca. Action-based Modeling of Complex Networks. 7(1):1–10.

[67] Edward A Bender and E. Rodney Canfield. The asymptotic number of labeled graphs with given degree sequences. 24(3):296–307.

[68] Enrico Amico and Joaquín Goñi. Mapping hybrid functional-structural connectivity traits in the human connectome. 2(3):306–322.

[69] Sebastian Moreno and Jennifer Neville. An investigation of the distributional characteristics of generative graph models. In *Proceedings of the 1st Workshop on Information in Networks*.

[70] S. Moreno, S. Kirshner, J. Neville, and S. V. N. Vishwanathan. Tied Kronecker product graph models to capture variance in network populations. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1137–1144.

[71] Enrico Amico and Joaquín Goñi. The quest for identifiability in human functional connectomes. 8(1).

[72] David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E.J. Behrens, Essa Yacoub, and Kamil Ugurbil. The WU-Minn Human Connectome Project: An overview. 80:62–79.

[73] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images. 17(2):825–841.

[74] Matthew F. Glasser, Stamatios N. Sotiropoulos, J. Anthony Wilson, Timothy S. Coalson, Bruce Fischl, Jesper L. Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R. Polimeni, David C. Van Essen, and Mark Jenkinson. The minimal preprocessing pipelines for the Human Connectome Project. 80:105–124.

[75] J Xu, S Moeller, J Strupp, E Auerbach, L Chen, DA Feinberg, K Ugurbil, and E Yacoub. Highly accelerated whole brain imaging using aligned-blipped-controlled-aliasing multiband EPI. In *Proceedings of the 20th Annual Meeting of ISMRM*, volume 2306.

[76] Kawin Setsompop, Borjan A. Gagoski, Jonathan R. Polimeni, Thomas Witzel, Van J. Wedeen, and Lawrence L. Wald. Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty. 67(5):1210–1224.

[77] David A. Feinberg, Steen Moeller, Stephen M. Smith, Edward Auerbach, Sudhir Ramanna, Matt F. Glasser, Karla L. Miller, Kamil Ugurbil, and Essa Yacoub. Multiplexed Echo Planar Imaging for Sub-Second Whole Brain FMRI and Fast Diffusion Imaging. 5(12):e15710.

[78] Steen Moeller, Essa Yacoub, Cheryl A. Olman, Edward Auerbach, John Strupp, Noam Harel, and Kâmil Uğurbil. Multiband multislice GE-EPI at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fMRI. 63(5):1144–1153.

[79] Jonathan D. Power, Anish Mitra, Timothy O. Laumann, Abraham Z. Snyder, Bradley L. Schlaggar, and Steven E. Petersen. Methods to detect, characterize, and remove motion artifact in resting state fMRI. 84:320–341.

[80] Dongyang Zhang and Marcus E. Raichle. Disease and the brain's dark energy. 6(1):15–28.

[81] Elizabeth M. C. Hillman. Coupling Mechanism and Significance of the BOLD Signal: A Status Report. 37:161.

[82] Danielle S. Bassett and Olaf Sporns. Network neuroscience. 20(3):353–364.

[83] Matthew F. Glasser, Timothy S. Coalson, Emma C. Robinson, Carl D. Hacker, John Harwell, Essa Yacoub, Kamil Ugurbil, Jesper Andersson, Christian F. Beckmann, Mark Jenkinson, Stephen M. Smith, and David C. Van Essen. A multi-modal parcellation of human cerebral cortex. 536(7615):171–178.

[84] Stephen M. Smith, Karla L. Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F. Beckmann, Thomas E. Nichols, Joseph D. Ramsey, and Mark W. Woolrich. Network modelling methods for FMRI. 54(2):875–891.

[85] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted Graph Cuts without Eigenvectors A Multilevel Approach. 29(11):1944–1957.

[86] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1067–1077. International World Wide Web Conferences Steering Committee.

[87] B. T. Thomas Yeo, Fenna M. Krienen, Jorge Sepulcre, Mert R. Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L. Roffman, Jordan W. Smoller, Lilla Zöllei, Jonathan R. Polimeni, Bruce Fischl, Hesheng Liu, and Randy L. Buckner. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. 106(3):1125–1165.

[88] Ed Bullmore and Olaf Sporns. Complex brain networks: Graph theoretical analysis of structural and functional systems. 10(3):186–198.

[89] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: Uses and interpretations. 52(3):1059–1069.

[90] M. E. J. Newman. Coauthorship networks and patterns of scientific collaboration. 101:5200–5205.

[91] Andrea Avena-Koenigsberger, Joaquín Goñi, Ricard Solé, and Olaf Sporns. Network morphospace. 12(103):20140881.

[92] Joaquín Goñi, Andrea Avena-Koenigsberger, Nieves Velez de Mendizabal, Martijn P. van den Heuvel, Richard F. Betzel, and Olaf Sporns. Exploring the Morphospace of Communication Efficiency in Complex Networks. 8(3):e58070.

[93] Alex Fornito, Andrew Zalesky, and Michael Breakspear. Graph analysis of the human connectome: Promise, progress, and pitfalls. 80:426–444.

[94] Olaf Sporns. Contributions and challenges for network models in cognitive neuroscience. 17(5):652–660.

