Dynamic Mode Decomposition and Network Evolution

A Thesis

Presented to the

Faculty of

San Diego State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Applied Mathematics

with a Concentration in

Dynamical Systems

---

by

Robert Simpson

Spring 2021

SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Robert Simpson:

Dynamic Mode Decomposition

and Network Evolution

Christopher Curtis, Chair
Department of Mathematics and Statistics

Christopher O'Neill
Department of Mathematics and Statistics

Xialu Liu
Department of Management Information Systems

Approval Date

DEDICATION

To my family, who have been wholly supportive.

ABSTRACT OF THE THESIS

Dynamic Mode Decomposition
and Network Evolution
by
Robert Simpson
Master of Science in Applied Mathematics with a Concentration in Dynamical Systems
San Diego State University, 2021

Complex networks are structurally non-trivial and require a large set of tools to analyze their characteristics. In this thesis, we implement a standard statistical covariance method to analyze local network structure. In addition, we implement the data-driven methods: Dynamic Mode Decomposition (DMD) and Kernel Dynamic Mode Decomposition (KDMD). These methods are grounded in Koopman theory and give us a dynamical systems perspective into network development. With feature matrices built from snapshots of motif counts throughout a network's development, we characterize the dynamic behavior of the local network structure. Through DMD and KDMD, we identify sets of DMD and KDMD modes. Analyzing the modes, we identify spatiotemporal coherent structures in the data. The DMD and KDMD algorithms produce modes of low error, which are good approximations to true Koopman modes.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to thank Professor Chris Curtis for his invaluable insight and mentorship throughout the making of this thesis. I would also like to thank Professor Chris O'Neill and Professor Xialu Liu for taking part as members of the thesis committee. Thank you to the Professors of the SDSU Department of Mathematics and Statistics, who have encouraged my learning and development in applied mathematics these last several years. Finally, thank you to Samuel Cordaro, Sierra Cordaro, Sergio Contreras, and Austin Simpson, who helped me proof-read and improve the overall quality of this thesis.

This thesis is partially protected against the evil forces of the Montezuma Publishing thesis reviewers by the magic of the Department of Mathematics and Statistics Master's Thesis LaTeX Template.

CHAPTER 1

Motifs

Motifs are the primary object of interest as a way of characterizing the local structure of a network. The dynamic motif count indicates the network and the communities within are evolving according to certain rules. In the context of static networks, the frequency of motifs has been shown to highlight network properties [1] [18].

Motifs are the fundamental components of complex networks. The topological structure of complex networks is tied to the frequency and distribution of motifs. Initially, the counting of these motifs was treated as a static problem where the frequencies of motifs affect functions on networks [13]. As the significance of motifs has become more apparent, authors have begun examining the emergence of motifs via temporal edges [17]. Our motif counts will act as a vector of features characterizing the network at each point in time. Before we examine our specific motifs, we must first define common objects of graph theory such as cycles, paths, and stars which will aid us in understanding the motifs we would like to examine.

1.1 Graph Theory

In the pursuit of counting our particular motifs, we require some ideas from graph theory. We will count three-cycles, four-cycles, and five-cycles as motifs, but we also require the non-simple cycle motifs. Much of the literature on motifs focuses on directed triangle motifs [15] [17] [19], but the motifs can take on other shapes. The motifs in this thesis are undirected but more varied in edge and node count. Graphs are the common language of motifs. We define a graph G by the following definition.

Definition 1. *Let $G = (V, E)$ be a graph with V being a set of vertices (or nodes), and E, a set of edges. If v is a vertex of G we write $v \in V(G)$. If $u, v \in V(G)$ and there is an edge between them we write $\{u, v\} \in E(G)$*

Graphs may be directed or undirected. For directed graphs, $\{u, v\} \in E(G)$ is taken to mean there exists an edge from u to v in G. Undirected implies the edge between u and v has no notion of direction.

Definition 2. *The adjacency matrix A for any graph G with $n = |V|$ vertices is a matrix of size $n \times n$. The element $a_{i,j}$ is defined to be*

$$a_{i,j} = \begin{cases} 1 & e_{i,j} \in E(G) \\ 0 & e_{i,j} \notin E(G) \end{cases}$$

The adjacency matrix is critical to all of our calculations to come. It renders any graph amenable to the tools of linear algebra.

Definition 3. *We call $f : G \to H$ a homomorphism, if f maps endpoints in $G = (V(G), E(G))$ to endpoints in $H = (V(H), E(H))$. i.e.*
$\forall u, v \in V(G) \quad \{u, v\} \in E(G) \Rightarrow \{f(u), f(v)\} \in E(H)$.

For example, in Figure 1.1 we can define a mapping f such that:

$$f(0) = A$$
$$f(1) = A$$
$$f(2) = B$$
$$f(3) = C$$

Thus f is a homomorphism by Definition 3. We now define the isomorphism and automorphism.

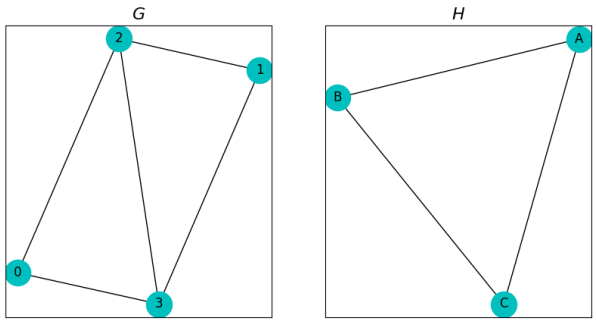

Figure 1.1. A homomorphism from G to H.

Definition 4. *We call $f : G \to H$ an isomorphism if f is a homomorphism and is bijective.*

We can define an isomorphism f between G and H in the figure 1.2 such that:

$$g(0) = B$$
$$g(1) = A$$
$$g(2) = C$$
$$g(3) = D$$

Definition 5. *An automorphism is an isomorphism between a graph G and itself. The automorphism is edge-preserving, $\{u, v\} \in G \implies \{f(v), f(u)\} \in G$.*

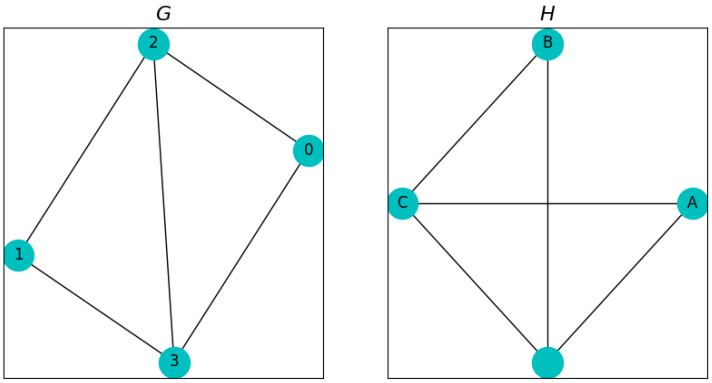

Figure 1.2. An isomorphism between G and H.

Automorphisms represent the symmetries of a graph. For instance, the graph H of vertices $\{A, B, C\}$ and the set of undirected edges $\{A, B\}, \{B, C\}, \{C, A\}$, has six possible automorphisms - three from rotation and three from reflection of the graph. When counting motifs on graphs, the algorithms only count up to symmetry. For instance, where we find an induced subgraph isomorphic the graph H we count the subgraph as a single appearance of the triangle motif.

Definition 6. *Let $G = (V, E)$ be a graph. We call $G' = (V', E')$ a subgraph, denoted $G' \subset G$, if $V' \subseteq V \wedge E' \subseteq E \cap (V' \times V')$. Furthermore we call G' an induced subgraph of G if $\forall u, v \in V$ we have $\{u, v\} \in E \iff \{u, v\} \in E'$.*

Induced subgraphs are vital to our understanding of how motifs interact as we add edges or nodes to a given graph. In Figure 1.3, G has a subgraph which does not

include the edge $\{1, 2\}$. If the edge is included in the set, then we have an induced subgraph of G.

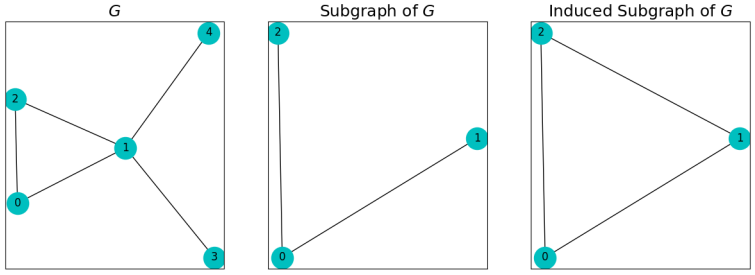

Figure 1.3. Example of subgraphs and induced subgraphs on G.

Definition 7. *Let $G'' \subset G$ and furthermore let there exist an isomorphism between G'' and G'. We call G'' an appearance of G'. Provided the number of appearances of G' is greater than some N we call G' a motif or pattern. The count of G' refers to the number of appearances of G' in G.*

Definition 8. *A walk $W = \{v_0, e_1, v_1, \ldots v_n\}$ is a sequence of vertices and edges of G such that for $0 \leq k \leq n-1$, $e_i = \{v_k, v_k + 1\}$.*

Definition 9. *A cycle Cn is a walk of n vertices, whose first and last vertex are the same.*

Simple Cycles (cycles where every vertex, except the first and last, are unique) are often also referred to by the geometric objects they resemble. A three-cycle is a triangle and a four-cycle is a square. The three-cycle, four-cycle, and five-cycle will feature as motifs in our feature vectors. A five-cycle is shown in Figure 1.4.

Definition 10. *A bipartite graph is a graph whose nodes may be separated into two disjoint sets U and V such that there exists an edge between all vertices in U and all vertices in V.*

The C_3 and C_4 motif counts are themselves a measurement of node clustering in the graph. The global clustering coefficient defined in Chapter 1 explicitly requires the C_3 count in its calculation. We count a single bipartite motif: the star S_3. An example of S_4 can be found in Figure 1.5.

Definition 11. *A star S_n denotes the complete bipartite graph $K_{1,n}$. In other words, a tree with one internal node, but n branches.*

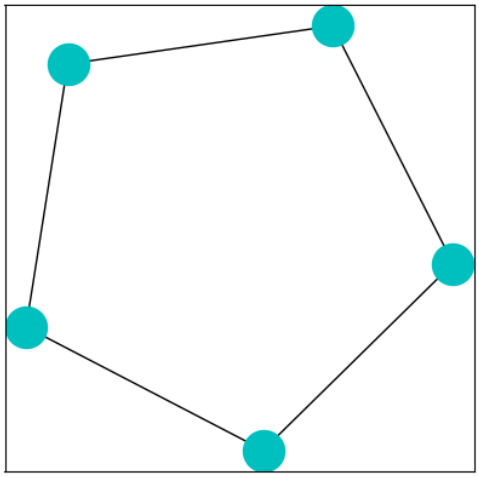

Figure 1.4. A simple cycle of length five, denoted C_5.

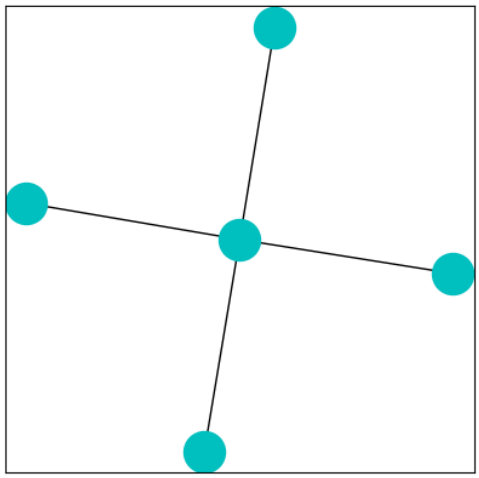

Figure 1.5. The star S_4.

1.2 Non-Simple Cycle Motifs

We wish to consider motifs that we refer to as H_3 up to H_{13}. These motifs, and their enumerations, are described in Alon et al [2]. In Figure 1.6 are the graphs of all eleven non-simple cycle motifs.

Definition 12. *Let f be a homomorphism between graphs G and H. We say that H is a homomorphic image of G provided f is surjective.*

Definition 13. *A graph $H = (V(H), E(H))$ is said to be k-cyclic, for $k > 3$, if it is a homomorphic image of the cycle C_k. The number of different homomorphisms from C_k to H is denoted by $C_k(H)$. H is k-cyclic if and only if $C_k(H) > 0$.*

These motifs are non-simple cycles. For the motifs in Figure 1.6, we can classify them to their homomorphic images. H_3, H_4, H_6, H_9, and H_{11} are all six-cyclic. H_5 is the only five-cyclic graph. However H_5, H_6, H_7, H_8, H_{10}, H_{12}, and H_{13} are all seven-cyclic.

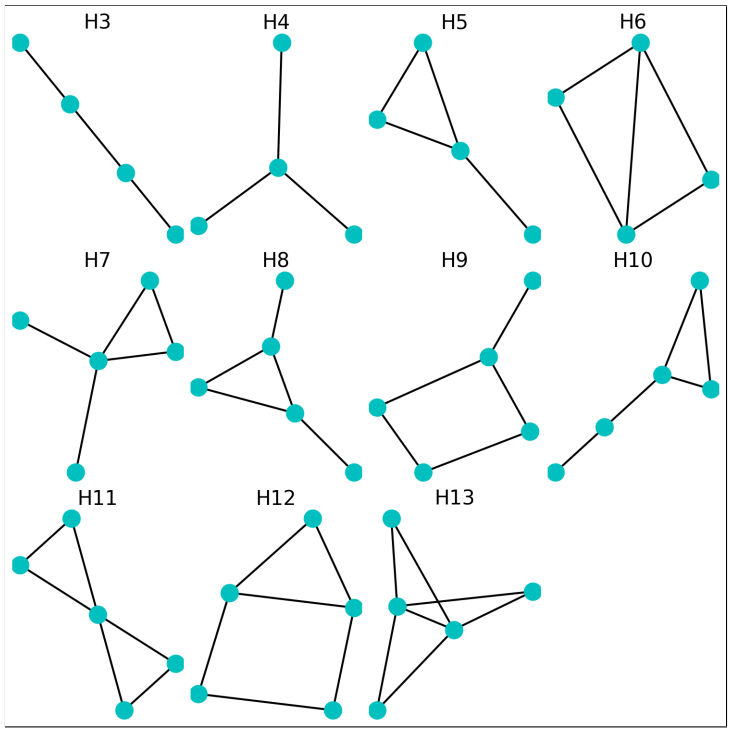

Figure 1.6. The eleven non-simple motifs. This set is comprised of structurally different graphs. For instance, there is a three-walk (H_3), a star S_3 (H_4), the star S_3 with two vertices attached (H_5), and an S_3 with two edges added. The motif counts differ wildly depending on a graph's attachment mechanism(s).

We will also consider the simple cycles of C_3, C_4, C_5 as motifs. Motif counts can be generated from a network's adjacency matrix in the following ways. Let A be the adjacency matrix of some arbitrary network with greater than four nodes. Let $N_G(M)$ denote the total count of motif M in the network described by graph G.

When counting motifs, we let E denote the set of edges in the network, $e_{i,j}$ denotes a particular edge between the ith and jth nodes. When counting motifs we treat edges as being undirected. d_i denotes the degree of the ith node. We define the degree d_i of vertex v_i as the number of edges connected to v_i. A is the adjacency matrix of the graph G. Finally, $a_{i,j}^{(k)}$ denotes the kth power of the matrix element at the ith row and jth column of matrix A. The formulae for motif counts are as follows:

$$
\begin{aligned}
N_G(C_3) &= \frac{1}{6}\operatorname{tr}(A^3) \\
N_G(C_4) &= \frac{1}{8}\left(\operatorname{tr}(A^4) - 4\sum_{i=1}^{n}\binom{d_i}{2} - 2\sum_{i,j\in E} e_{i,j}\right) \\
N_G(C_5) &= \frac{1}{2}\sum_{(i,j)\in E}(d_i-1)(d_j-1) - 3N_{C_3} \\
N_G(H_4) &= \sum_{i=1}^{n}\binom{d_i}{3} \\
N_G(H_5) &= \frac{1}{2}\sum_{i=1}^{n} a_{i,i}^{(3)}(d_i-2) \\
N_G(H_6) &= \sum_{(i,j)\in E}\binom{a_{i,j}^2}{2} \\
N_G(H_7) &= \frac{1}{2}\sum_{i=1}^{n} a_{i,i}^{(3)}\binom{d_i-2}{2} \\
N_G(H_8) &= \sum_{(i,j)\in E} a_{i,j}^{(2)}(d_i-2)(d_j-2) - 2N_G(H_6) \\
N_G(H_9) &= \sum_{i-1}^{n}(d_i-2)\sum_{j\neq i}\binom{a_{i,j}^{(2)}}{2}
\end{aligned}
$$

$$N_G(H_{10}) = \frac{1}{2}\sum_{i=1}^{n} a_{i,i}^{(3)} \sum_{j\neq i} a_{i,j}^{(2)} - 3N_G(C_3) - 2N_G(H_5) - 4N_G(H_6)$$

$$N_G(H_{11}) = \sum_{i=1}^{n} \binom{\frac{1}{2}a_{i,i}^{(3)}}{2} - 2N_G(H_6)$$

$$N_G(H_{12}) = \sum_{(i,j)\in E} a_{i,j}^{(2)} a_{i,j}^{(3)} - 9N_G(C_3) - 2N_G(H_5) - 4N_G(H_6)$$

$$N_G(H_{13}) = \sum_{(i,j)\in E} \binom{a_{i,j}^{(2)}}{3}$$

We want to understand how each motif count affects another given the addition of new nodes or edges in any simulation. Some motif graphs contain an appearance of another motif and this will affect how they interact with one another. In Chapter 5, an analysis of edge addition on the motif graphs shows simple changes to a motif's graph may cause a combinatorial effect generating a large number of new motif appearances.

CHAPTER 2

Network Theory

The social network has taken on new meaning with the advent of the computer, particularly with mobile technology. Social media allows for information (and misinformation) to spread rapidly within and across communities. These communities and their interactions are well-modeled by network theory, an extension of graph theory. Individuals are represented as nodes in these networks, and their connections as edges. These nodes and their connections form patterns in a network. These patterns, or motifs, offer a way to characterize the local structure of the network making the motifs informational features. Motif counts change over time as users enter and leave the network. The dynamic behavior of these motif counts, and how they correlate with one another, offers a way to understand how the network changes locally in time. This analysis can extend easily beyond social networks into other domains. In the following section, we first establish some of the fundamentals of network theory.

2.1 Complex Networks

The terms "network" and "graph" can be used interchangeably as networks are represented via vertices and edges, but networks are contextual. In its most general sense, a network is comprised of objects and connections between them. These connections may be directed, undirected, weighted - representing any number of different relationships. The versatility and effectiveness of this approach has encouraged network modeling in a variety of fields: physics, sociology, biology, economy, chemistry [16]. Some networks are small and simple. Most networks are large, containing many nodes and connections, many of which are topologically (or structurally) non-trivial. These we refer to as complex networks, and they are commonly found in those fields mentioned previously.

Complex networks differ from other networks as edges found between vertices form in patterns neither completely random nor regular. Such networks often have degree distributions that are fat-tailed, meaning a few nodes are of relatively high degree, while most nodes are not. These networks are commonly called scale-free networks. The Barabási–Albert model we examine later falls under this category. These networks also cluster, which correlates with the scale-free property.

2.2 Statistical Properties of Networks

Network science has many statistical measures to differentiate networks from one another. These measures offer different levels of insight into a network and its structure. One such measure is the notion of centrality. There are several different types of centrality, but each represents a way to denote the most important vertices within a given network. One such measure is degree centrality. Given a graph G, a vertex v_i's degree d_i is the number of edges connected to v_i. Aptly named, degree centrality assigns a weight to each vertex determined by its degree d_i. In Chapter 3 and Chapter 4, degree centrality is useful as the dynamics of the Barabási–Albert model directly depend on it. The centrality measures are informative regarding the connectivity of the network, but leave much to be desired in the way of understanding structure.

We make use of the clustering coefficient in Chapters 3 and 4, to characterize graph dynamics. Thij [22] notes that clustering coefficients are a way to understand how a network's density changes over time. He further notes that future study of the proposed Twitter model in Chapter 4 should include an analysis of its temporal clustering coefficients. To define the clustering coefficient, we first define the neighborhood N_i of a vertex v_i,

$$N_i = \{v_j : e_{ij} \in E \vee e_{ji} \in E\}$$

Where E is the set of edges in the graph. A node v_i's neighborhood is the set of nodes, which have an edge between them and v_i. The local clustering coefficient of a node v_i in the undirected graph G is defined as

$$C_i = \frac{2|p_i|}{k_i(k_i - 1)}$$

where we define p_i

$$p_i = \{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}$$

This can also be calculated by way of the adjacency matrix described in Definition 2.

$$C_i = \frac{1}{k_i(k_i - 1)} \sum_{j,k} a_{ij} a_{jk} a_{ik}$$

The average clustering coefficient is calculated by arithmetic mean.

$$\bar{C} = \frac{1}{|V|} \sum_i C_i$$

The global clustering coefficient is simply

$$C_G = \frac{\text{Total Number of Triangles}}{\text{Total Possible Triangles}}$$

Clustering allows us to understand how dense a network is relative to a complete graph.

2.3 The Social Network Twitter

Twitter is now, as of 2021, an eminent example of complex networks in social media. On Twitter, users can follow one another. A user populates their follower's feeds with their tweets (message posts) and retweets (sharing message posts). As of 2019, there are 330 million monthly active users on Twitter, thirteen years after the platform was established in 2006. However, according to Pew Research, the top ten percent of Twitter users tweet 138 tweets per month, while the bottom ninety percent of Twitter users only tweet twice per month. The top ten percent of users have a median of 387 followers, while the bottom ninety percent have only a median of 19 followers. The top ten percent also follow more users, a median of 456 accounts, compared to a median of 74 accounts for the bottom ninety percent [24]. This itself is suggestive of Twitter being a scale-free network, but analysis confirms this [3]. Twitter, as a network, exhibits a fat-tailed degree distribution. Moreover, its estimated average clustering coefficient is 256,000 times larger than expected for a random graph.

Twitter's status as a complex network aside, the platform is also notable for its position among social networks not only for its size, but also its influence on culture and politics. During the 2016 election, a survey of thirty million tweets from two million users linked articles that were found to be spreading false information. Moreover, this information spread based on community structure within an inclusive left-right influencer network. Twitter is cliquey and the communities form around shared interests. In online communities where exposure to the same memetic theme occurs frequently, both facts and rumors may spread easily and quickly [5].

Twitter also has a substantial economic impact. Twitter sentiment is known to precede fluctuations in the stock market [7] and Bitcoin prices [21]. However, only a handful of Twitter users actually have influence (although groups of people could theoretically be enough to influence market prices [20]). Elon Musk is one such person who, as of March 2021, has influenced multiple markets driving Tesla's share price up and down [11], as well as causing jumps in cryptocurrency prices [8] [9]. Musk, is as of 2021, the forty-third largest Twitter account, giving him much more influence than the vast majority of users.

Twitter's suitability as a subject of network science is clear. The users and followers (and retweets) are naturally described by vertices and edges. One can generate networks from Twitter in two ways. First, there are the user accounts linked

to one another through followers and follows, in-degrees and out-degrees. An example of this is given in Figure 2.1. One can also construct networks of tweets and retweets as discussed in Chapter 4. A user posts a message, which is retweeted by a portion of their followers, which is then again retweeted by a portion of their followers. It is this latter case we will consider in the Thij model.

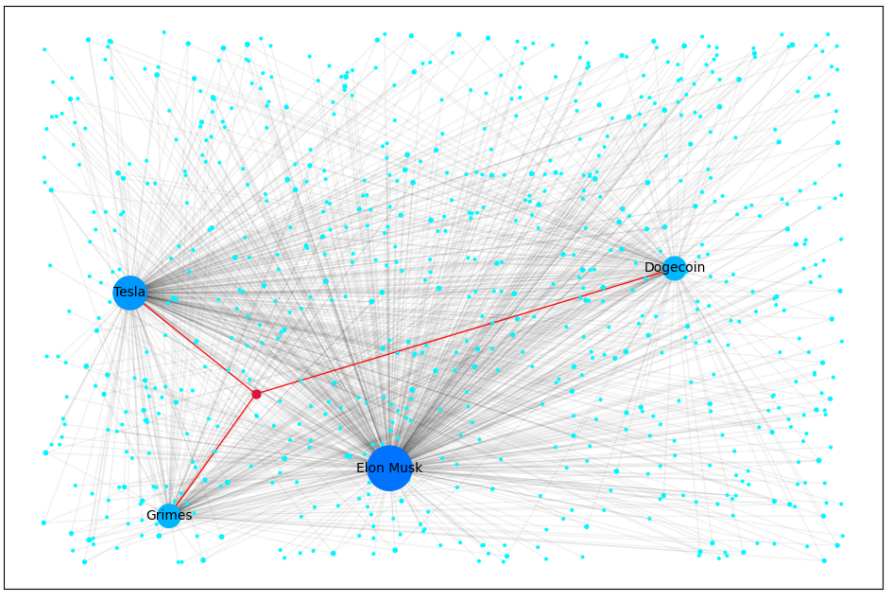

Figure 2.1. All the circles represent nodes or users in this Twitter network. The edges between the nodes denote follows. The red node is a single user in the Twitter network following Tesla, Dogecoin, and Grimes (denoted by the red edges), but no others. The red user receives all tweets and retweets from Tesla, Dogecoin, and Grimes. This user may retweet these messages outside the network, potentially bringing in more users to the network. Note, the red node and its adjacents are isomorphic to S_3 (or the H_4 motif).

CHAPTER 3

Barabási–Albert Model

The early models of networks failed to capture the characteristics that appear in empirical data. An early model proposed by Paul Erdős and Alfréd Rényi, appropriately called the Erdős–Rényi model, generates graphs of fixed node count where each pair of nodes has the same probability to have an edge between them [12]. Réka Albert and Albert Barabási proposed the Erdős–Rényi model lacked key pheonmena of most real-world networks [4]. First, real networks often have degree distributions that are not explainable by the degree distribution of the Erdoes-Renyi model. Second, real networks have a sizable largest connected component - a large cluster of nodes inside the network, forming a hub of activity. Finally, the local clustering coefficient of most networks decreases as the node degree decreases, but is independent of overall graph size. Réka Albert and Albert Barabási developed a model of complex networks encompassing those we find common in practice. Edges are made to follow a preferential attachment mechanism, which ascribes probabilities of attachment by the relative degree of the nodes in a network. It is a point of contentious debate if networks are scale-free and their degree distributions follow a power-law. A network is said to follow a power-law if the fraction of nodes P having k edges is

$$P(k) \approx k^{-\gamma}$$

The preferential attachment mechanism of the Barabási–Albert model generates this power-law. The Barabási–Albert model typically exhibits $2 < \gamma < 3$.

To model such a network, we begin at $t = 0$ by initializing m nodes and randomly distribute a number of edges between them according to a uniform distribution. Then at each time-step $t > 0$, we introduce a new node and k edges between that node and the existing nodes. We assign probabilities $p(n)$ of attachment to the nth node via the probability distribution:

$$p(n) = \frac{d_n}{\sum_{i=1}^{N} d_i}$$

where d_n is the degree of the nth node and N is the total number of nodes at $t - 1$. Nodes which have a high degree are probabilistically more likely to be attached to new nodes.

In Figure 3.1, we see that only certain motifs can be generated in the $k = 1$ BA model. Motif counts in the Barabási–Albert model are dependent upon the initialization of the m nodes and the choice of k. For example, if $k = 1$, the model cannot complete any new cycles. If the initial graph at $t = 0$ does not contain any C_3 or C_4 appearances, then any motif which has an induced subgraph isomorphic to C_3 or C_4 cannot appear. The BA model for $k = 1$ can only attach nodes to the existing C_3 or C_4 appearances and in that way generate new H_7, H_8, or H_9 appearances. We can observe the development of the graph over time in Figure 3.2. In Figure 3.3, we can see the a visualization of the network at the final time-step.

The BA model with $k = 2$ is capable of generating those new C_3, C_4, and C_5 appearances in the network quite easily as seen in Figure 3.4. In Figure 3.5, we see the development of the network for $k = 2$. In Figure 3.6, we see a visualization of the nodes and edges of the network in the final time-step.

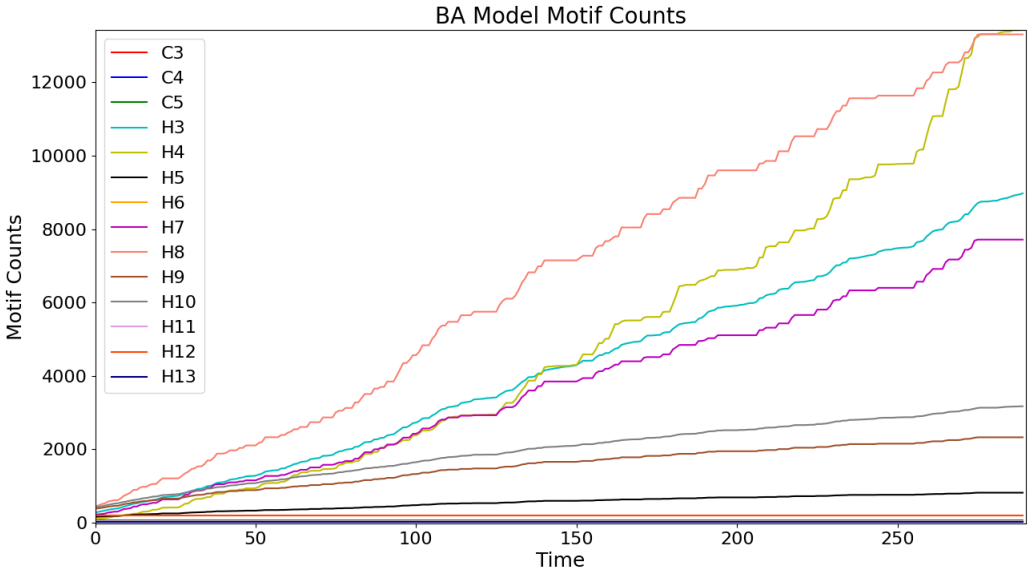

Figure 3.1. This Barabási–Albert model is initialized with $m = 8$ nodes. At each time-step a new node enters and is attached to $k = 1$ nodes by the preferential attachment mechanism. All simulations are terminated after the network reaches a size of 300 nodes.

The Barabási–Albert model is a good candidate as a baseline to our more complex model. The Barabási–Albert model represents a simpler, non-trivial model, which is widely acknowledged as a useful tool for understanding networks across a variety of disciplines. The preferential attachment mechanism only allows for the

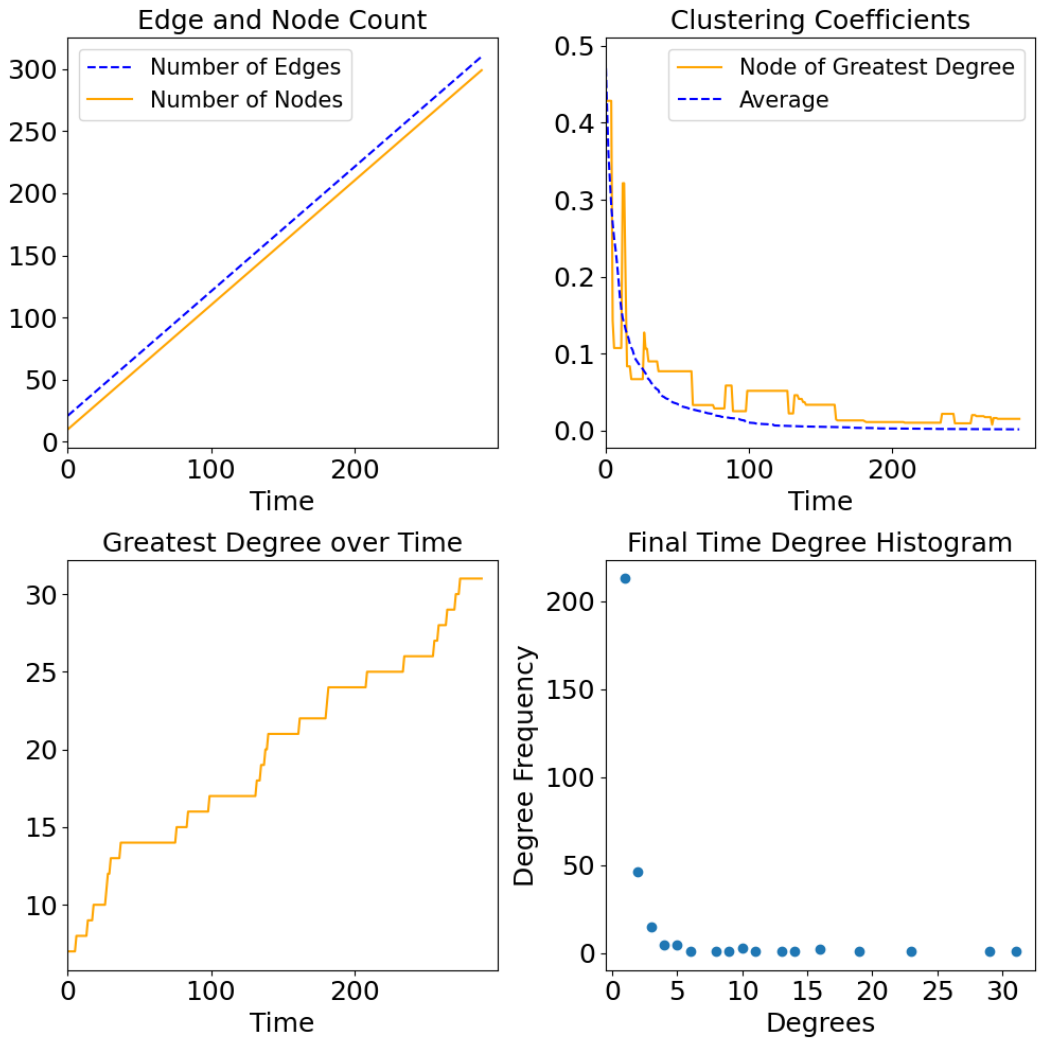

Figure 3.2. Statistics characterizing the development of the Barabási–Albert model for $k = 1$. Edges and nodes grow linearly. We also see as Barabási noted, the edge density and clustering coefficients decrease asymptotically.

addition of nodes and a set number of edges at teach timestep. There are limitations to the Barabási–Albert model. The nodes and edges are restricted to linear growth, which may fail to capture certain phenomena that appear empirically. The model is also incapable of removing nodes, a point on which Barabási himself has elaborated. Our next model does account for the first of these limitations at the cost of increased complexity.

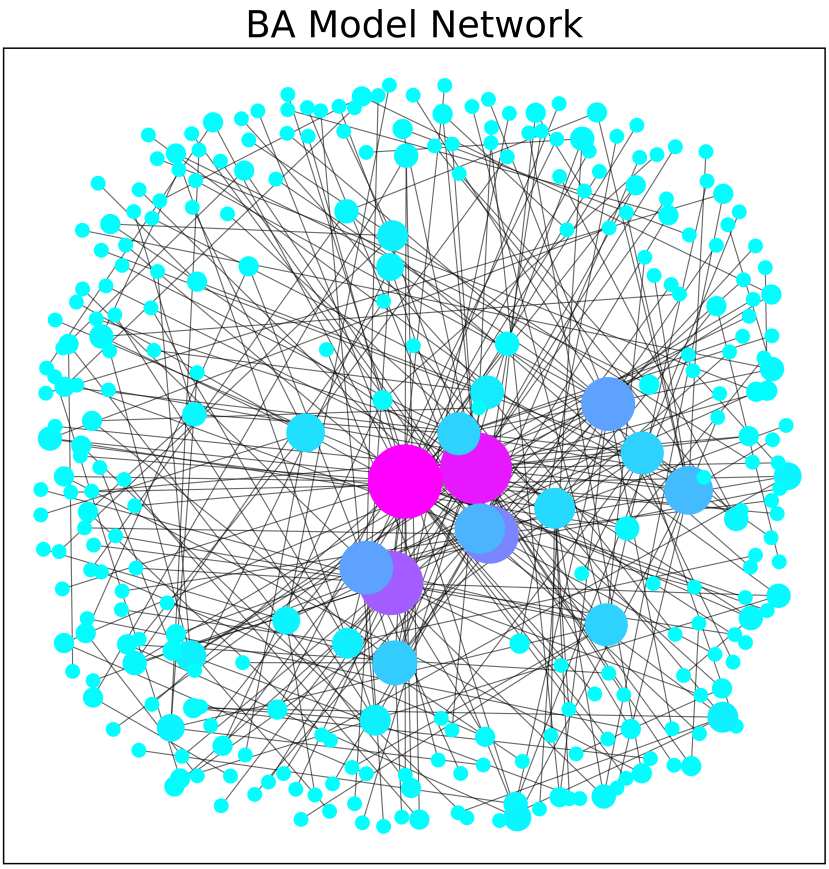

Figure 3.3. The Barabási–Albert model for $k = 1$ at the final time-step. The color and size of the nodes reflect the degree centrality of each node.

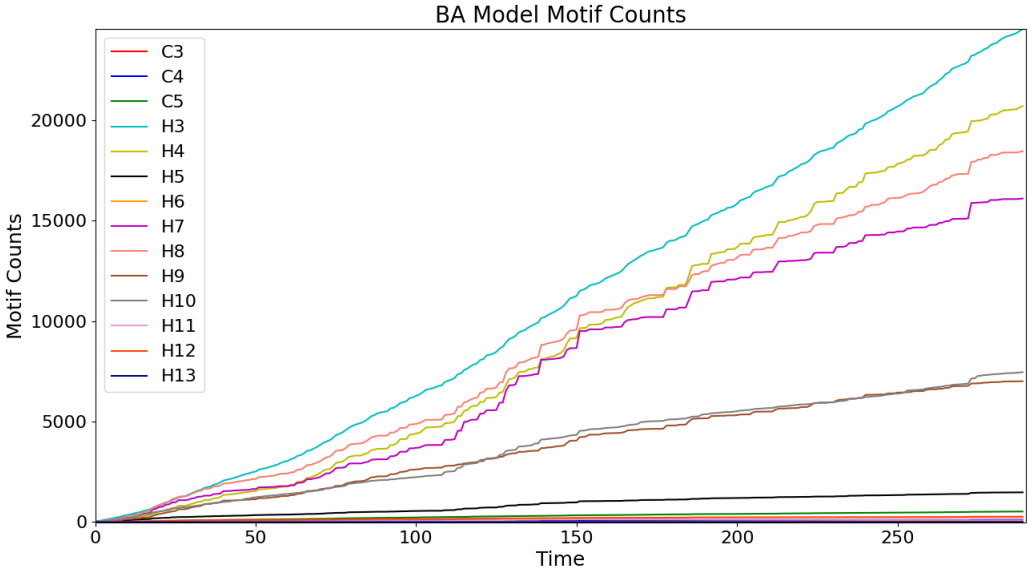

Figure 3.4. This Barabási–Albert model is initialized with $m = 5$ nodes. At each time-step a new node enters and is attached to $k = 2$ nodes. We see that different motif appearances correlate for the $k = 2$ simulation and other motif appearances, like those of C_3 and C_5, can be generated.

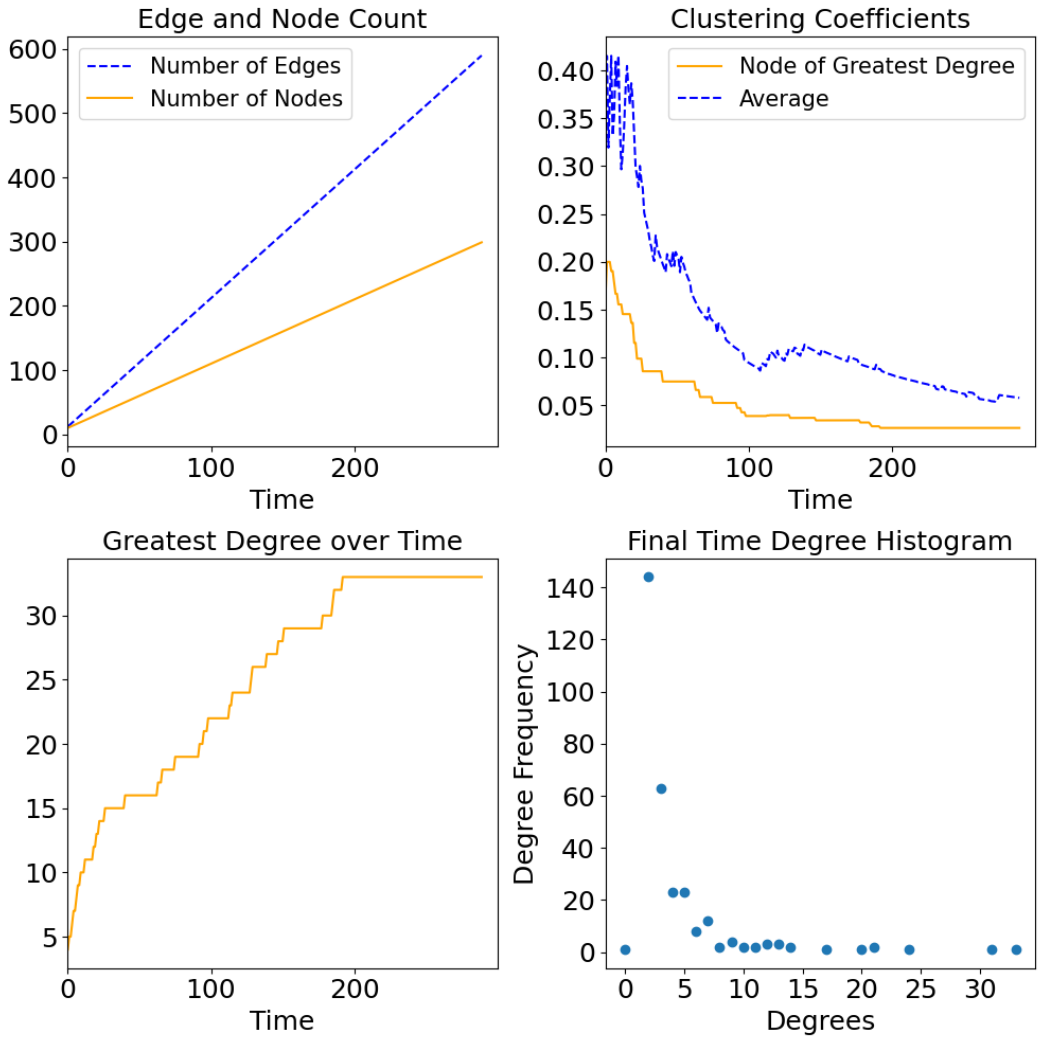

Figure 3.5. Statistics characterizing the development of the Barabási–Albert model for $k = 2$. Edges and nodes grow linearly, with 2 edges added at each time-step. The clustering coefficients asymptotically approach zero. The histogram at the final time shows the vast majority of nodes, approximately 220, have two or three attachments while three vertices have a degree greater than 35.

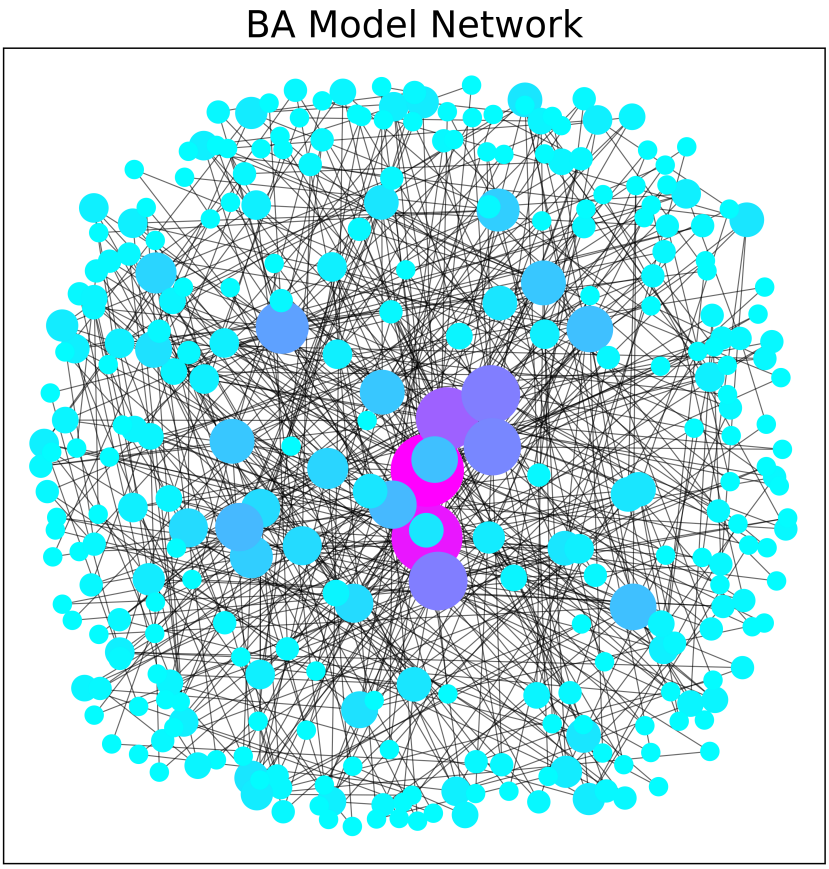

Figure 3.6. The Barabási–Albert model for $k = 2$ at the final time-step. Once again the color and size reflect the degree-centrality of each node.

CHAPTER 4

The Thij Model

The Thij model is a particular random graph that seeks to model the development of a Twitter network [22]. Twitter offers a platform where a user may post a message to all of their followers' feeds. Those followers may then ignore, like, reply (comment), or retweet (quote tweet) that message. In the instance they choose to retweet the message, that message is posted to their respective feed, and their own followers now have the opportunity to retweet that same message. If a message has received a significant number of retweets the message is more likely to be seen and thus retweeted again. A preferential attachment mechanism drives the popularity of the most popular tweets [3]. The Thij model incorporates a superstar mechanism that was first proposed in [6]. The superstar mechanism is ascribed to original tweets, which have much higher chance of being retweeted within a network.

A user can retweet different original messages at different times, meaning edges can be generated between existing nodes in the network. Accounting for this, one can produce a model better suited to Twitter simulations than the Barabási–Albert model described in Chapter 3.

We wish to simulate a network of retweets. There are original message nodes from users u_i and retweets from users v_i. All users are capable of retweeting or posting original messages. The simulation starts with an initial message node from user u_0 at time $t = 0$. For all time-steps, there is now the possibility of three events: $T1$, $T2$, and $T3$.

***T*1**: A new message node from user u_k appears.

***T*2**: A user v_k enters the retweet network and retweets an existing user's message. The retweeted message is either an original message node u_i or another user v_i's retweet. This user v_k retweets an original message node u_i with probability q and any other node with probability $\frac{1-q}{N}$. N is the total number of nodes in the network.

***T*3**: An existing user v_i retweets another existing user u_i or v_j. The retweeter retweets a message node with probability q and all other message or retweet nodes with probability $\frac{1-q}{N}$.

For clarity, we make a distinction between the posts of original messages and retweets. v_i may be a simple retweet, or a quote tweet, meaning v_i may reasonably retweet v_k who may have added commentary to the original message. We must also note that there will be multiple message nodes in the retweet graph, and thus we have to decide for any given event which particular message node should be assigned the q probability. Here we introduce a preferential attachment mechanism. At any time t, given a $T2$ or $T3$ event occurring, the probability of a particular message node being chosen is almost the same mechanism described in the Barabási–Albert model. Instead, a message node is chosen by its total descendants and not by the degree of the message node itself. In essence, a message node is selected based upon the number of those who have retweeted the message and all those who have retweeted retweets of that message. A visualization of each event is seen in Figure 4.1

We now must assign probabilities of $T1$, $T2$, and $T3$ events. Let $P(T_i)$ denote probability of event T_i. Let λ, and p be parameters such that $\lambda \geq 0$, and $1 \geq p \geq 0$.

$$P(T1) = \frac{\lambda}{1+\lambda}$$
$$P(T2) = \frac{p}{1+\lambda}$$
$$P(T3) = \frac{1-p}{1+\lambda}$$

Our choices of λ and p will drastically affect the dynamics of the graph, as well as the motif counts that make up its structure. We want to consider a series of cases for different probabilities allowing us to make informed predictions about the graph's development over time. The motif counts for various parameters help demonstrate the differences in scale and structure which arise from different probability distributions. Consider the case when $0.5 > \lambda > 0$ and $0.5 \gg p > 0$. In this case, it is the $T3$ event expected to be the most prevalent. For these parameter values, we see in Figure 4.2 the first four greatest motif counts almost everywhere in order are H_7, H_8, H_9, and H_{10}. The $T3$ mechanism allows for more C_3's and C_4's to form within the network. The development of the network over time and it's final state are seen in Figures 4.3 and 4.4.

Next, consider $1 > \lambda > 0.5$ and $0.5 \gg p > 0$. Here the $T1$ event should dominate the dynamics with many new message nodes introduced to the overall retweet network. In this scenario, we have a greater likelihood of $T3$ events over $T2$ events. This means those nodes without connections are likely to become connected to other nodes. In

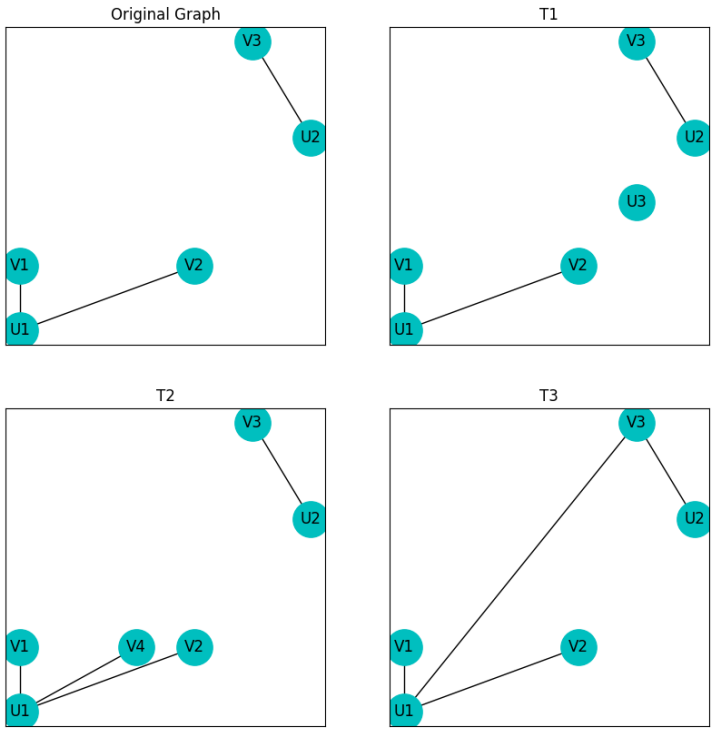

Figure 4.1. We see the $T1$, $T2$, and $T3$ events on a simple graph. In the $T1$ case we see a new message node $U3$ appears. It has yet to be connected to anything at all. In $T2$ we see a new node $V4$. This is a retweet of the message $U1$. Finally in the last plot we see a $T3$ event. Here the user $V3$, who has already retweeted $U2$, now retweets message $U1$.