[95] A. Sanfeliu and K. Fu. A distance measure between attributed relational graphs for pattern recognition. SMC-13(3):353–362.

[96] John W. Raymond, Eleanor J. Gardiner, and Peter Willett. RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs. 45(6):631–644.

[97] M. L. Menéndez, J. A. Pardo, L. Pardo, and M. C. Pardo. The Jensen-Shannon divergence. 334(2):307–318.

[98] H. Bunke. On a relation between graph edit distance and maximum common subgraph. 18(8):689–694.

[99] Mattia Tantardini, Francesca Ieva, Lucia Tajoli, and Carlo Piccardi. Comparing methods for comparing networks. 9(1):17557.

[100] Natasa Pržulj, Derek G Corneil, and Igor Jurisica. Modeling interactome: Scale-free or geometric? 20(18):3508–3515.

[101] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, pages 488–495.

[102] M. Rahman, M. A. Bhuiyan, and M. Al Hasan. Graft: An efficient graphlet counting method for large graph analysis. 26(10):2466–2478.

[103] M. Terrenoire and D. Tounissoux. Sample Size Sensitive Entropy. In Josef Kittler, King Sun Fu, and Louis-François Pau, editors, *Pattern Recognition Theory and Applications*, NATO Advanced Study Institutes Series, pages 53–62. Springer Netherlands.

[104] Dali Guo, Viplove Arora, Enrico Amico, Joaquín Goñi, and Mario Ventresca. Dynamic generative model of the human brain in resting-state. In Chantal Cherifi, Hocine Cherifi, Márton Karsai, and Mirco Musolesi, editors, *Complex Networks & Their Applications VI*, pages 1271–1283. Springer International Publishing.

[105] Petra E. Vértes, Aaron F. Alexander-Bloch, Nitin Gogtay, Jay N. Giedd, Judith L. Rapoport, and Edward T. Bullmore. Simple models of human brain functional networks. 109(15):5868–5873.

[106] Richard F. Betzel, Andrea Avena-Koenigsberger, Joaquín Goñi, Ye He, Marcel A. de Reus, Alessandra Griffa, Petra E. Vértes, Bratislav Mišic, Jean-Philippe Thiran, Patric Hagmann, Martijn van den Heuvel, Xi-Nian Zuo, Edward T. Bullmore, and Olaf Sporns. Generative models of the human connectome. 124:1054–1064.

[107] M. E. J. Newman. Modularity and community structure in networks. 103(23):8577–8582.

[108] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes.

[109] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar Variational Autoencoder.

[110] Yaxiang Fan, Gongjian Wen, Deren Li, Shaohua Qiu, Martin D. Levine, and Fei Xiao. Video anomaly detection and localization via Gaussian Mixture Fully Convolutional Variational Autoencoder. 195:102920.

[111] A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. 8(3):714–735.

[112] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. 20(1):61–80.

[113] Alessio Micheli. Neural Network for Graphs: A Contextual Constructive Approach. 20(3):498–511.

[114] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks.

[115] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep Sets. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3391–3401. Curran Associates, Inc.

[116] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks.

[117] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. 37(2):233–243.

[118] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. 11:3371–3408.

[119] Carl Doersch. Tutorial on Variational Autoencoders.

[120] Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders.

[121] Thomas N Kipf and Max Welling. Variational graph auto-encoders.

[122] Danielle S. Bassett, Perry Zurn, and Joshua I. Gold. On the nature and use of models in network neuroscience. 19(9):566–578.

[123] O. Parker Jones, N.L. Voets, J.E. Adcock, R. Stacey, and S. Jbabdi. Resting connectivity predicts task activation in pre-surgical populations. 13:378–385.

[124] Alexander D. Cohen, Ziyi Chen, Oiwi Parker Jones, Chen Niu, and Yang Wang. Regression-based machine-learning approaches to predict task activation using resting-state fMRI. 41(3):815–826.

[125] Javier Rasero, Hannelore Aerts, Marlis Ontivero Ortega, Jesus M. Cortes, Sebastiano Stramaglia, and Daniele Marinazzo. Predicting functional networks from region connectivity profiles in task-based versus resting-state fMRI data. 13(11):e0207385.

[126] C. Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-rank representations for triangle-rich complex networks. 117(11):5631–5637.

[127] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-Level Variational Autoencoder: Learning Disentangled Representations From Grouped Observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[128] Benjamin A. Seitzman, Caterina Gratton, Timothy O. Laumann, Evan M. Gordon, Babatunde Adeyemo, Ally Dworetsky, Brian T. Kraus, Adrian W. Gilmore, Jeffrey J. Berg, Mario Ortega, Annie Nguyen, Deanna J. Greene, Kathleen B. McDermott, Steven M. Nelson, Christina N. Lessov-Schlaggar, Bradley L. Schlaggar, Nico U. F. Dosenbach, and Steven E. Petersen. Trait-like variants in human functional brain networks. 116(45):22851–22861.

[129] Xi-Nian Zuo and Xiu-Xia Xing. Test-retest reliabilities of resting-state FMRI measurements in human brain functional connectomics: A systems neuroscience perspective. 45:100–118.

[130] Nathan O. Hodas and Panos Stinis. Doing the Impossible: Why Neural Networks Can Be Trained at All. 9.