Figures 4.5 and 4.7, we can visualize the motif development and the overall network. Looking at the final time degree distribution in Figure 4.6, we see the majority of nodes are still unconnected.

What about the case $0.5 \gg \lambda > 0$ and $1 > p \gg 0.5$? The probability distribution is such that adding a new node with an edge is the most likely outcome. These parameter choices combined with the superstar parameter $q = 0.9$ implies that $T2$ and $T3$ attachments are very likely to target the root message node. In Figure 4.8, we see the resulting star pattern in the motif counts and the end-result in Figure 4.10.The preferential attachment mechanism encourages an induced subgraph S_k to form with $k \gg 1$. For any $k \gg 3$, S_k contains many appearances of H_4. For every increase in k, another increase becomes more likely. The networks development over time can be seein in Figure 4.9.

Last, $1 > \lambda > 0.5$ and $1 > p \gg 0.5$, we see an increased chance of adding many new root messages, but still a good possibility of introducing a new node with a new edge, but a decreased possibility of adding in only new edges. For an example of the resulting motif counts we can look to Figure 4.11. The network at the final time=step is shown in Figure 4.13. Changes in node and edge count throughout the simulation are seen in Figure 4.12.

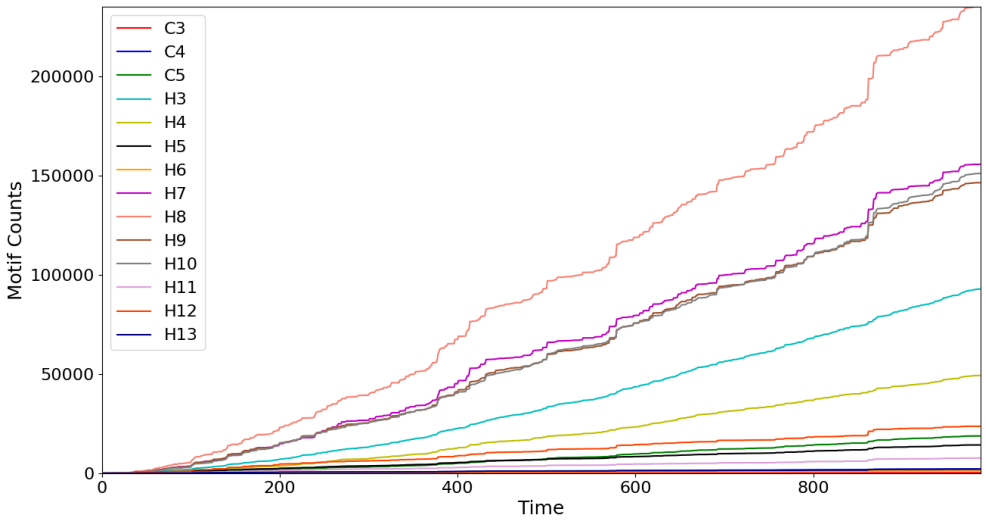

Figure 4.2. Here, for $\lambda = 0.2, p = 0.2$, we see that H_8's lead with H_7, H_9, and H_{10} counts. These motifs are closely correlated with one another throughout the time series.

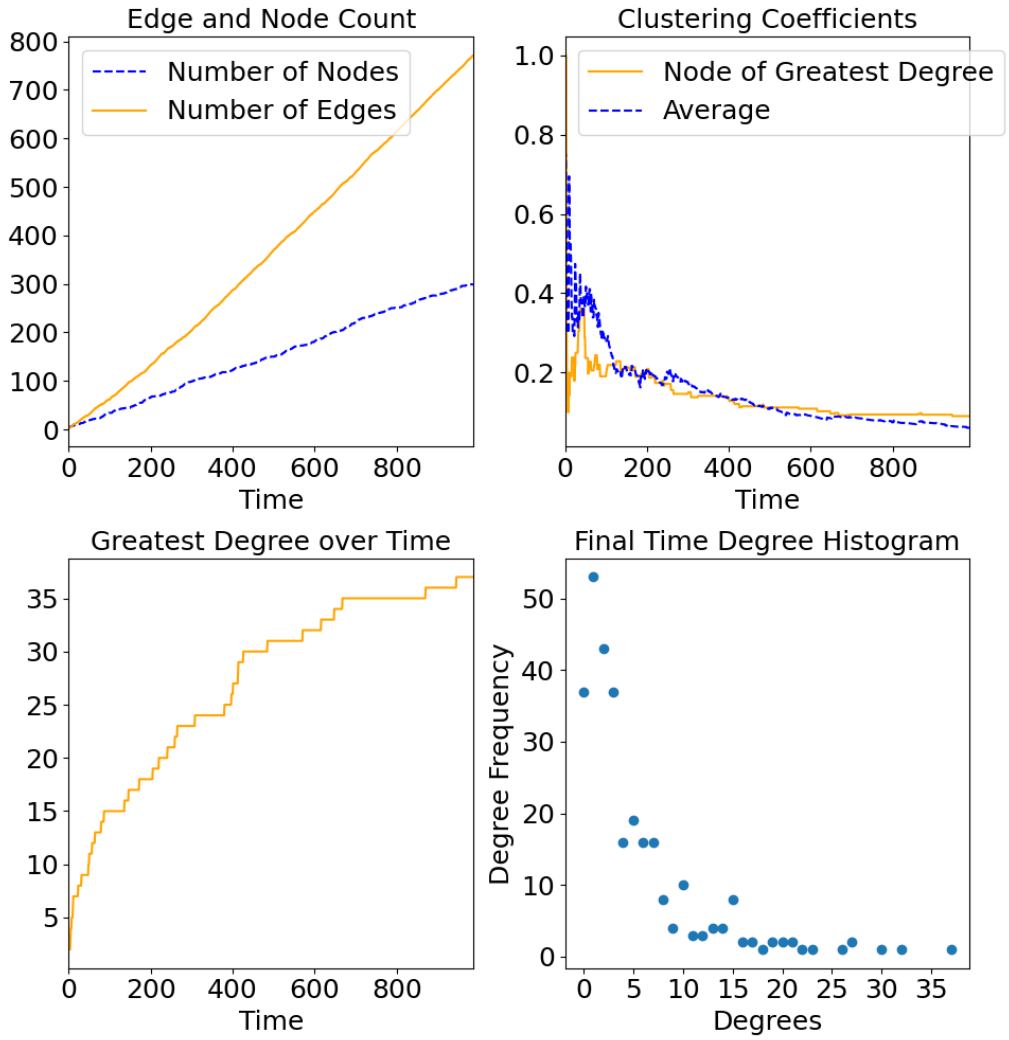

Figure 4.3. Compared to the Barabási–Albert model, for the Thij simulation neither edge count nor node count must grow strictly linear at each time-step. The edge density tends toward zero, although a $T3$ event amounts to a small increase in edge density. The clustering coefficients tend toward zero asymptotically, but are surprisingly large early in the simulation. Finally, we see a final time degree histogram that is similar.

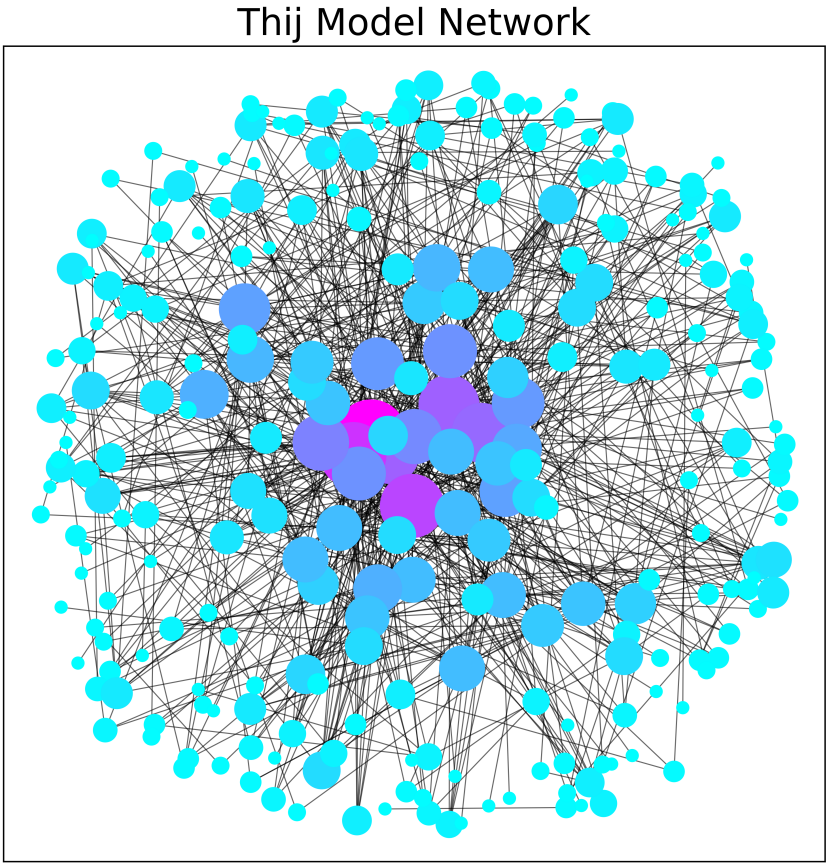

Figure 4.4. The network for $\lambda = 0.2$, $p = 0.2$ at the final time-step.

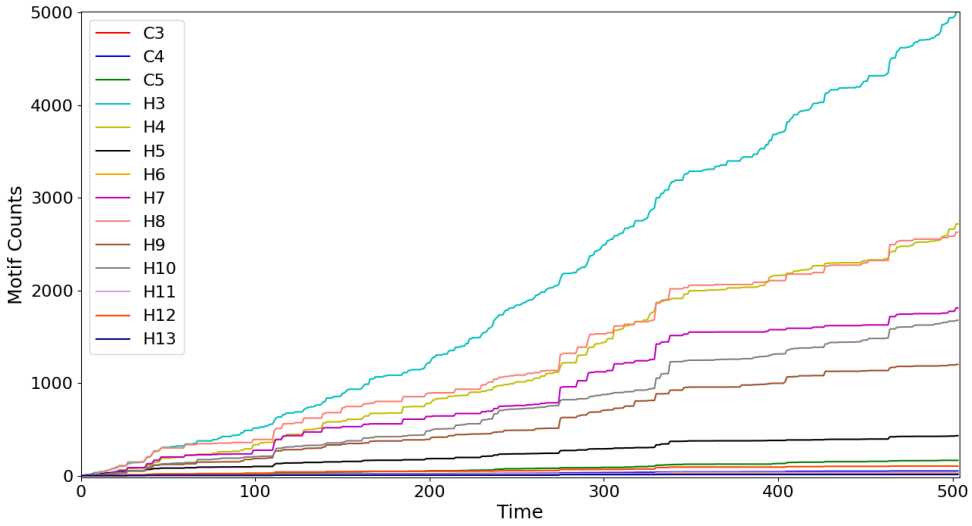

Figure 4.5. For $\lambda = 0.8, p = 0.2$ many new message nodes appear ($T1$ events), but with low p we should see many $T3$ events connecting these nodes. H_3 motifs are the most prevalent followed by H_8's and H_4's.

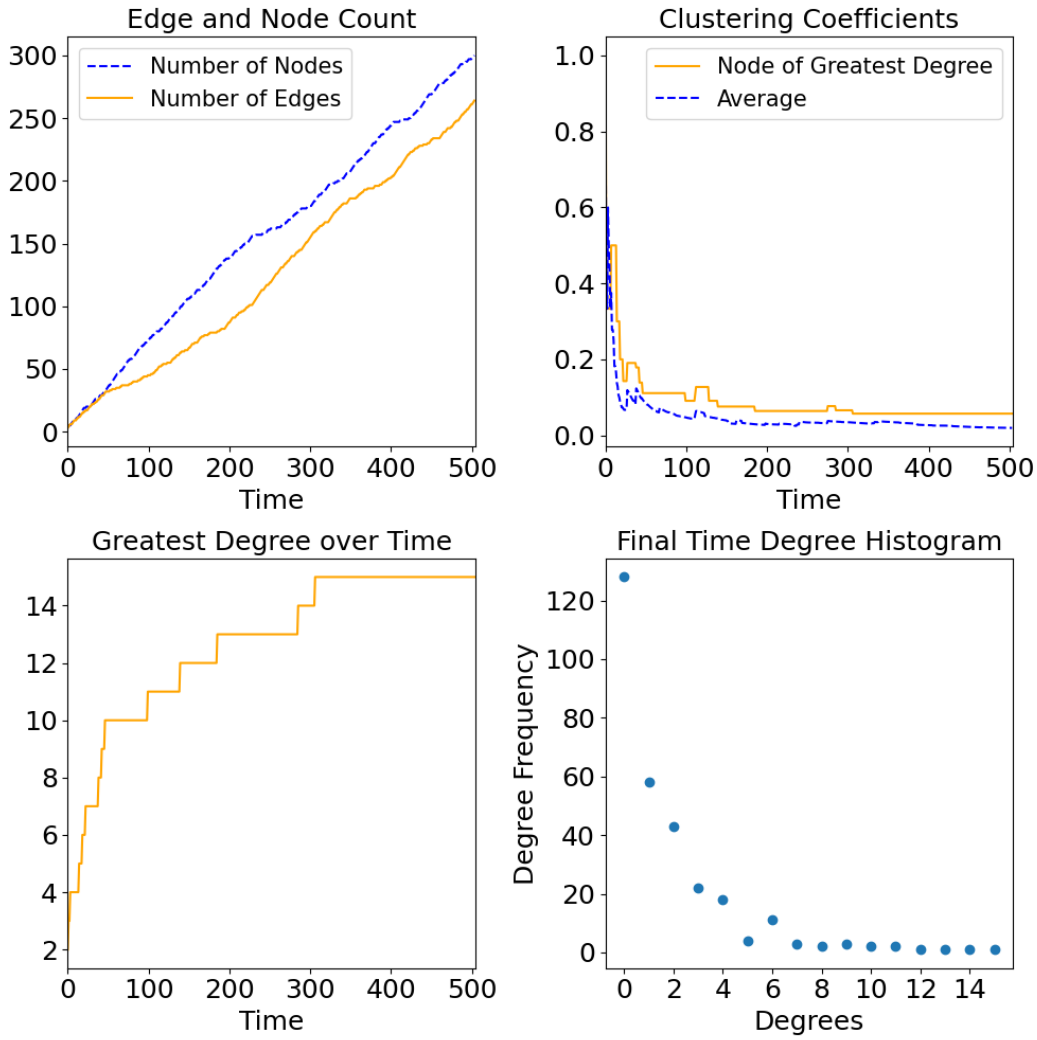

Figure 4.6. For $\lambda = 0.8, p = 0.2$, edges and nodes travel tightly together with roughly a ratio of one-to-one. Here we see many nodes unattached with degree zero. For those that are attached, we do see a power law describing degree distribution, but one that is not quite as strong as those found in other simulations.

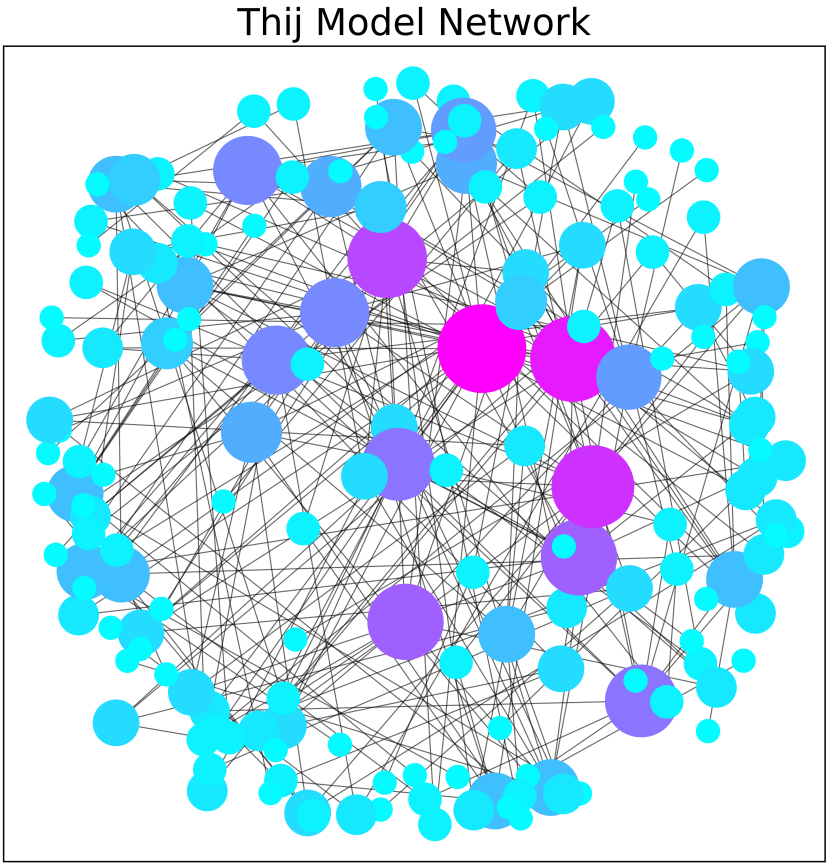

Figure 4.7. The network for $\lambda = 0.8$, $p = 0.2$ at the final time-step.

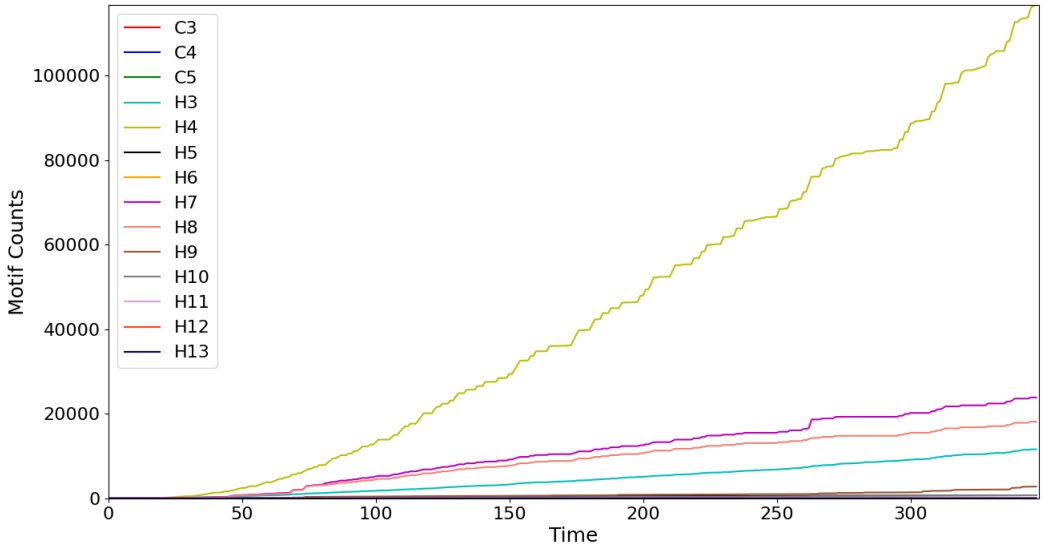

Figure 4.8. For the parameter values $\lambda = 0.2, p = 0.8$, there is a decreased likelihood of new message nodes appearing, but high p means a greater likelihood of $T2$ events which we speculate lead to a large count of H_4's. We see that H_7 and H_8 counts steadily increase. This is discussed further in Chapter 5.

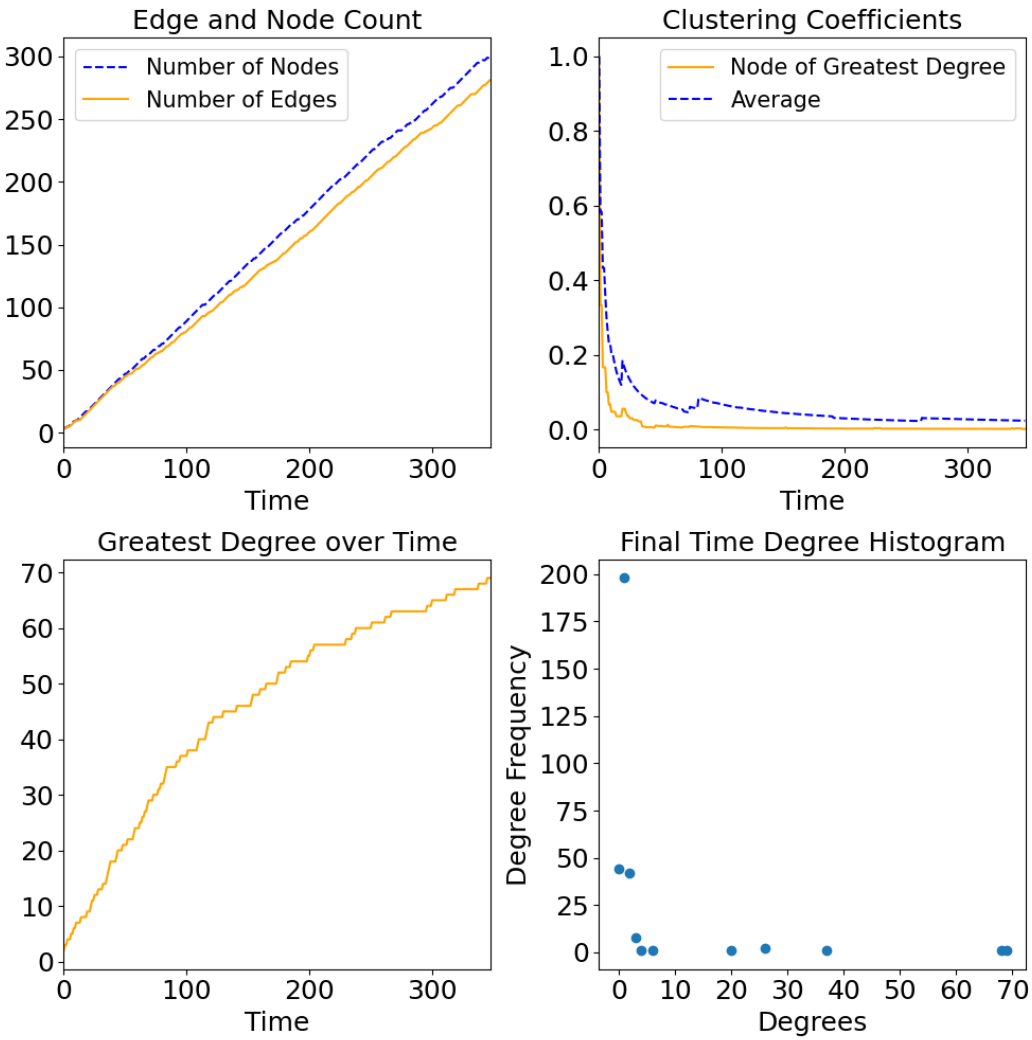

Figure 4.9. For $\lambda = 0.2, p = 0.8$, we see two nodes with degrees greater than seventy but an abundance of nodes with only one or two connections. We can see this reflected in the graph of the network in figure 4.10.

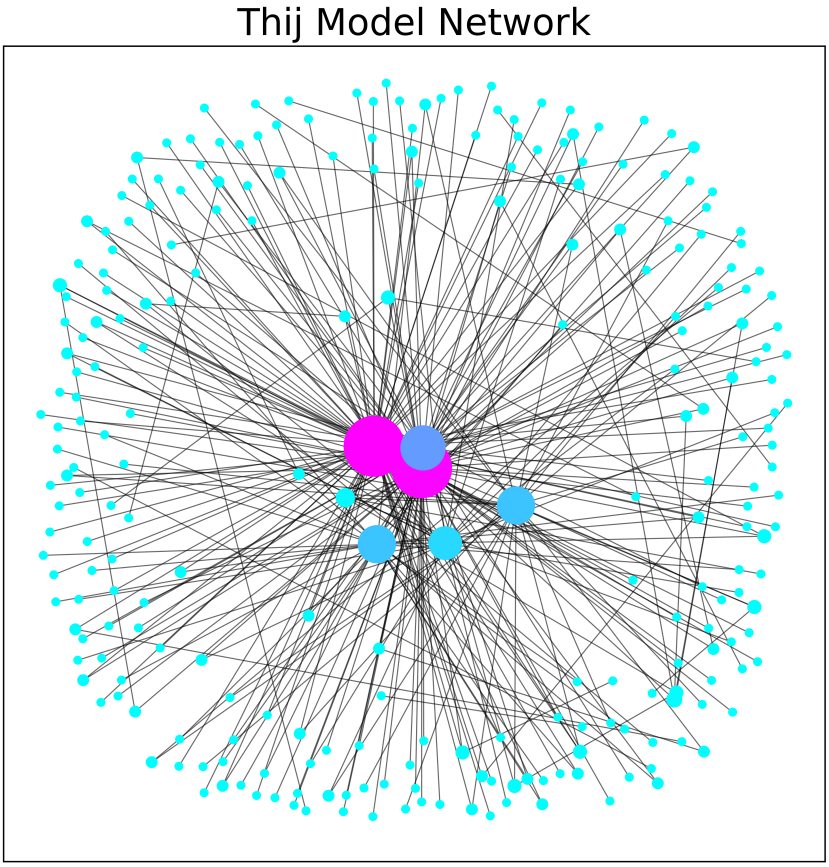

Figure 4.10. The network for $\lambda = 0.2$, $p = 0.8$ at the final time-step.

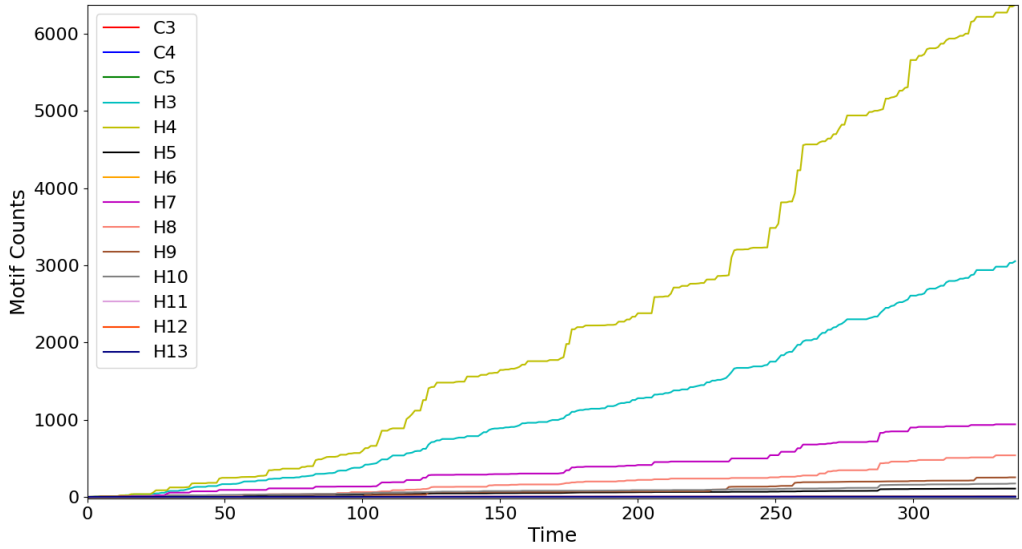

Figure 4.11. **For $\lambda = 0.8, p = 0.8$, like the simulation in figure ??, we see a prominence of H_4's. The scales of the motif counts between simulations are separated by several orders of magnitude. In this simulation, there is a relatively high count of H_3's. We can explain the difference in magnitude due to many $T1$ events introducing many unconnected nodes, but the occurrence of $T2$ events is sufficient to make H_4 the motif of highest count.**

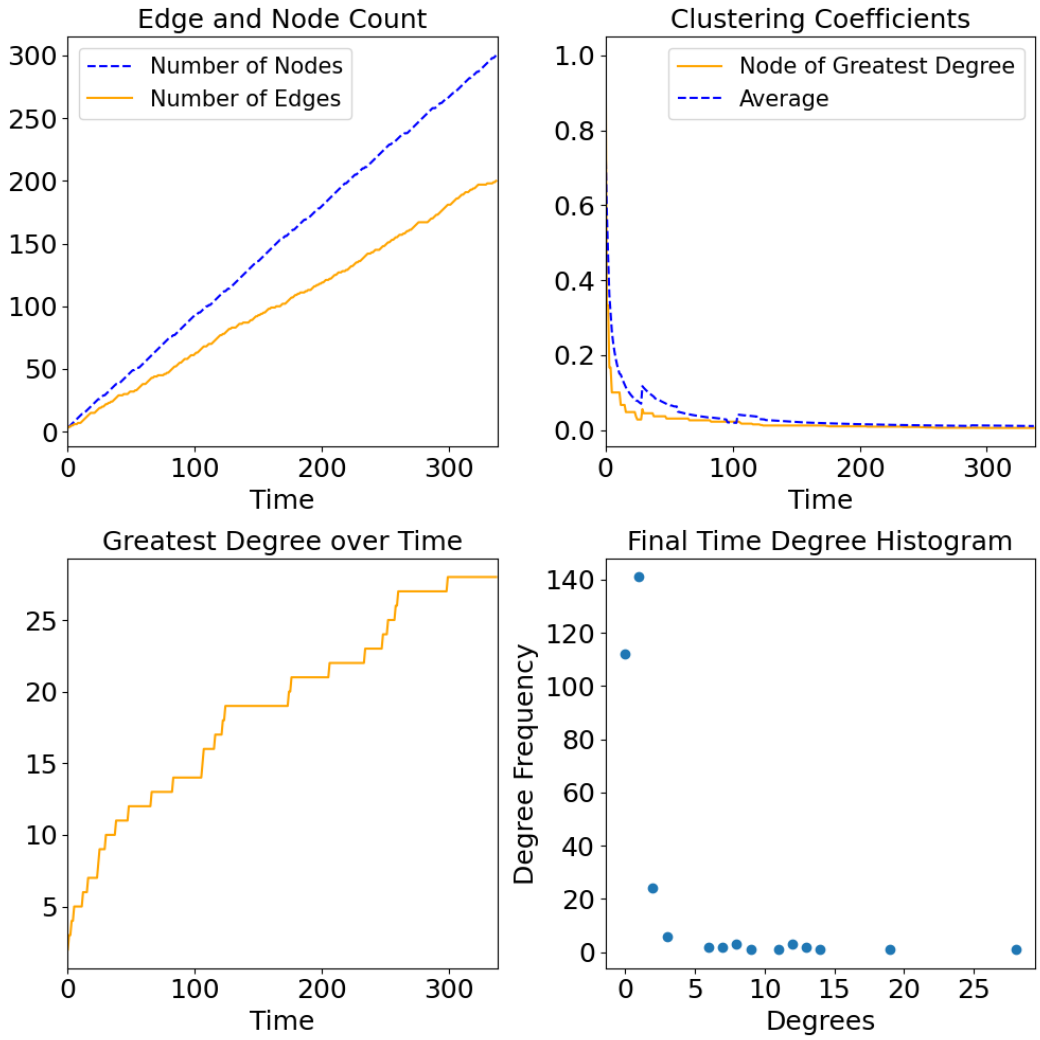

Figure 4.12. **The $\lambda = 0.8, p = 0.8$ Twitter simulation produces many more nodes than edges, because of the frequency of $T1$ events. The vast majority of nodes only have degrees of one or two, while we see a single node with 25 connections. This might suggest many small clusters of nodes with a single larger cluster around a single message node. The clustering coefficient suggest a low number of triangles within the graph.**

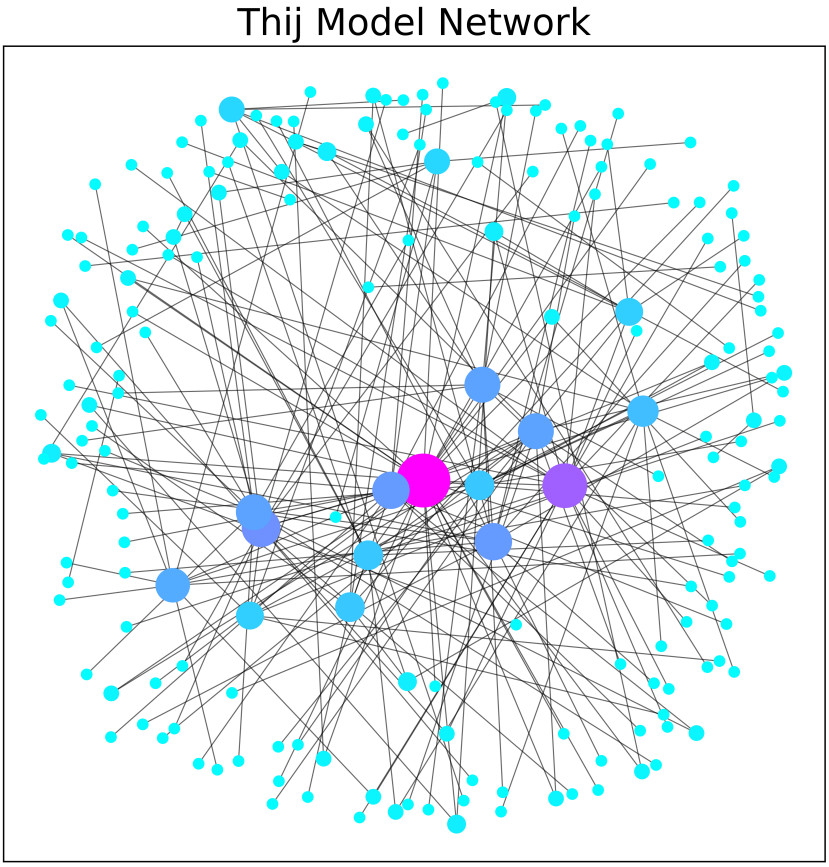

Figure 4.13. The network for $\lambda = 0.8$, $p = 0.8$ at the final time-step.

The dynamics and end state of the network vary across parameter choices. Given greater p values and smaller λ values, a few nodes exhibit relatively large degree distributions. The ratio of edges to vertices for the Twitter model varies much more than the Barabási-Albert model. This is expected given the probabilistic nature of how many edges may be added in a given turn (0 or 1) compared to the given k edges of the Barabási–Albert model. This ratio affects not only the network structure but the time-scale over which the network node count grows.

Each graph spans over *very* different time frames. Each graph was allowed to grow to a maximum of three hundred nodes before ending the simulation. Ending the simulation with a size threshold makes a comparison of the network topology more feasible. For $\lambda = 0.2$ and $p = 0.8$, the model was able to reach three-hundred nodes very quickly, in approximately three-hundred time-steps, whereas for $\lambda = 0.2$, $p = 0.2$ it took nearly a thousand time-steps. This is a consequence of λ controlling the frequency of a new node entering without a new edge and p controlling the frequency of new nodes added with edges or if only new edges are introduced.

CHAPTER 5

Barabási–Albert Model Motif and Thij T2 Event Motif Dynamics

To gain insight into graph dynamics, we analyze how the composition of a motif's graph changes upon the addition of a node to a given motif, and attaching that node to an existing node. The tables below count the appearances of a motif in the respective graphs. For example in Figure 5.2, after adding a node to the root of S_3 or H_4, we count four appearances of H_4 as the motif H_4 is isomorphic to four induced subgraphs in the newly generated motif.

5.1 The H_3 Motif

We begin by considering the H_3 motif. The H_3 motif is simply a four-path. A preferential attachment mechanism on this motif will most likely connect a new node to one of the center nodes with probability 0.67, and to an outer node with a probability of 0.33. In the Thij model, this change is based upon which node has the superstar quality. We can visualize these changes in Figure 5.1

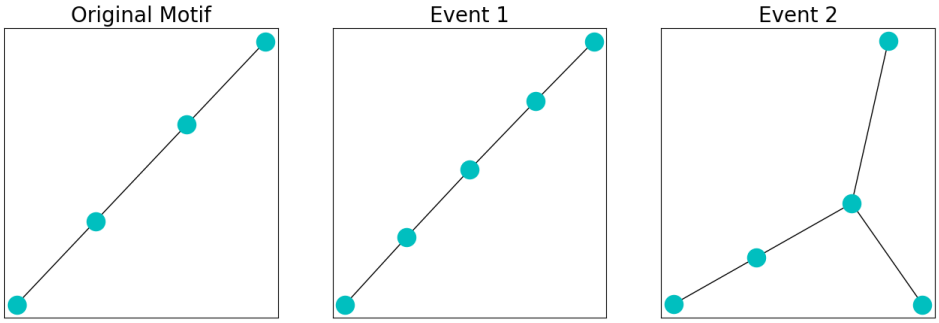

Figure 5.1. The possible graphs generated by adding a node to the H_3 graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2
H_3	1	2	2
H_4	0	0	1

Table 5.1. The rows denote counts of isomorphisms that can be found in the original motif graph or the graphs produced from $T2$ events applied to the original motif. The H_3, a four walk, can be found twice in the first event or second event.

5.2 The H_4 Motif

The H_4 motif is one of the motifs that are of primary interest given that for any star S_k with $k \geq 3$ we will find $\binom{k}{3}$ appearances of the motif. Given an induced subgraph isomorphic to the star S_k, attaching a node to the root node of S_k will generate k new appearances of the H_4 motif. The second event seen in Figure 5.2 illustrates this phenomena. For large clusters of H_4 motifs the H_4 motif count will grow rapidly in time. The probability for the first event on the original motif is 0.5, and for the second event 0.5. A significant difference in the occurrence of the first event over the second event will encourage more of the first event in the future.

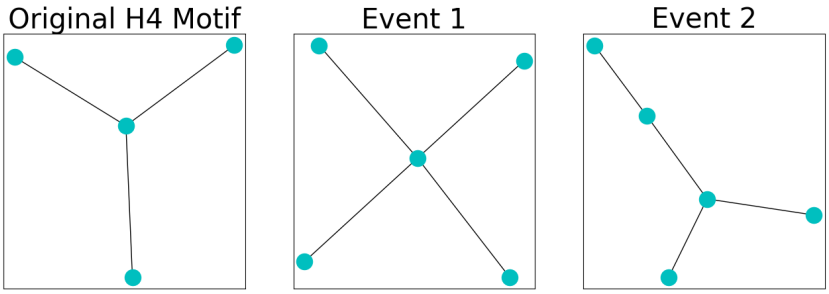

Figure 5.2. The possible graphs generated by adding a node to the H_4 graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2
H_3	0	0	2
H_4	1	4	1
H_5	0	0	0

Table 5.2. Motif Counts of the H_4 motif and the possible motifs given a $T2$ event.

5.3 The H_5 Motif

H_5's are of interest due to the relationship they carry to the H_4, H_7, and H_8 motifs. The induced subgraph isomorphic to the C_3 in the motif, means we find two appearances of H_5 in the H_7 and H_8 motifs. The H_4 is isomorphic to an induced subgraph of the H_5.The probabilities, given by a preferential attachment mechanism, of the events in Figure 5.3 the first event, 0.375, for the second event, 0.125, and the third event, 0.5. The last event only has the highest probability due to symmetry.

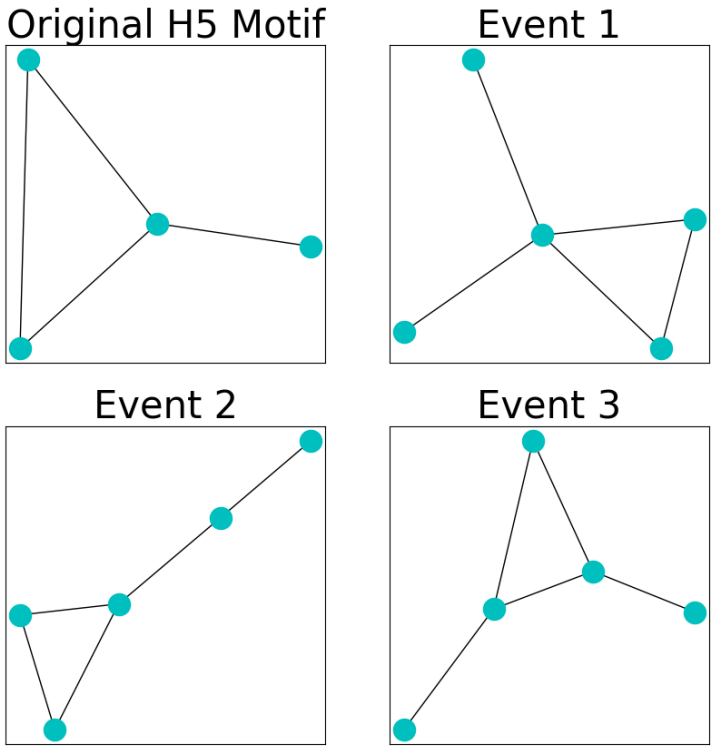

Figure 5.3. The possible graphs generated by adding a node to the H_5 graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2	Event 3
H_3	2	4	4	5
H_4	1	4	1	2
H_5	1	2	1	2
H_6	0	0	0	0
H_7	0	1	0	0
H_8	0	0	0	1
H_9	0	0	1	0

Table 5.3. Motif counts of the possible $T2$ events on the H_5 motif.

5.4 The H_6 Motif

The H_6 motif is formed by starting with an H_4 and adding two edges between the three outer nodes. It is almost a complete four-node graph. The H_6 motif count is not relatively high compared to other motifs in the Thij model as it would require exact $T3$ events to generate them. Moreover, this $T3$ event has to occur between what are likely non-root nodes. However, in the preferential attachment model for $m > 2$ it is more likely to see many H_6 appearances because the preferential attachment model adds multiple edges from a single new node. The possible $T2$ events on the H_6 motif are seen in Figure 5.4.

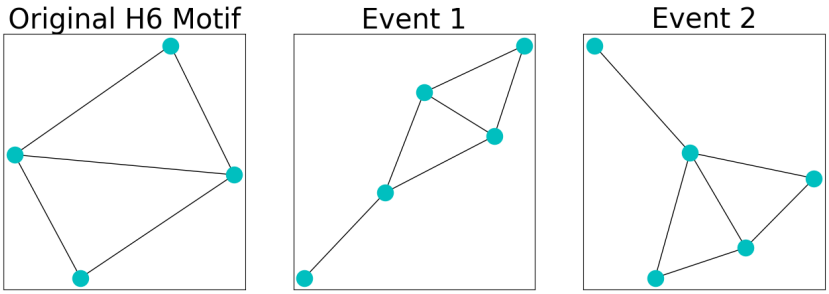

Figure 5.4. The possible graphs generated by adding a node to the H_6 graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2
H_3	6	10	10
H_4	2	3	5
H_5	4	5	6
H_6	1	1	1
H_7	0	0	2
H_8	0	2	2
H_9	0	1	1
H_{10}	0	2	0

Table 5.4. Motif counts for variations of the $T2$ event on the H_6 motif.

5.5 The H_7 Motif

H_7 motifs feature prominently in the Thij model for $0 < p \ll 1$. This is another consequence of S_k induced subgraphs in the networks. Connecting any two of the outer edges of S_k will generate $\binom{k-2}{2}$ H_7 motifs. Assuming a graph has formed with nodes attached to all three vertices of a C_3, the H_7 count is the sum of $\binom{d_i-2}{2}$ for $i = 1, 2, 3$ where d_i is the degree of a single vertex of the C_3. We subtract 2 from the degree d_i recognizing the connections to the other vertices in the C_3 graph. Now for each vertex in the C_3 we can pick two attached nodes not in the C_3. These two connected nodes and the C_3 make up an appearance of $H7$. We count these across all three nodes giving $\sum_{i=1}^{3} \binom{d_i-2}{2}$ Adding a node to any one of those vertices v_i in the C_3, assuming $d_i \geq 4$, will generate $d_i - 2$ new H_7 appearances, a new appearance for each node connected to vertex v_i. The $T2$ events acting on the H_7 motif are seen in Figure 5.5.

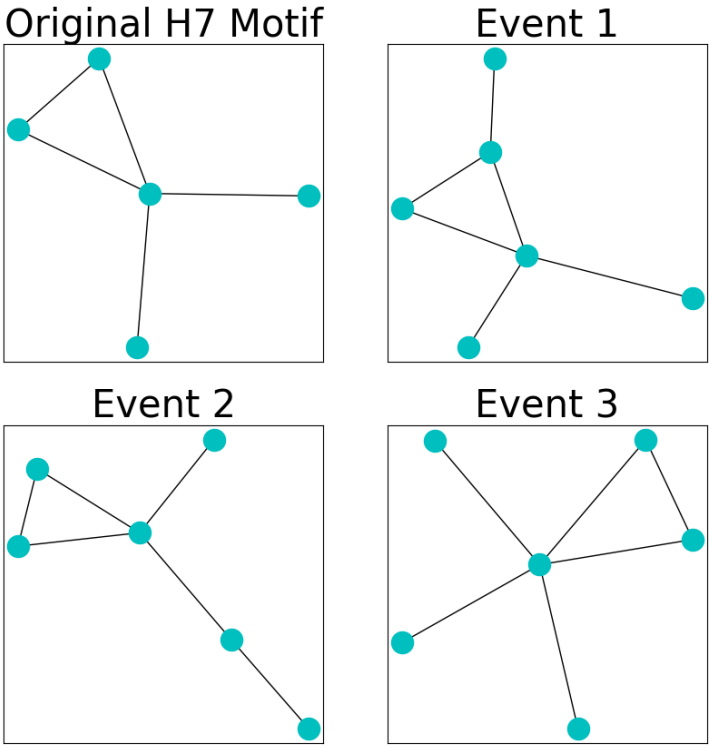

Figure 5.5. The possible graphs generated by adding a node to the H_7 graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2	Event 3
H_3	4	8	7	6
H_4	4	5	4	10
H_5	2	3	2	3
H_6	0	0	0	0
H_7	1	1	1	3
H_8	0	2	0	0
H_9	0	0	0	0
H_{10}	0	0	1	0

Table 5.5. Motif counts of the H_7 motif and the possible additions of $T2$ event nodes.

5.6 The H_8 Motif

The H_8 motif is another we expect to appear fairly often. The H_8 is two H_4's sharing an edge and a node. We can also characterize it as a C_3 graph with two nodes attached to distinct vertices on the C_3. Given a C_3 graph with at-least one node attached to each vertex, we have $(d_i - 2)(d_j - 2)(d_k - 2)$ $H_8's$. The number of new H_8's by connecting a node to vertex v_i is given by $(d_j - 2)(d_k - 2)$. In Figure 5.6, we see the possible graphs produced by a $T2$ event acting on the H_8 graph.

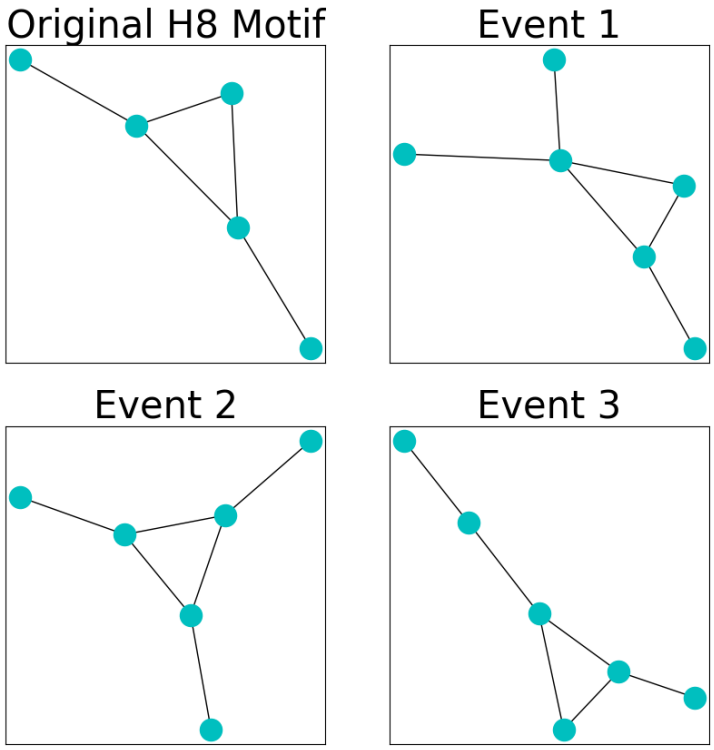

Figure 5.6. The possible graphs generated by adding a node to the H_8 graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2	Event 3
H_3	5	8	9	7
H_4	2	5	3	2
H_5	2	3	3	2
H_6	0	0	0	0
H_7	0	1	0	0
H_8	1	2	3	1
H_9	0	0	0	0
H_{10}	0	0	0	1

Table 5.6. Motif counts formed by the possible $T2$ events on the H_8 motif.

5.7 The H_9 motif

The H_9 motif appearances do not form around stars in the same way we might expect H_7 and H_8 appearances. The H_9 motif could develop in a way similar to the H_5. This is because the H_9 has an induced subgraph isomorphic to the C_4. The H_9 appearance can be produced by attaching a node to any one of those vertices in the induced subgraph. Given a C_4 graph and attachment of new nodes to any of its four vertices, the count of H_9's is simply the sum of the degrees of each vertex minus 2. The growth is additive, not combinatorial in the manner of H_7's or H_8's. In Figure 5.7, we see the possible graphs produced by a $T2$ event on the H_9 graph.

Motif Count	Original Motif	Event 1	Event 2	Event 3	Event 4
H_3	6	8	8	9	8
H_4	1	1	4	2	2
H_5	0	0	0	0	0
H_6	0	0	0	0	0
H_7	0	0	0	0	0
H_8	0	0	0	0	0
H_9	1	1	2	2	2

Table 5.7. Motif counts graphs formed by possible $T2$ events on the H_9 motif.

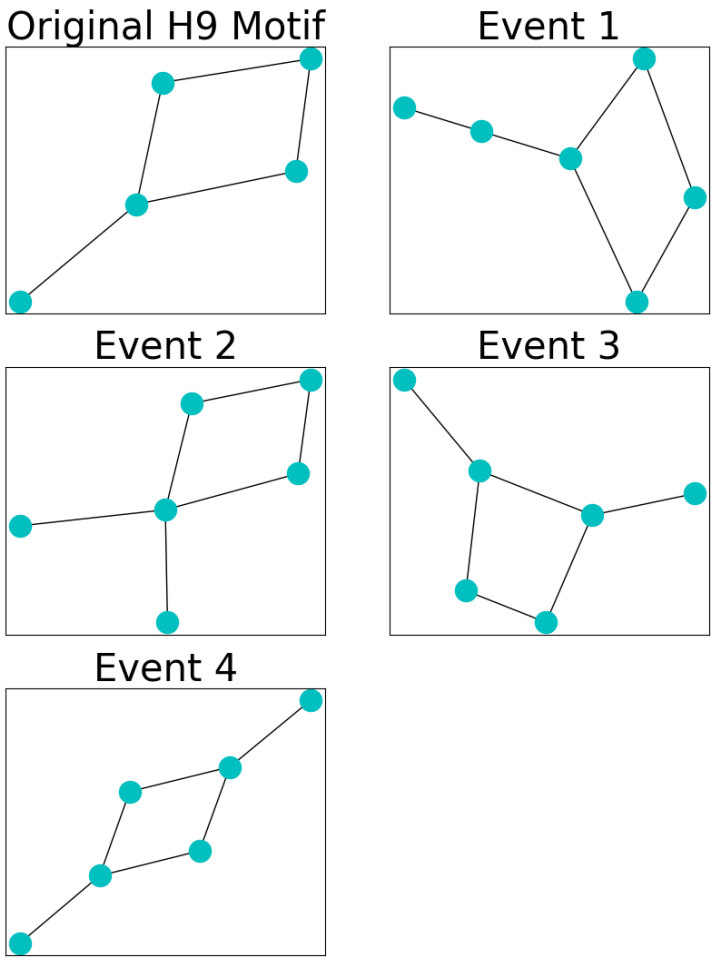

Figure 5.7. The possible graphs generated by adding a node to the H_9 graph and connecting it to an existing node.

5.8 The H_{10} Motif

The H_{10} motif graph is a H_9 graph with an extra vertex. There is an edge connecting that vertex to the single vertex of degree one in the H_9 graph as seen in Figure 5.8. Many H_{10} appearances could be generated from a single H_9 if vertices attach to the single vertex of degree one in the H_9 motif. A star graph would form with that vertex at the center. The growth for the H_{10} in the preferential mechanism with $k = 1$ is additive. For $k > 1$, it is possible clusters of triangles could form causing the

H_{10} motif count to increase faster than by the addition of a single node and edge at each time-step.

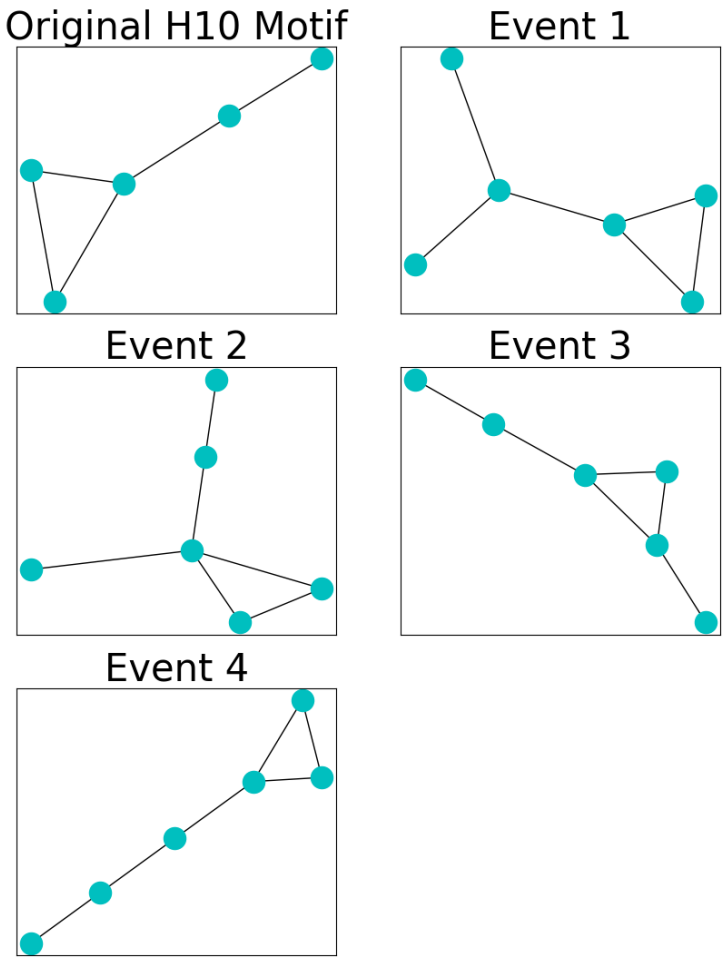

Figure 5.8. The possible graphs generated by adding a node to the H_{10} graph and attaching it to an existing node.

Motif Count	Original Motif	Event 1	Event 2	Event 3	Event 4
H_3	4	6	7	7	5
H_4	1	2	4	2	1
H_5	1	1	2	2	1
H_6	0	0	0	0	5
H_7	0	0	1	0	0
H_8	0	0	0	1	0
H_9	0	0	0	0	0
H_{10}	1	2	1	1	1

Table 5.8. Motifs counts of the possible $T2$ event on the H_{10} motif.

5.9 The H_{11} Motif

The H_{11} motif, shaped like a bow-tie, is formed by two three-walks that share a common vertex as seen in Figure 5.9. This motif does need $k \geq 2$ in the Barabási–Albert Model or a $T3$ event to form an edge between two existing nodes. The H_{11} motif does not appear commonly without those necessary criteria.

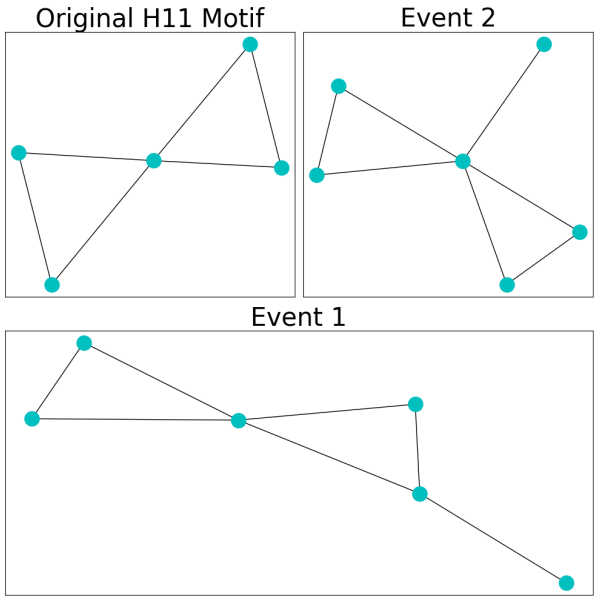

Figure 5.9. The possible graphs generated by adding a node to the H_{11} graph and connecting it to an existing node.

Motif Count	Original Motif	Event 1	Event 2
H_3	8	12	12
H_4	4	5	10
H_5	0	5	6
H_6	0	0	0
H_7	2	2	6
H_8	0	2	0
H_9	0	0	0
H_{10}	4	5	4
H_{11}	1	1	1

Table 5.9. Motif counts of the graphs generated by possible $T2$ events on the H_{11} motif.

5.10 The H_{12} Motif

H_{12}, shaped like a house and seen in Figure 5.10, contains five nodes, six edges, with an induced subgraph isomorphic to C_4 and a single vertex attached to two vertices themselves connected. The H_{12}, like other motifs, containing an induced subgraph isomorphic to C_4, is not a priori expected to have a relatively large count given the preferential attachment mechanism.

Motif Count	Original Motif	Event 1	Event 2	Event 3
H_3	10	14	13	14
H_4	2	5	3	3
H_5	2	3	2	3
H_6	0	0	0	0
H_7	0	1	0	0
H_8	1	2	1	3
H_9	2	3	3	2
H_{10}	2	2	3	2
H_{11}	0	0	0	0
H_{12}	1	1	1	1

Table 5.10. Variations of the $T2$ event on the H_{12} motif.

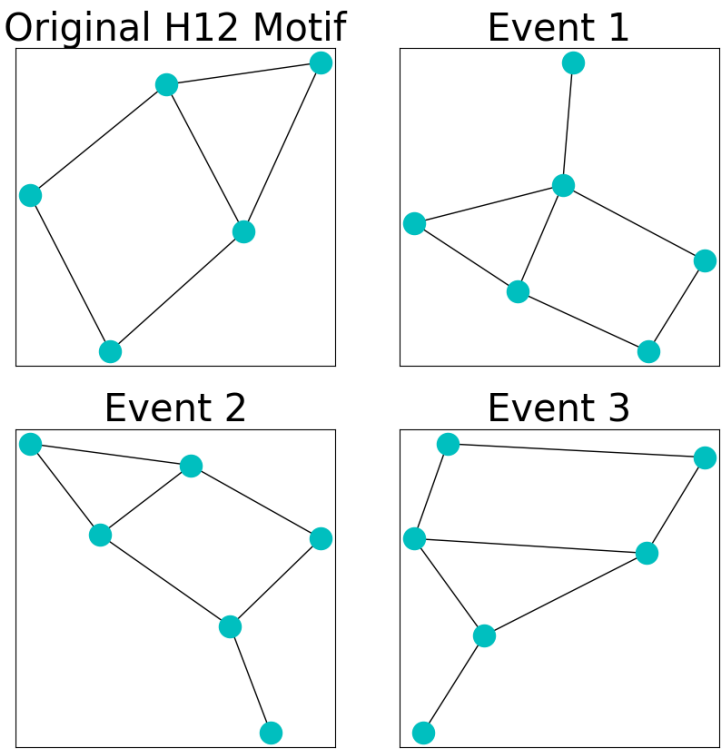

Figure 5.10. The possible graphs generated by adding a node to the H_{12} graph and connecting it to an existing node.

5.11 The H_{13} Motif

The H_{13} could plausibly form in the $k \geq 2$ case for the Barabási–Albert model, but it would require a new node consistently attached correctly to an H_6 appearance. For this process of generating new H_{13}'s, the increase in the motif counts is additive. Thus for the Barabási–Albert model of $0 < k < 3$, the presence of an H_{13} is sensitive to the initial graph. The visualization of the H_{13} graph and the possibe graphs formed by the $T2$ event can be seen in Figure 5.11.

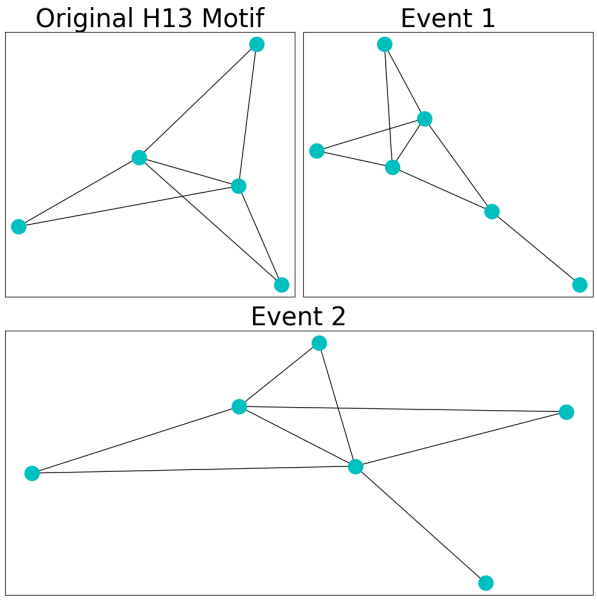

Figure 5.11. The H_{13} graph and possible node attachments up to symmetry.

Motif Count	Original Motif	Event 1	Event 2
H_3	18	24	24
H_4	8	14	9
H_5	12	15	13
H_6	3	3	3
H_7	6	12	6
H_8	6	12	10
H_9	6	9	8
H_{10}	0	0	4
H_{11}	0	0	0
H_{12}	0	0	0
H_{13}	1	1	1

Table 5.11. Motif counts of the possible $T2$ events on the H_{13} motif.

5.12 Summary of Preferential Attachment and $T2$ Event Motif Evolution

Motif development in the Barabási–Albert model is dependent on the initialization of nodes and the choice of k. Adding $k = 1$ edges for every new node limits the motifs that can appear. There will be no new H_6, H_{11}, H_{12}, or H_{13} appearances generated. There is no possible way for them to form as there is no node of degree one in those motif graphs. Only upon taking $k > 1$ could those motif counts change over time. Some graphs are more likely to have a high number of appearances for $k = 2$ than others, like the H_4, H_7, and H_8 motifs.

This analysis also applies to the $T2$ event in the Thij model. The occurrence is similar to the preferential attachment model with $k = 1$ as it is the addition of a node connected to a single existing node. The $T2$ event, unlike the BA model, selects a message tree and then uses a superstar attachment mechanism to attach to a node with probability $q = 0.9$. Even when the network is relatively small, nodes will still overwhelmingly attach to the root message node. This mechanism is much more likely to generate H_4's, as seen in Chapter 4. If the root message node is the vertex of a C_3 isomorphic subgraph, then we may see H_7 and H_8 counts rapidly increase over time, correlating with the H_4 count.

5.13 Twitter Model Specific Motif Evolution

In Chapter 4, we specified three possible events in the Thij model: a new root node, a new node with an attached edge, or a new edge between existing nodes ($T1$, $T2$, $T3$ respectively). The only events specific to the Thij model are $T1$ and $T3$ events. $T1$ events add a new message node, but they do not immediately affect any change in the motif counts. They could potentially have an impact on motif enumeration with the right $T3$ event or a sequence of $T2$ events. The $T3$ event we briefly consider because a $T3$ event can change the composition of the network in ways $T2$ events cannot.

The $T3$ event is a much more complex mechanism than the $T2$ event. $T3$'s can add edges *between* different motif graphs and to a single motif's graph. If two motifs are disjoint, a $T3$ event bridges them and acts as multiple $T2$ events on both motifs. This is still a simple case, but for the eleven non-simple cycle motifs considered there are 55 different possibilities of inter-motif interaction. If the $T3$ event bridges two large disjoint graphs, this could affect many different motif appearances simultaneously.

The $T3$ event opens up new possibilities for the model. C_3 and C_4 appearances are more likely to form around root nodes. A $T2$ event alone cannot generate new C_3 or C_4 appearances. We may see more H_6 up to H_{13} appearances because they require C_3

and C_4 appearances in the network in order to form. For $p = 0.2$, the Thij model exhibits significantly more clustering throughout time because of this $T3$ attachment mechanism. Measuring the impact of a small p parameter is for the tools of statistical analysis because there are too many cases to consider by graphical analysis.

CHAPTER 6

Covariance Analysis Between Motif Counts

Given the analysis in Chapter 5, a $T2$ or $T3$ event applied to any motif graph will generate a given amount of new motif appearances. The analysis of $T2$ events on single motifs suggests some motif counts like H_4, H_7, and H_8 should show a relationship in the time series data. We compute covariance matrices for the time series generated by the motif counts with the following definition.

Definition 6.1. *The element $C_{i,j}$ of the ith row and jth column of the covariance matrix $C \in \mathbb{R}^{n\times n}$ is defined to be the covariance between x_i and x_j, that is:*

$$C_{i,j} = \mathbf{E}((x_i - \mathbf{E}(x_i))(x_j - \mathbf{E}(x_j)))$$

$\mathbf{E}(x_i)$ denotes the expected value of the variable x_i. The value $C_{i,i}$ is called the variance of variable x_i.

The covariance matrix offers insight into the relationships between different motif counts throughout the time series. We want to measure how two motif counts change with respect to one another. The analysis in the figures below is used to support our graphical analysis of $T2$ type events in Chapter 5. We should also begin to understand how the prevalence of $T3$ events affects the motif counts. Given that the $T3$ event is more difficult to analyze using graph theory, statistical analysis should offer more information concerning the effects of the $T3$ event. We begin with a simulation of the Barabási–Albert model, whose covariance matrix gives the heat map in Figure 6.1.

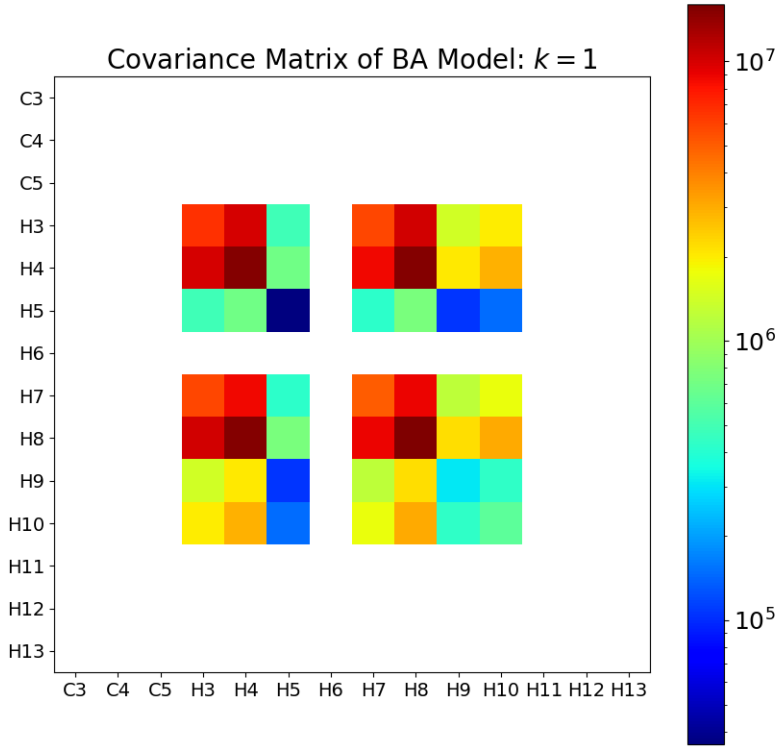

Figure 6.1. In this figure, we have Barabási–Albert model with eight initial nodes and $k = 1$.

The Barabási–Albert model for $k = 1$, is only capable of generating certain new motif appearances after the initialization, as it only adds one edge at a time. To grow certain motif counts requires an event similar to the $T3$ event, which can add edges between existing nodes. The only motif counts that increase after the initialization of the $k = 1$ Barabási–Albert model are H_3, H_4, H_5, H_7, H_8, H_9, and H_{10}. The formation of subgraphs isomorphic to the C_3 and C_4 graphs at initialization makes it possible for these counts to grow over time. One point of interest in the Barabási–Albert $k = 1$ simulation, is the extremely high covariance between the H_4 and H_8 motif counts. Each of these motif counts has a high variance as well. Why these coefficients are higher than the covariance between the H_7 motif count and either the H_4 or H_8 counts is not obviously attributable to the specific atttachment mechanism.

The strength of covariance between motif counts changes between simulations. The covariance coefficients change not only in scale. Which motif counts exhibit the

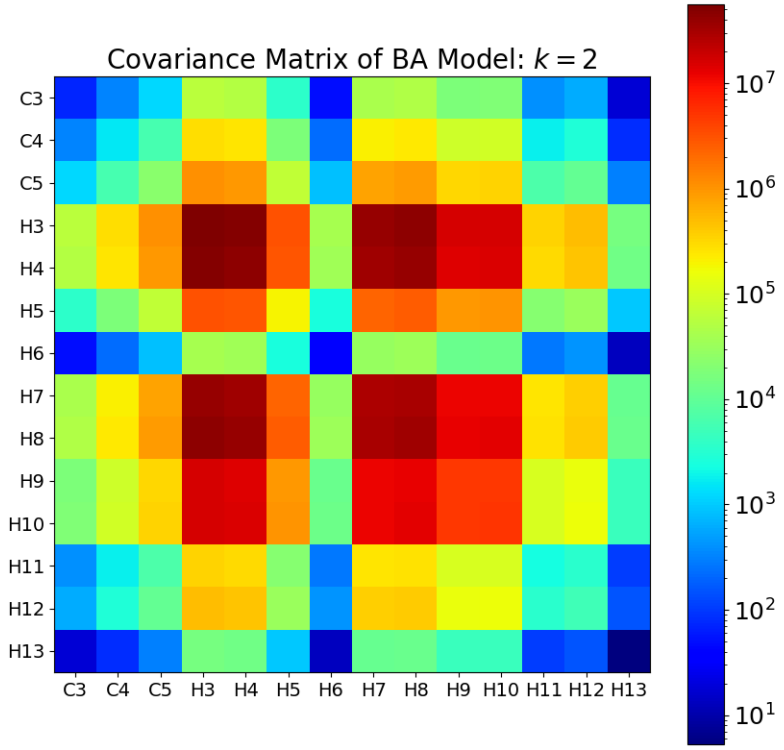

Figure 6.2. The Barabási–Albert model with $m = 3$ initial nodes and $k = 2$.The H_3, H_4, H_7 and H_8 motif counts exhibit high covariance.

largest covariances differs between simulations. In Figure 6.1 and Figure **??**, the covariance matrix supports our a priori graphic analysis in Chapter 5. The covariance matrices between the $k = 1$ and $k = 2$ Barabási-Albert simulations are more alike than different. In both models, there is a strong relationship between the H_3, H_4, H_7, and H_8 counts. The covariance matrices support the analysis of interactions between the H_4, H_7, and H_8 counts.

In Figure 6.3, a different set of motifs exhibit strong covariances when compared to the Barabási–Albert model. For this simulation, there is a higher chance of a $T3$ event occurring. The H_8 count has a very strong variance when compared to the other motif counts, and strong covariances with the H_7, H_9, H_{10} motif counts. The latter three motif counts also have relatively high covariances with one another. Additionally, the H_3 count shows a high covariance with the H_7, H_8, H_9, and H_{10} motif counts.

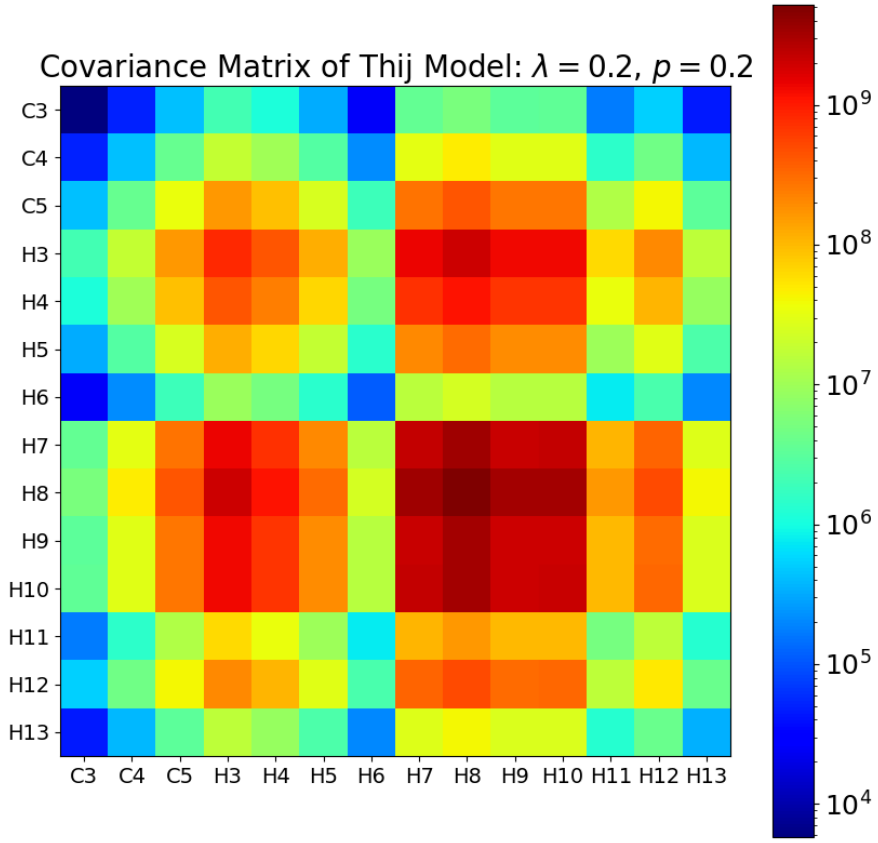

Figure 6.3. The covariance matrix of a Thij simulation with $\lambda = 0.2$ and $p = 0.2$. There is a strong covariance relationship between H_7, H_8, H_9, H_{10}.

The Thij simulation for $\lambda = 0.2$, $p = 0.8$ has an increased chance of $T2$ events. In Figure 6.4, we see that the variance of the $H4$ motif count is very high. In Chapter 5, we showed that the $H4$ count increases in a combinatorial explosion when attaching nodes to the root vertex. The H_4 variance is much higher than the covariance or variance between any other motifs.

The Thij simulation for $\lambda = 0.8$, $p = 0.2$, has an increased chance to add unconnected nodes and some chance to add edges between existing nodes. The Figure 6.5 shows the H_4, H_7, H_8, H_{10} motif counts have a strong covariance with the $H3$ motif count given these parameter values. The motif H_3 itself is of relatively high variance. For the H_7 motif count up the H_{10} motif count there is some positive

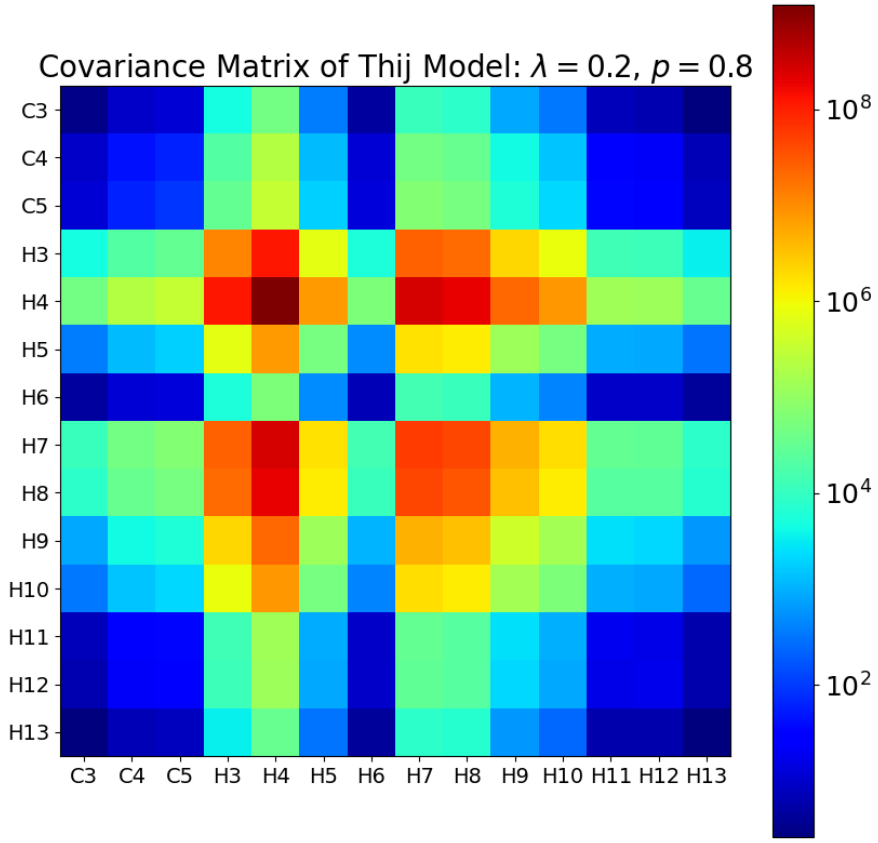

Figure 6.4. $\lambda = 0.2$ **and** $p = 0.8$**. The covariance coefficients are relatively very small, except for the** H_4 **variance.**

covariance. The covariance between these motifs was noticeably larger in the Thij model with $\lambda = 0.2$, $p = 0.2$ in Figure 6.3. We cannot say if it is the same mechanism at work.

The last Thij simulation, with $\lambda = 0.8$, $p = 0.8$, has an increased probability of $T1$ and $T2$ events. The covariance matrix, in Figure 6.6, shows a high variance in the $H4$ motif count, as well as strong covariance between the H_3 and H_4 counts. This covariance matrix looks similar to the covariance matrix produced for the Thij simulation $\lambda = 0.2$, $p = 0.8$. The scales between the two covariance matrices are seperated by a factor of 10^{-2}. This pattern in the covariance matrix could be evidence of the S_k, $k >> 1$ induced subgraphs for high values of p.

Overall we see that for parameter choices shown, the covariances of the motif counts in the Thij model differ from those shown in either of the Barabási–Albert

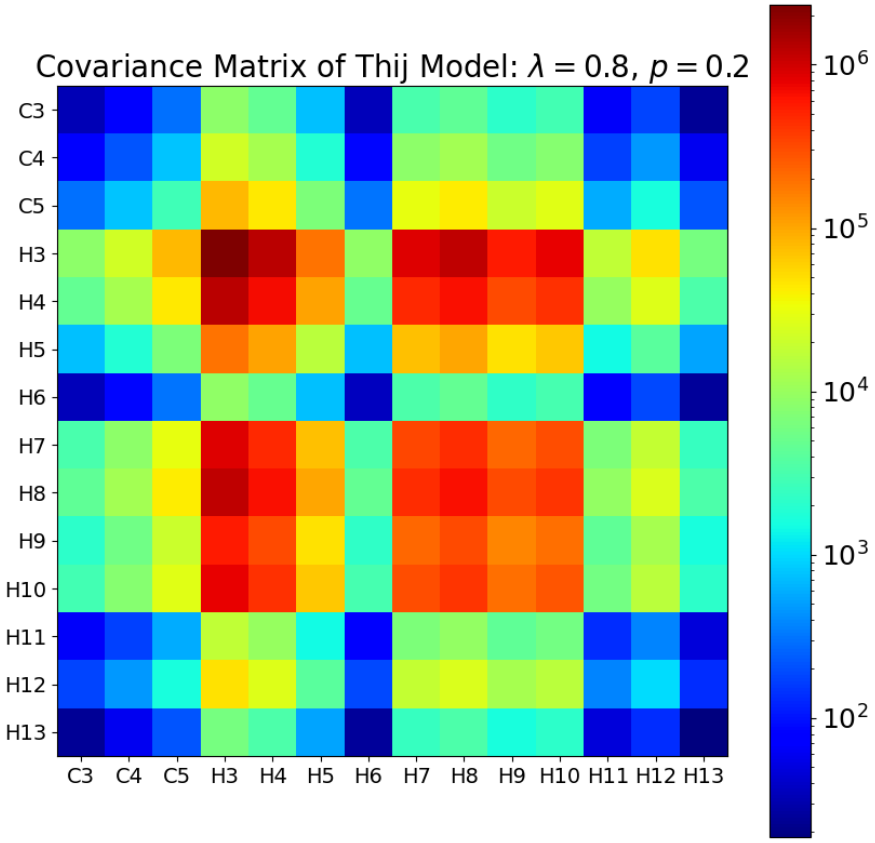

Figure 6.5. $\lambda = 0.8$ **and** $p = 0.2$**. There is strong covariance between the** H_3 **count with other motif counts.**

model simulations. Moreover, even the covariances across Thij simulations for different parameter values vary. The attachment events and their probability distributions have a pronounced effect on which motif counts correlate and which motif counts are the most dominant throughout the time-series data.

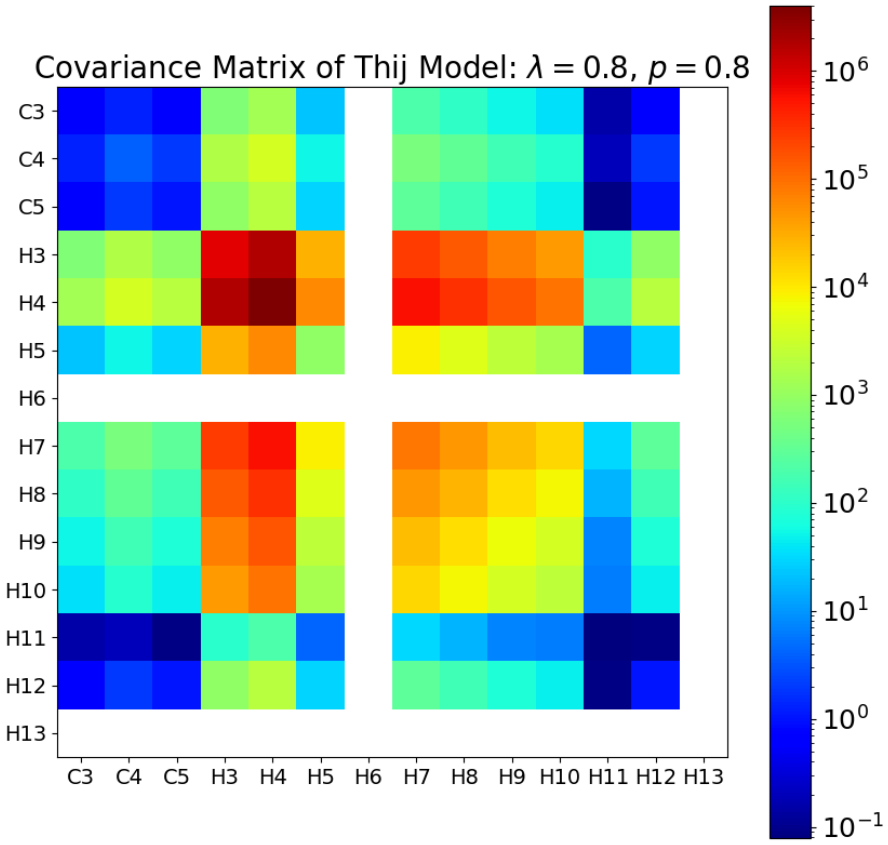

Figure 6.6. $\lambda = 0.8$ **and** $p = 0.8$**. The covariances of this simulation suggest once again an overlapping of** H_4 **graphs.**

CHAPTER 7

Dynamic Mode Decomposition

In addition to using statistical methods, we can view the motif counts through the lens of dynamical systems. Through Dynamic Mode Decomposition, we generate spatiotemporal coherent structures (modes). These modes have associated temporal behaviors: growth, decay, and oscillation. The modes provide insight into the underlying mechanics of the system. The DMD algorithm is closely connected to Koopman operator theory. This connection is made stronger through modifications to the DMD algorithm.

7.1 The Koopman Operator

Suppose we have a continuous, finite-dimensional, non-linear dynamical system

$$\frac{dy}{dt} = f(y) \quad y(0) = x \in \mathbb{R}^N$$

with $N \gg 1$. $y(t)$ is the state of the dynamical system at time t. Sampling the dynamical system every Δt we get the discrete time-series

$$y_{k+1} = F(y_k),$$

with $y_k = y(t_k) = y(k\Delta t)$. A non-linear dynamical system is generally difficult to solve by analytical means. We seek a new coordinate system where the dynamics could be described linearly. We seek a φ such that $z = \varphi(y)$ where the dynamics are much easier to evaluate in the z-coordinates. First, we define the Koopman operator.

Definition 7.1. *We define a Hilbert space of observables $L_2(O) = L_2(\mathbb{R}^N, \mathbb{R}, \mu)$, with an associated norm $\int_{\mathbb{R}^N} |g(x)|^2 d\mu(x) < \infty$, where μ is some appropriately chosen measure. The Koopman operator $K : L_2(O) \to L_2(O)$ is a mapping between the Hilbert space of observables unto itself, such that*

$$Kg(x_k) = g(F(x_k)) = g(x_{k+1}), \quad g \in L_2(O),$$

where $F : \mathcal{M} \to \mathcal{M}$, where $\mathcal{M}$ is a smooth n-dimensional manifold.

The Koopman operator is an infinite-dimensional, linear operator which advances the dynamical system forward a single step in time. The infinite dimensionality of the Koopman operator introduces difficulties. For this reason, we

examine the spectral decomposition of the Koopman operator [14]. For an eigenfunction $\varphi(x_k)$ of the Koopman operator K and its associated eigenvalue μ, we have

$$K(\varphi(x_k)) = \varphi(x_{k+1}) = \mu\varphi(x_k)$$

A vector of observables g can be written in terms Koopman eigenfunctions:

$$g(x_k) = \sum_{j=1}^{\infty} \varphi_j(x_k)\xi_j$$

which implies we can evolve the system like so

$$g(x_{k+1}) = K(g(x_k)) = \sum_{j=1}^{\infty} \mu_j\varphi_j(x_k)\xi_j$$

The eigenfunctions define a set of coordinates on which we can advance the measurements with a linear dynamical system. However, finding the exact eigenfunctions, modes, and eigenvalues of K analytically is difficult for any meaningful problem as the Koopman operator is infinite-dimensional. We can, however, seek to approximate a finite number of Koopman modes and eigenvalues. Thus we resort to numerics to find them via the Dynamic Mode Decomposition.

7.2 Dynamic Mode Decomposition

Suppose we have discrete samples of a non-linear, dynamical system. The discrete flow map is again denoted $x_{k+1} = F(x_k)$. These samples form a snapshot matrix X.

$$X = [x_1, x_2, \ldots, x_n]$$

Let $g(x_k)$ be a finite dictionary of observables formed from the state space. We introduce the operator A:

$$Ag = Ag(x_k) = g(x_{k+1}),$$

A advances measurements along the flow F by Δt. We calculate eigenvectors and eigenvalues of A in the following manner [10]. Beginning with our data snapshot matrix composed of the relevant observables, we write

$$G = [g(x_0), g(x_1), g(x_2), \ldots, g(x_n)] = [g_1, g_2, \ldots, g_n],$$

which we can then break into two matrices:

$$G_+ = [g_1, g_2, g_3, \dots g_n]$$
$$G_- = [g_0, g_1, g_2, \dots g_{n-1}]$$

The matrix G_+ is the matrix G_- taken forward one step in time. The matrix A above relates the two matrices.

$$G_+ = AG_-$$

Finding A means solving the optimization problem

$$\|G_+ - AG_-\|_F$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The solution to the optimization problem is found using the Moore-Penrose psuedoinverse.

$$A = G_+ G_-^\dagger,$$

where $G_-^\dagger$ is the psuedoinverse of G_-. We compute the r-rank Singular Value Decomposition (SVD) and generate a low-rank approximation of the matrix A which we shall call $\tilde{A}$.

$$\begin{aligned} G_+ &= \tilde{A}G_- \\ &= \tilde{A}U_r S_r V_r \quad \text{by SVD} \\ G_+ V_r^* S_r^{-1} U_r^* &= \tilde{A} \end{aligned}$$

U_r^* and V_r^* are the conjugate transposes of the matrices U_r and V_r. An eigendecomposition on the left-hand side results in the DMD modes ξ_j and eigenvalues μ_j. We can then propel the discrete dynamical system forward from the initial conditions b_n via

$$g_k \approx \sum_{n=1}^{r} b_n \mu_n^k \xi_n$$

We can also write a solution to the continuous dynamical system for all time:

$$g(t) \approx \sum_{n=1}^{r} b_n e^{\frac{\ln(\mu_n)t}{\Delta t}} \xi_n$$

Once we have the DMD modes and eigenvalues, we can begin to look at which DMD modes "contribute the most" throughout the snapshot matrix by examining the associated eigenfunction values and eigenvalues which describe their temporal behavior: growth, decay, or oscillation. We have not discussed how to choose g. If we choose g to be the identity mapping, then we describe the method as the standard DMD algorithm. Choosing any other function of g in a meaningful way requires prior knowledge of the system or trial-and-error. A mapping g can extend the dictionary of observables beyond the state space using an appropriate basis. This is aptly called Extended Dynamic Mode Decomposition (EDMD) [14]. A quick example of such a choice might be:

$$Y = [y_1, y_2, y_3, \ldots, y_n], \quad y_i \in \mathbb{R}^N$$

Y being the snapshot matrix of y at each time-step.

$$G(Y) = [g(y_1), g(y_2), g(y_3), g(y_4), \ldots, g(y_n)]$$

$$g(y_i) = [y_{i,1}, y_{i,2}, \ldots, y_{i,N}, (y_{i,1}^2), (y_{i,1}y_{i,2}), \ldots, (y_{i,N}^2)]$$

This example of g is one of many possible functions. EDMD strengthens the connection between Dynamic Mode Decomposition and Koopman Operator Theory. Given a choice of g such that $\phi_j \in \text{span}\{g_k\}$ then the eigenvalues of the Koopman operator are the DMD eigenvalues [14].

7.3 Kernel Dynamic Mode Decomposition

EDMD can quickly generate a large set of observables and the computational complexity of the problem can increase rapidly. Extending the state-space with a large dictionary of observables can generate very large matrices [23]. For that reason, one applies the kernel trick. Instead of evaluating the high dimensional state space directly, one can take inner-products of the state space using kernel functions to "collapse" the information of many nonlinear terms to a single value [14]. We introduce the notion of the kernel.

Definition 14. *We define the kernel function* $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$, $k(x, \hat{x}) = \langle \phi(x), \phi(\hat{x}) \rangle$. *$k$ maps a pair of vectors to an inner product of observables of the data.*

There are a variety of different kernels that we may choose. In this thesis, we use the polynomial kernel.

$$k(x, x') = (1 + x^T x')^p$$

Kernel functions allow us to store a large amount of information in an inner product. We generate matrices Φ^+ and Φ^- from the polynomial kernel function. Given our snapshot matrix Y follows:

$$Y = [y_1, y_2, y_3, \dots y_n]$$

Then the kernel between two snapshots y_i, and y_j is

$$k(y_i, y_j) = (1 + y_i^T y_j)^p.$$

We then construct the matrices $\Phi^+, \Phi^- \in \mathbb{C}^{n \times n}$.

$$\Phi^+_{i,j} = k(Y_i^+, Y_j^-)$$

$$\Phi^-_{i,j} = k(Y_i^-, Y_j^-)$$

where Y_i^+ is the ith column of the matrix Y^+ and Y_j^- the jth column of Y^-. Then, for every element in $\Phi^+_{i,j}$ and $\Phi^-_{i,j}$, we have a kernel between two snapshots in time. We find our KDMD modes as outlined in [23].

Given our matrices Φ^+, Φ^- we compute the Singular Value Decomposition of $\Phi^- = U\Sigma V^T$. We then compute the matrix $\hat{K}$

$$\hat{K} = (\Sigma^{-1}U^T)\Phi^+(U\Sigma^{-1})$$

We find the eigenvectors of A found in the columns of the matrix Ξ.

$$Y = \Psi_x \Xi$$

where the column vectors of Ψ_x are the approximated eigenfunction values. The numerical approximation to the eigenfunction values is obtained by the matrix multiplcation $U\Sigma_r\hat{\Xi}$.

7.4 Accuracy Criterion

For the DMD and KDMD algorithms, we evaluate how well the methods performed. We consider two different measures of accuracy. First, we define the one-step reconstruction error u. We define the matrix $\tilde{G}$ where the jth column vectors of $\tilde{G}$ are the one-step reconstructions of the state space at time j for $j = 1, 2, \dots, n$.

$$\tilde{G}_j = \sum_{k=1}^{r} \mu_k \varphi_k(x_{j-1})\xi_k$$

$$u = \frac{\|G_+ - \tilde{G}\|_F}{\|G_+\|_F}$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

We also wish to know if the approximated modes and eigenvalues are good approximations to Koopman modes. We define the Rowley measure [25]. Let ξ_j and μ_j be a Koopman eigenfunction for the dynamical system F and its corresponding eigenvalue. Provided that φ_j and μ_j are a true eigenpair it follows:

$$\varphi_j \circ F = \mu_j \varphi_j$$

Now, letting $\|\cdot\|$ denote the $L2$ norm, we would like to calculate

$$\frac{\|\varphi_j \circ F - \mu\varphi_j\|}{\|\varphi_j\|}$$

However, if we know F then DMD is not a useful tool. We must estimate F using a finite number of data points. We take two data points $x_k, x_{k+1} \in X$. We can now write

$$\tilde{r}_j = \frac{\sum_k |\varphi_j(x_{k+1}) - \mu_j\varphi_j(x_k)|}{\sum_k |\varphi_j(x_k)|}$$

This equation measures how much each eigenfunction φ_j behaves like a Koopman eigenfunction. We can use this to evaluate each mode individually and could potentially use it to select modes for low-rank reconstruction [25]. If $\tilde{r}_j$ is close to one, the eigenpair is unreliable because the difference in the eigenpair is of the same magnitude of the eigenfunction. Therefore a good eigenpair will have a mode error such that $0 < \tilde{r}_j \ll 1$.

7.5 Preprocessing Data for DMD and KDMD

Dynamic Mode Decomposition is sensitive to noise in the data. We implement several techniques to improve the errors of the DMD modes and eigenvalues. From our snapshot matrices of motif counts, we generate a Poisson process. A process is said to be Poisson if the events occur independently of one another and the number of events in any interval of time has Poisson distribution. k is the number of occurrences of events within a given interval of time. The interarrival times X_i are independent and distributed according to an exponential distribution for $i = 0, 1, 2, \ldots n$. The exponential distribution probability density function $g(x)$ is defined to be:

$$g(x; \gamma) = \gamma e^{-\gamma x}$$

We generate new snapshot matrices by generating interarrival time between events according to the exponential distribution, with $\gamma = 10$, thereby generating a Poisson process. It is this second snapshot matrix on which we apply the DMD and KDMD algorithm. After we distribute events in time according to this process, we smooth the data with a window average. The window we take to be between 500 to 1,000 snapshots depending on the length of the Poisson process data. This smoothing will help the DMD and KDMD to generate better fits. Lastly, we center the data about its mean.

CHAPTER 8

Results

Now we consider the results of the DMD and KDMD algorithms for different simulations of the BA model and the Thij model. The KDMD algorithm suffers the curse of dimensionality. For that reason, we run the DMD and KDMD algorithms on approximately the last 10,000 data snapshots for comparable results. Furthermore, when applying the SVD we truncate with a threshold α where keeping those singular values which satisfy $\log_{10}\left(\frac{\sigma_i}{\max(\sigma_i)}\right) > -\alpha$. This limits the number of total modes included in the results. In this thesis, we take $\alpha = 3$ across all simulations. We will first examine DMD and KDMD applied to the Barabási–Albert motif counts. Thereafter, we examine several parameter choices for the Thij model. For all DMD and KDMD results, the DMD and KDMD modes and eigenvalues are as discussed in Chapter 7. The φ_j modes tell us how much of the signal each respective Koopman mode contributes at a given time-step. We calculate $\varphi(x_k)$ by solving for Φ from the matrix equation $G_- = \Phi\Xi$, where Ξ is the matrix whose column vectors are DMD modes. For the KDMD algorithm we calculate $\varphi(x_k) = U\Sigma_r\hat{\Xi}$ [23].

8.1 Barabási–Albert $k = 1$

Examining the DMD modes in Figure 8.2 and Figure 8.3, we see that both sets of DMD and KDMD modes show a structure that connects the H_4 count with the H_7 and H_8 counts. Examining the DMD Phi modes, we see that the DMD modes 6 and 7 are the strongest through the whole time series. The last two KDMD modes pick up this connection between the H_3, H_4, H_7, and H_8 counts. Upon comparison with Figure 6.1, we see that the DMD and KDMD modes do reflect the strong covariances between these motifs.

In Figure 8.1, DMD has produced five modes of low-mode error. The seventh and eighth modes have mmode errors close to one. The remaining nodes 8-14 have eigenvalue zero and thus do not exhibit any dynamics. In [25], the authors remark that, generally, those modes with a zero eigenvalue are not of interest because they exhibit no dynamics. The KDMD modes show relatively good mode errors. The one-step reconstruction error for DMD is 0.97 and for KDMD the one-step reconstruction error is 0.036. DMD performs moderately wellin one-step reconstruction accuracy and finds

five modes with low mode error. KDMD has good reconstruction error and finds five modes of low mode error.

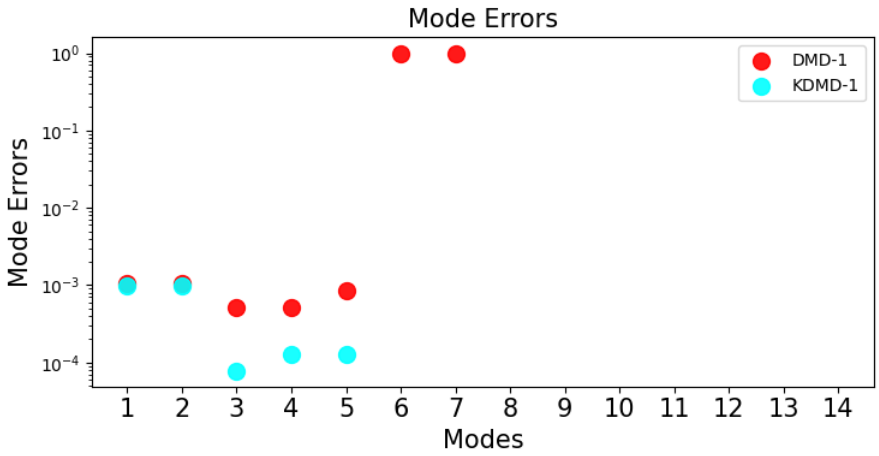

Figure 8.1. DMD and KDMD mode errors for the Barabási–Albert model with $k = 1$.

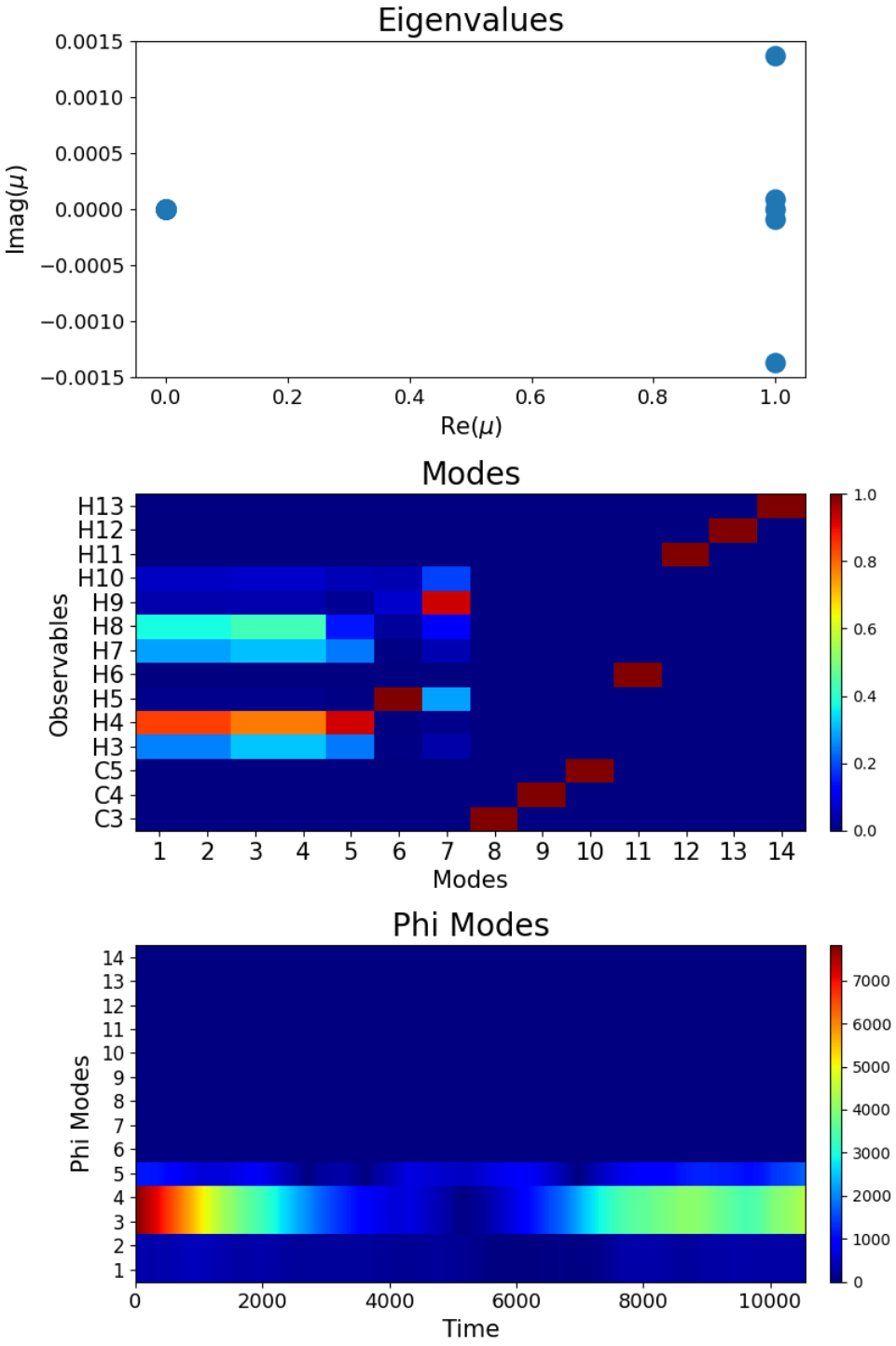

Figure 8.2. DMD modes, eigenvalues, and phi modes for the Barabási–Albert model with $k = 1$.

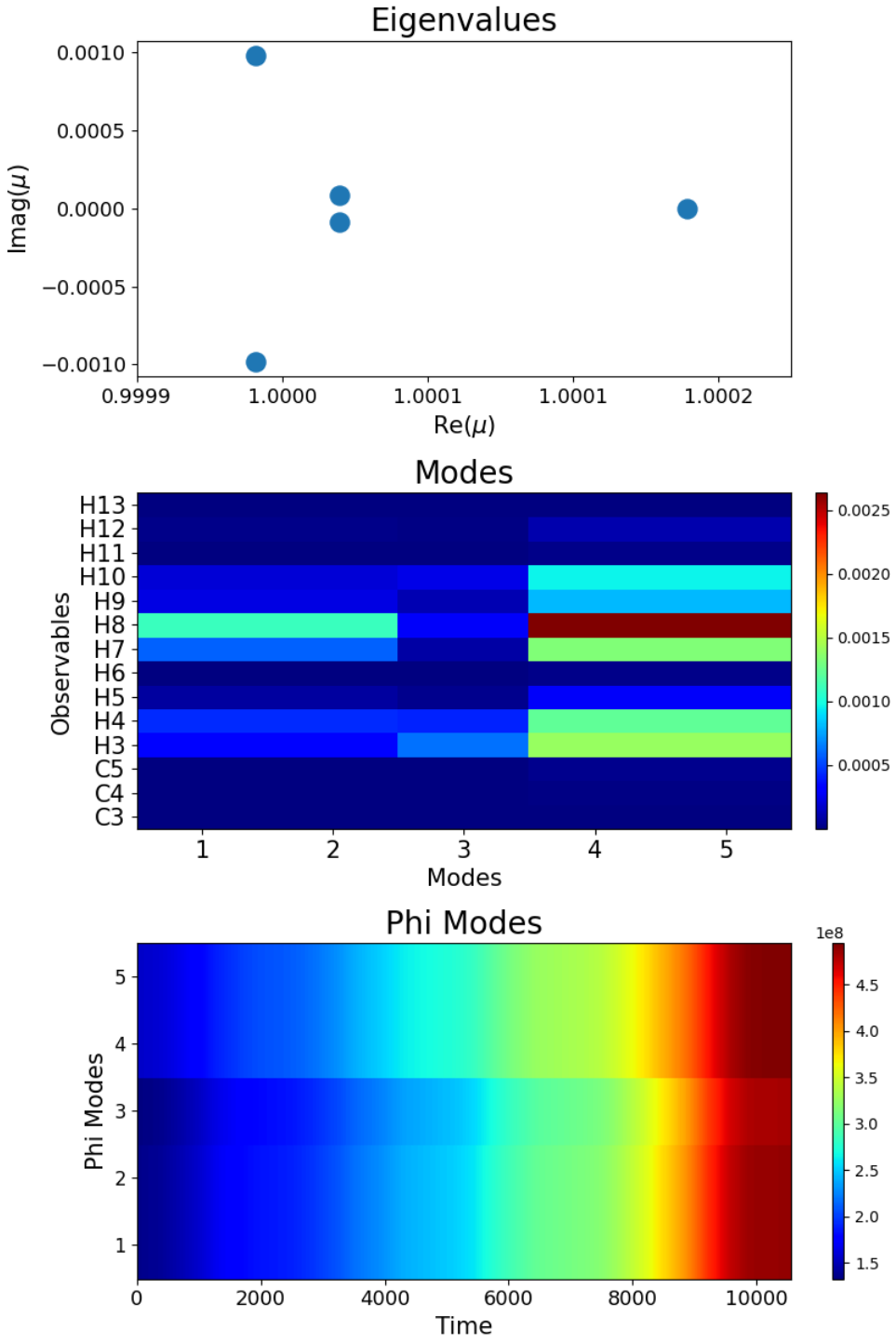

Figure 8.3. KDMD modes, eigenvalues, and phi modes for Barabási–Albert model with $k = 1$.

8.2 Barabási–Albert Model with $k = 2$

For the Barabási–Albert model with $k = 2$, the DMD algorithm picks out a single dominant mode, mode 8. If we examine this mode, in the second plot of Figure 8.5, the mode shows a strong connection between the H_3 and H_4 counts, as well as the H_7 and H_8 counts. The next two pairs of modes (modes 4, 5, 6, and 7) contribute a small amount to the signal as seen in the Phi mode plots, but show a similar structure to mode 8.

The KDMD algorithm has produced 5 modes in Figure 8.6. The SVD truncation has removed many POD modes based on their singular values. The Phi modes are approximately of the same magnitude throughout the time series. KDMD mode pair 2 and 3, show a strong connection in the dynamics of the H_3, H_4, H_7, and H_8 counts. The remaining modes show a similar connection, but it is not quite as strong. The associated eigenvalues cluster around the point $(1, 0)$ on the unit circle.

The DMD algorithm produced six modes with relatively low mode error, seen in Figure 8.4. The remaining modes (modes 1 and modes 9 through 14) have mode errors on the order of one, signaling they are not good approximations to true Koopman modes. The KDMD algorithm has produced a small set of modes, whose mode errors range between 10^{-2} and 10^{-4}, signaling that they are good approximations.

The one-step reconstruction error for DMD is 0.97 and for KDMD the one-step reconstruction error is 0.0156. For this particular simulation, KDMD does reconstruct the data fairly well.

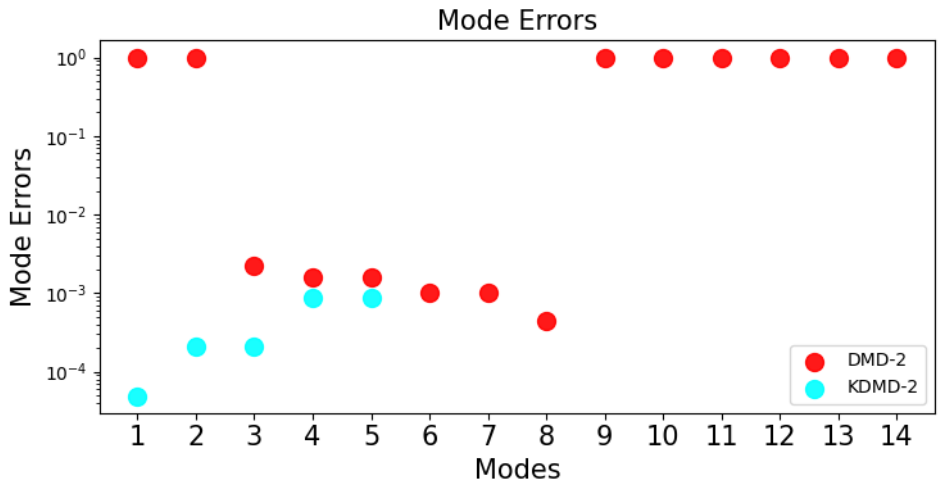

Figure 8.4. DMD and KDMD mode errors for the Barabási–Albert model with $k = 2$.

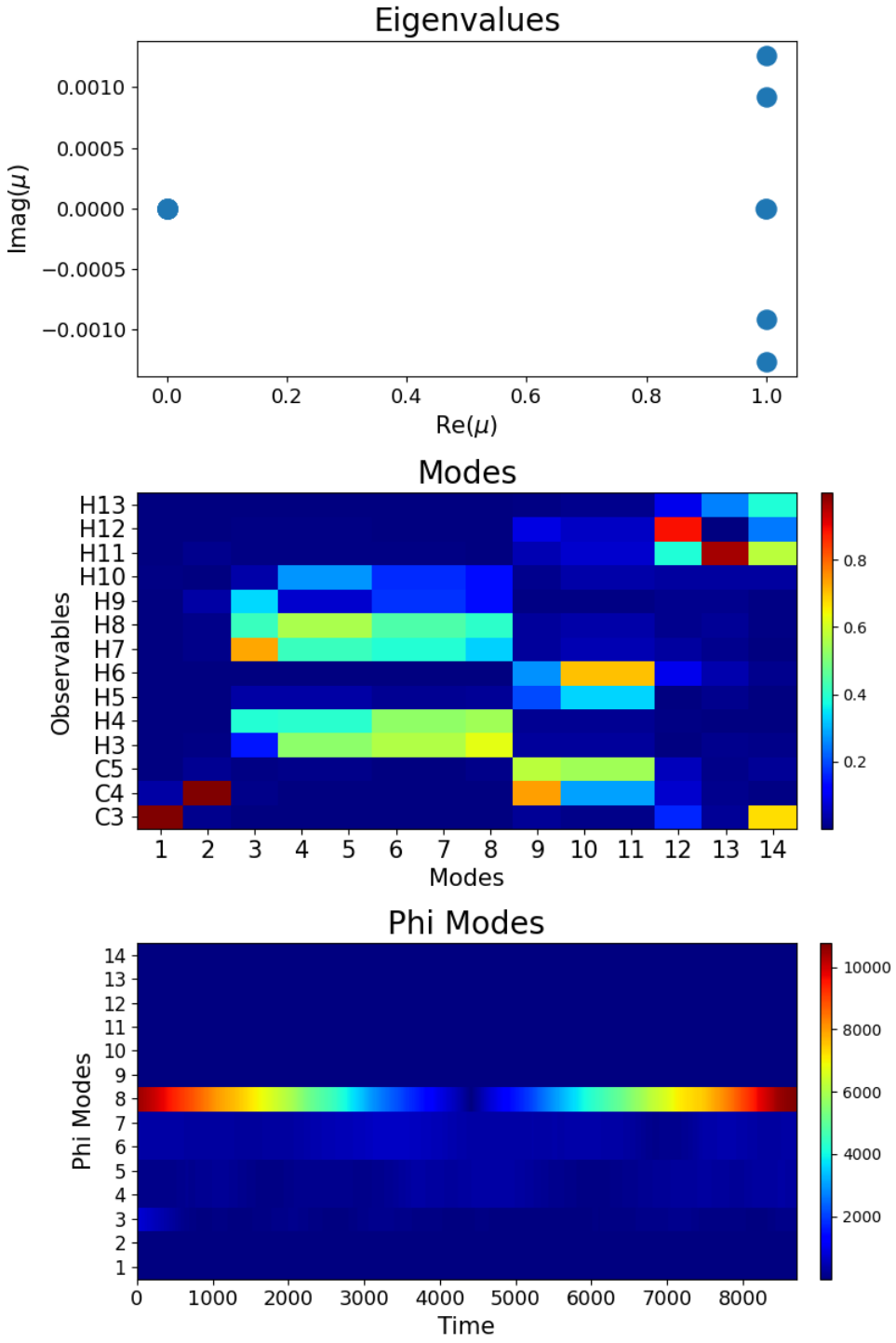

Figure 8.5. DMD modes, eigenvalues, and phi modes for the Barabási–Albert model with $k = 2$.

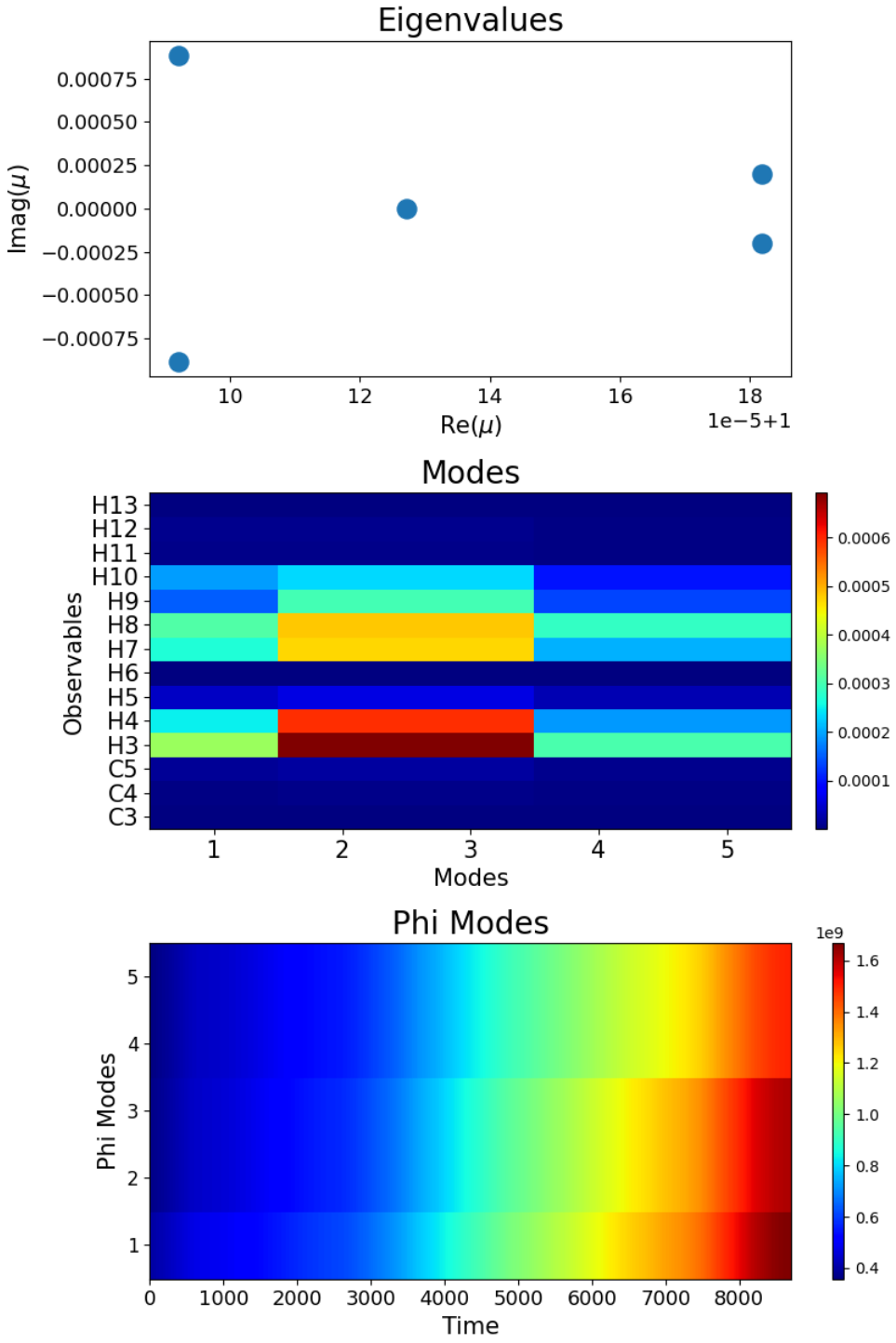

Figure 8.6. KDMD modes, eigenvalues, and phi modes for the BA model with $k = 2$.

8.3 Thij Model with $\lambda = 0.2$, $p = 0.2$

Analyzing the Phi mode plot in Figure 8.8, we see that mode 4 features the most heavily throughout the time series. Mode 4 shows a strong connection between the counts of H_7, H_8, H_9, and H_{10}. This mode is purely associated with growth as its eigenvalue has no imaginary part. From the Phi mode plot, the only other modes contributing to the signal are modes 1, 2, and 3. In Figure 8.7, we see these are the only modes with low mode error, while the other mode errors are approximately one, meaning they do not behave like true Koopman modes.

KDMD has picked out a similar structure, as the first and second KDMD modes in Figure 8.9, show a stronger connection between H_3 and the modes H_7, H_8, H_9, and H_{10}. Furthermore, all KDMD modes are very tightly clustered around $(1, 0)$ on the unit circle. We see that by truncating the SVD, KDMD has only produced four modes. All of these modes have good mode error, with two of them on the order of 10^{-4}.

The one-step reconstruction error for DMD is 0.99. For KDMD the one-step reconstruction error is 0.108. KDMD selects modes that have low-mode error and gives comparable reconstruction error.

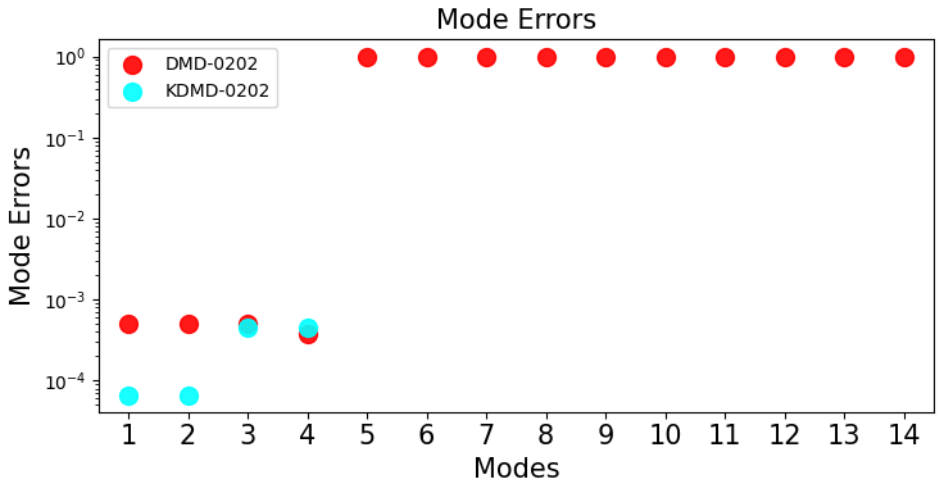

Figure 8.7. DMD and KDMD mode errors the Thij model with $\lambda = 0.2$, $p = 0.2$.

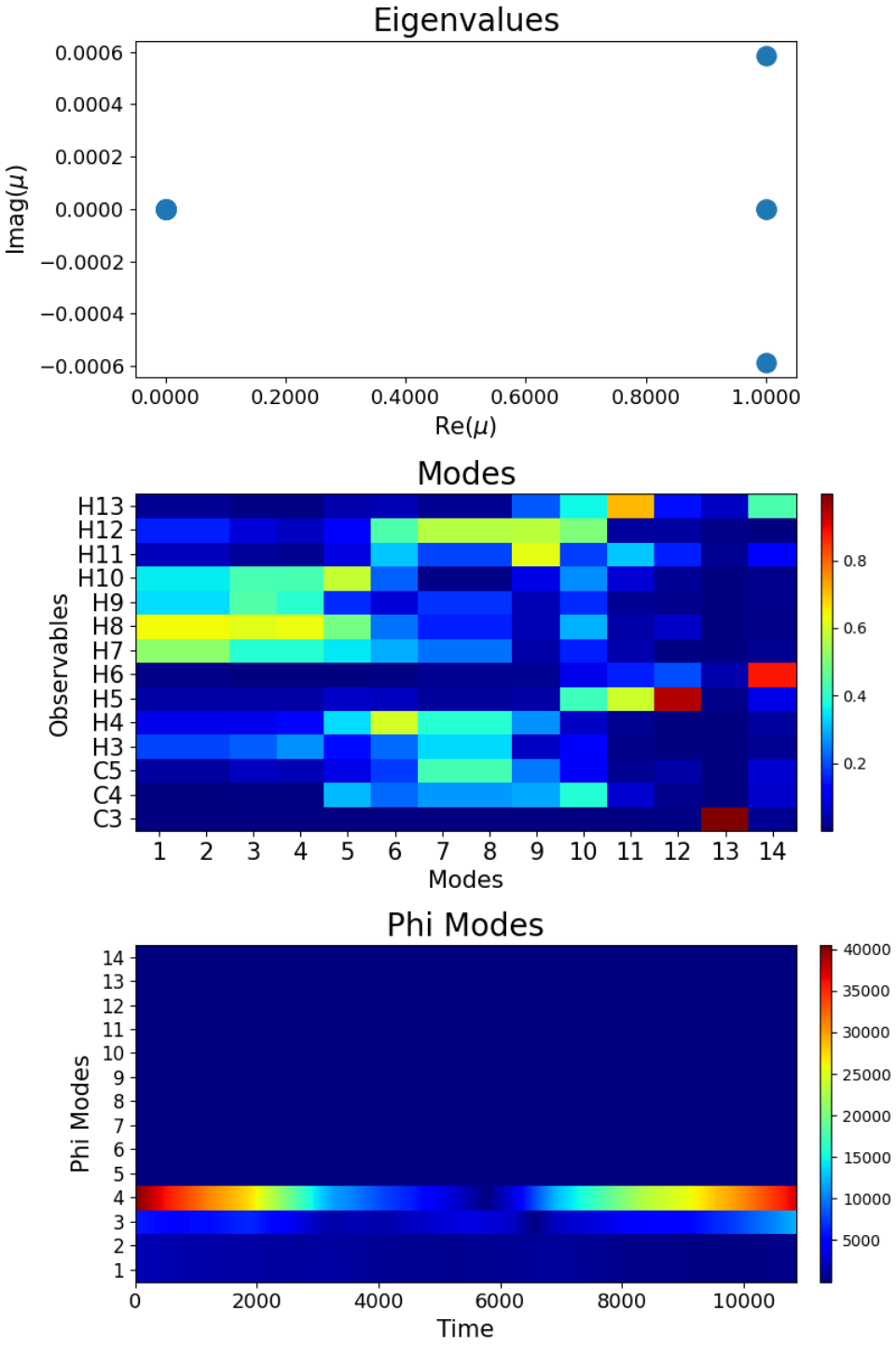

Figure 8.8. DMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.2$, $p = 0.2$.

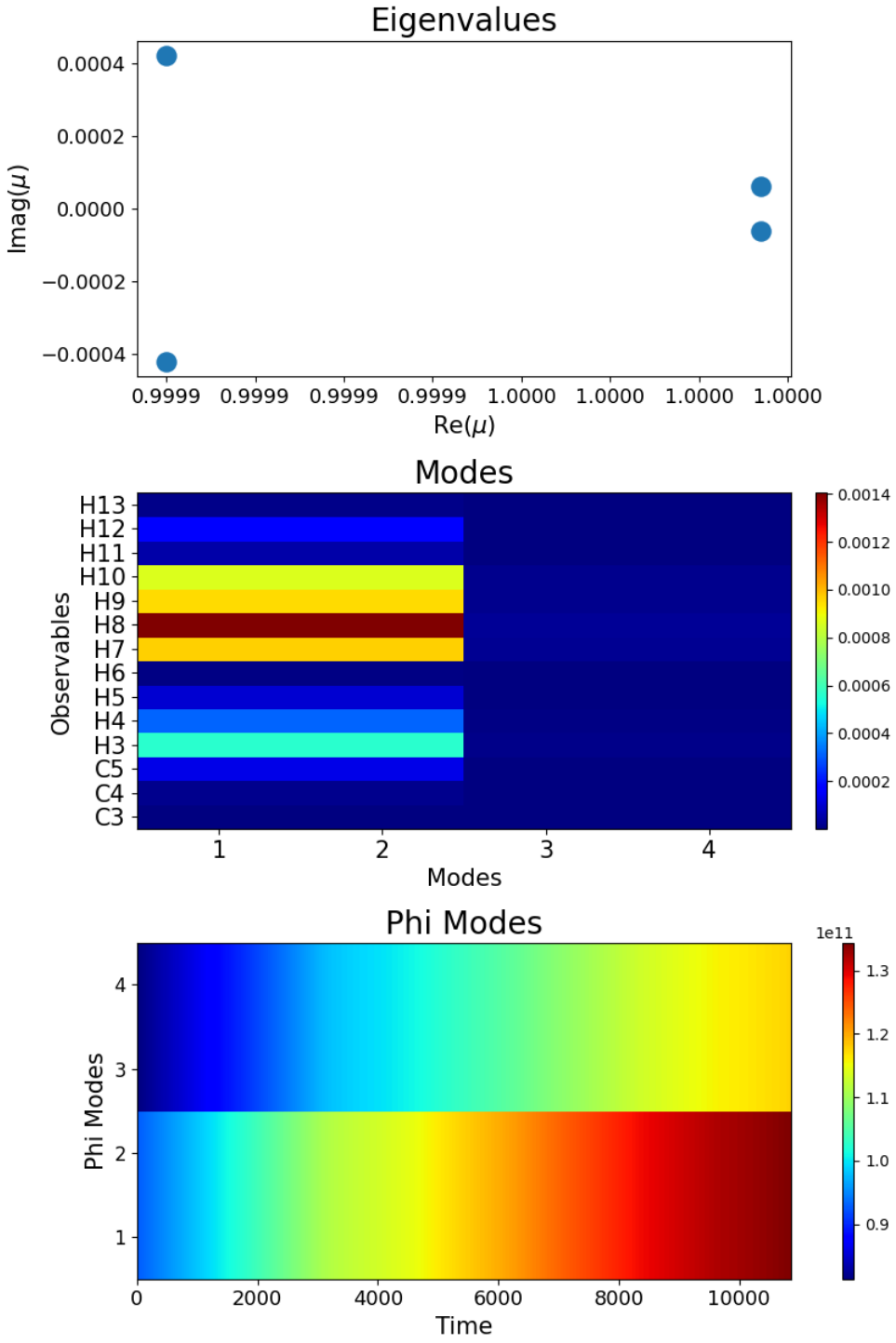

Figure 8.9. KDMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.2$, $p = 0.2$.

8.4 Thij Model with $\lambda = 0.2$, $p = 0.8$

For the Thij model with $\lambda = 0.2$, $p = 0.8$, DMD picks out three dominant DMD modes as seen in the Phi modes plot in Figure 8.11. Mode 3 is associated with a real eigenvalue and is thus associated with pure exponential growth in the system. The third DMD mode shows a dominant H_4 count with a soft association with H_3, H_7, and H_8 counts. Modes 4 and 5 show a similar relationship, but their complex pair eigenvalues account for small oscillations.

The KDMD algorithm is truncated such that it only produces five modes. In Figure 8.12, these modes show a relationship between the H_4 count and the H_3, H_7, and H_8 counts. The H_4 count is dominant.

The mode errors in Figure 8.10, show that many of the DMD modes have high mode errors. The first five DMD modes are relatively good mode approximations by the Rowley measure. All KDMD modes show low mode error of $10^{-}3$ or less. The one-step reconstruction error for DMD is 0.96 and for KDMD the one-step reconstruction error is 0.053. KDMD outperforms in reconstruction error and finds modes which behave like Koopman modes

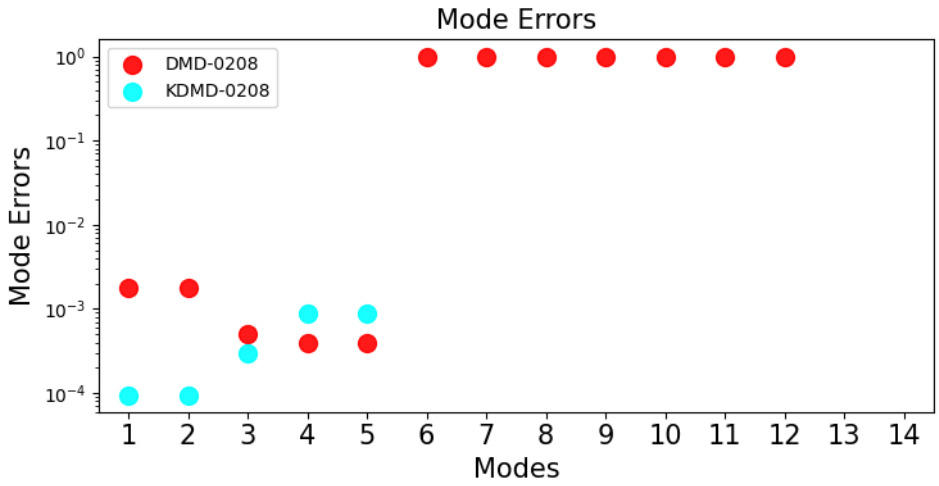

Figure 8.10. DMD and KDMD mode errors the Thij model with $\lambda = 0.2$, $p = 0.8$

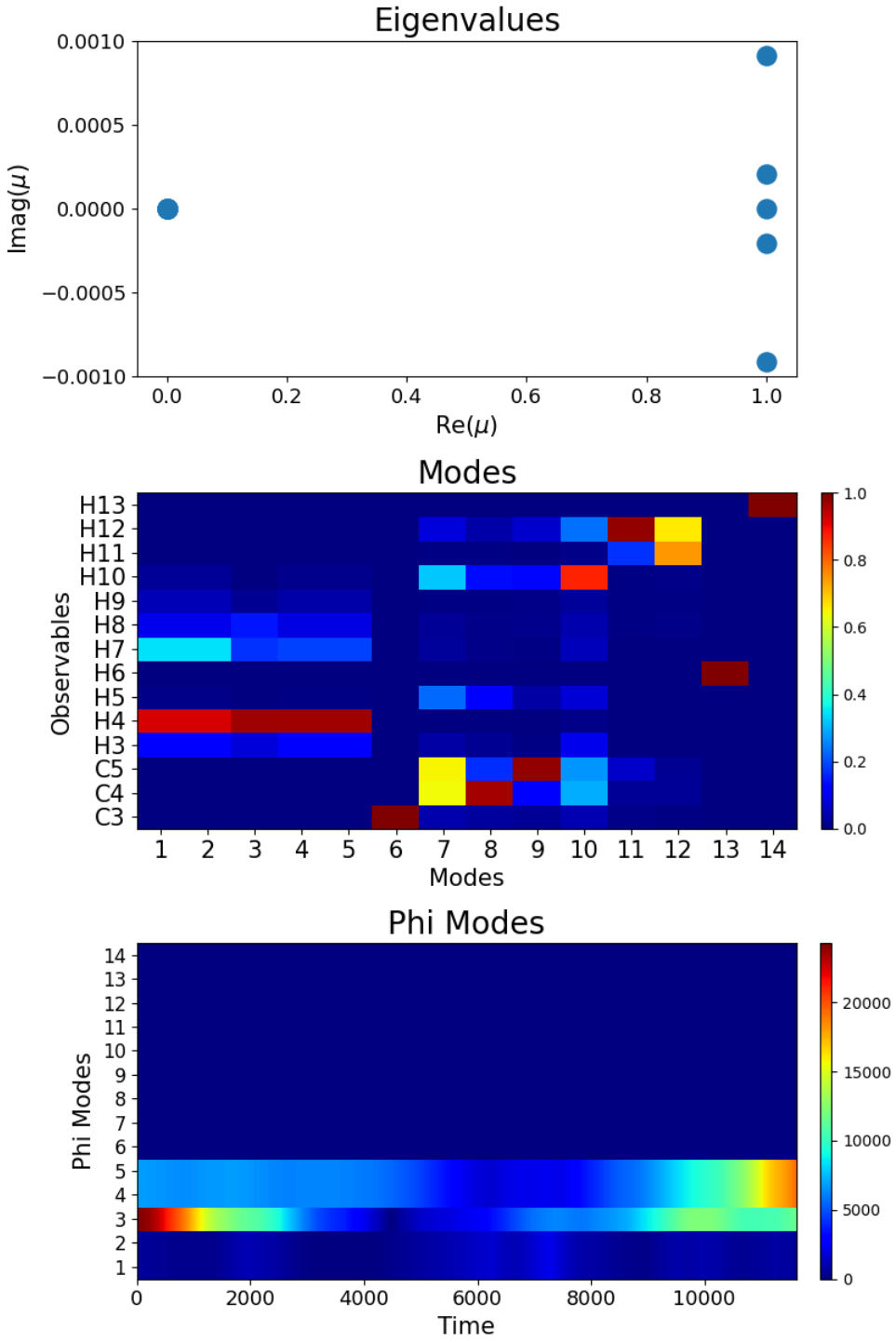

Figure 8.11. DMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.2$, $p = 0.8$.

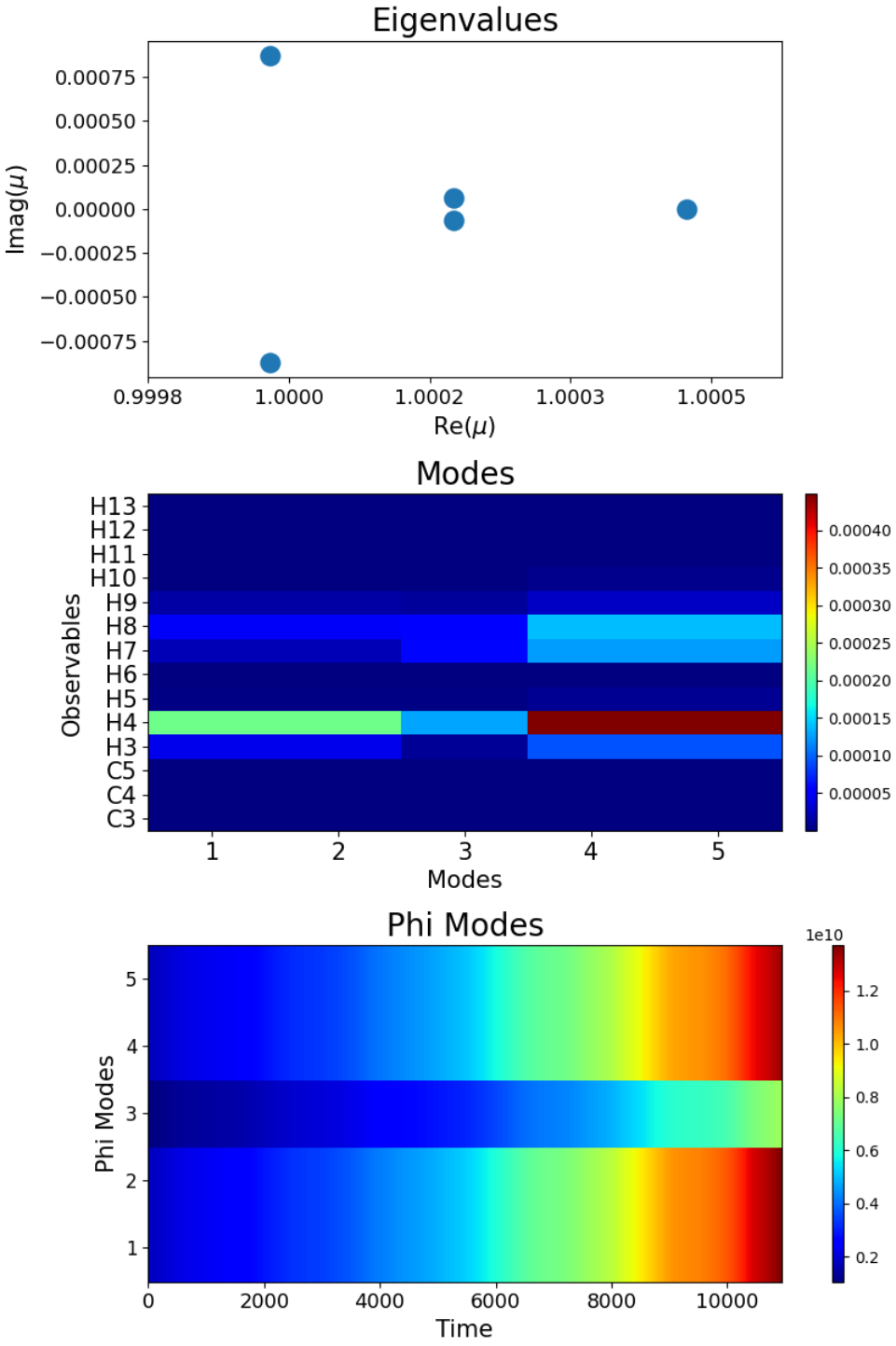

Figure 8.12. KDMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.2$, $p = 0.8$.

8.5 Thij Model with $\lambda = 0.8$, $p = 0.2$

Below in Figure 8.14, the DMD algorithm has produced a dominant mode 9. This mode makes up most of the signal upon examination of the Phi modes plot. Mode 9 shows the H_3 count grows quickly and has a strong connection to the H_4 and H_8 counts. Modes 7 and 8 show this structure as well, while modes 5 and 6 show a strong connection between the H_3, H_8, and H_9 counts. There are a few eigenvalues in the neighborhood of zero - the associated modes do not exhibit any dynamics or very small dynamics.

The KDMD algorithm shows a similar story. The KDMD modes 1 and 2, in Figure 8.15, shows a strong H_3 count and a strong connection to the H_4, H_7, H_8 and H_9 motif counts. The KDMD eigenvalues sit on the edge of the unit circle, their real parts close to one, and their imaginary parts close to zero.

In Figure 8.13, we see that the DMD modes 1, 2, 10, 11, 12, and 13 are not very good approximations to the DMD modes. DMD has, however, produced seven modes of low mode error. The KDMD algorithm produced two modes with mode errors on the order of 10^{-3}, and two modes on the order of 10^{-4}. The one-step reconstruction error for DMD is 0.99 and for KDMD the one-step reconstruction error is 0.0076. The KDMD one-step reconstruction error performs better than the DMD one-step reconstruction error, but KDMD finds less modes, all of which have low error by the Rowley measure.

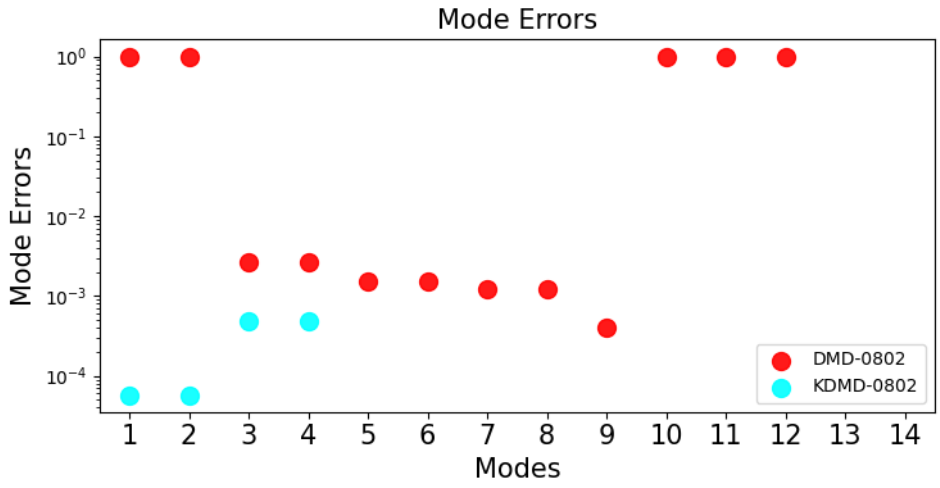

Figure 8.13. DMD and KDMD mode errors the Thij model with $\lambda = 0.8$, $p = 0.2$

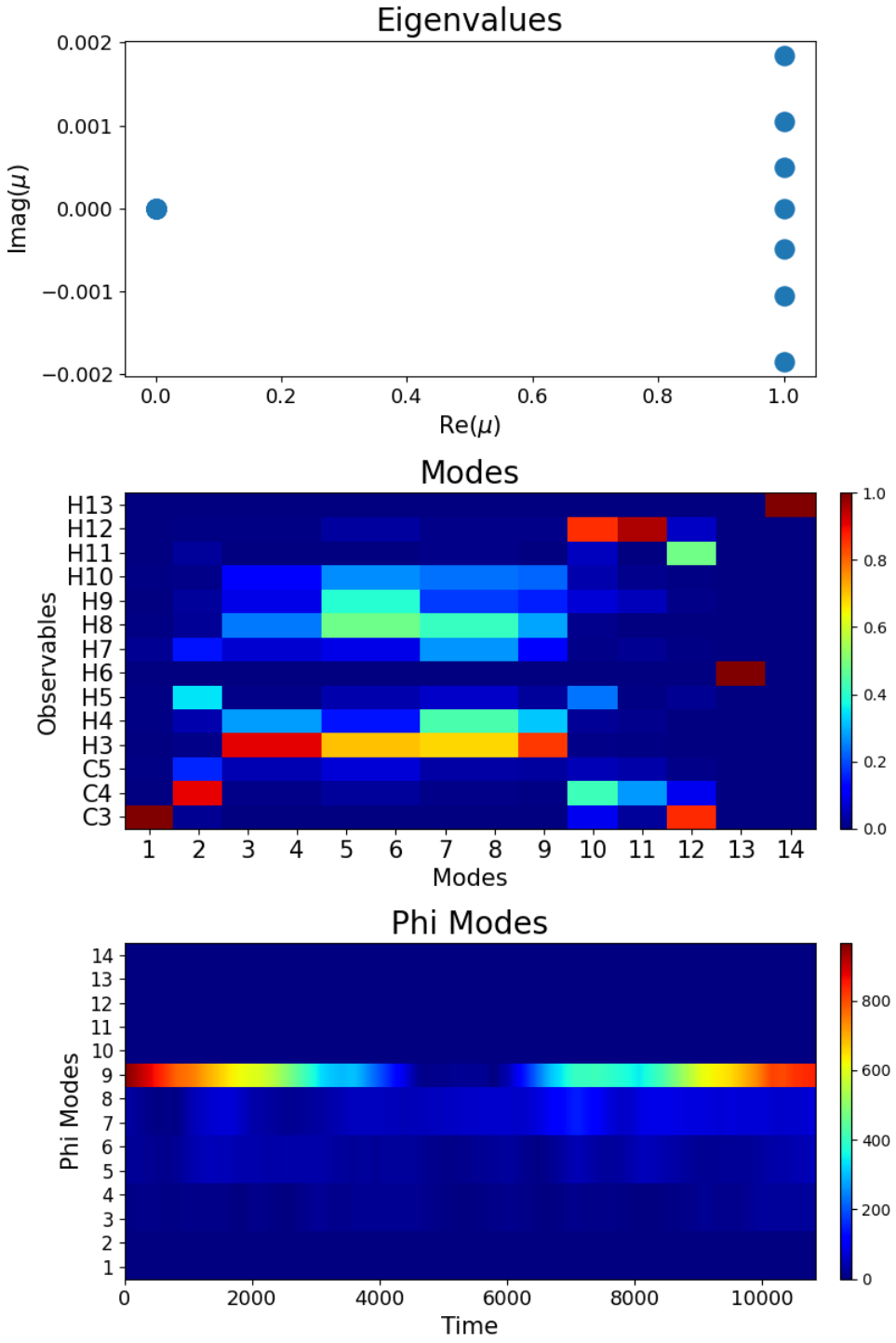

Figure 8.14. DMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.8$, $p = 0.2$.

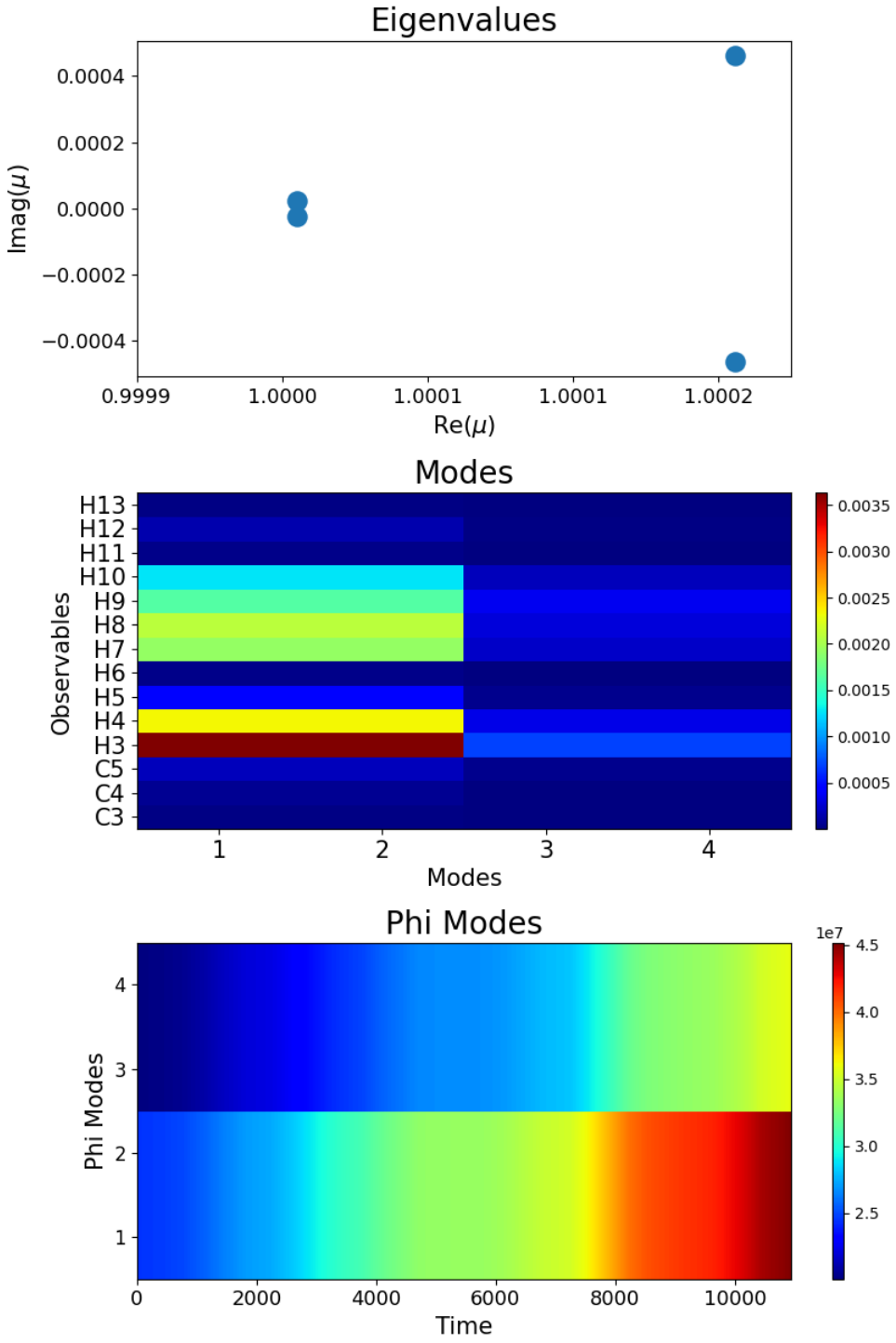

Figure 8.15. KDMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.8$, $p = 0.2$.

8.6 Thij Model with $\lambda = 0.8$, $p = 0.8$

Now examining the last Thij simulation with $\lambda = 0.8, p = 0.8$, in the DMD computation we see the results produced a single dominant mode as seen in the plot of the Phi modes in Figure 8.17. Mode 5 characterizes a large part of the dynamics and shows a strong correlation between the H_3 and H_4 counts. Modes 1 through 4 show the same connection. In Figure 8.18, the KDMD algorithm picks out the same structure in the KDMD modes 3, 4 ,5. The third KDMD mode, notably, is associated with a purely real eigenvalue and is thus associated with pure exponential growth.

In Figure 8.16, the DMD algorithm's first five modes are good approximations to trueKoopman modes. The remaining modes have errors close to one, meaning they do not behave like Koopman modes. KDMD produces a set of five modes. Four KDMD modes have a mode error on the order of 10^{-3}. A single KDMD mode has a mode error on the order of 10^{-4}. The one-step reconstruction error for DMD is 0.943, and for KDMD the one-step reconstruction error is 0.0524. DMD finds more modes, but KDMD generates low-error modes and has better one-step reconstruction accuracy.

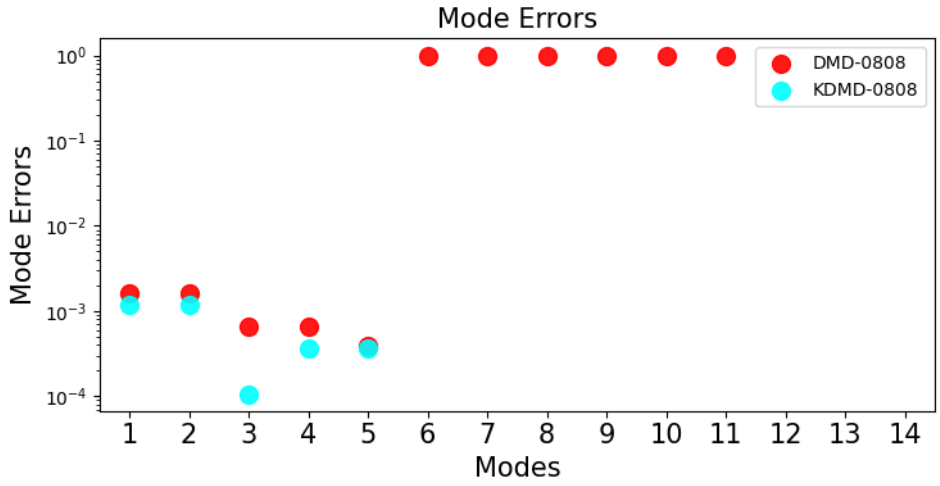

Figure 8.16. DMD and KDMD mode errors the Thij model with $\lambda = 0.8$, $p = 0.8$

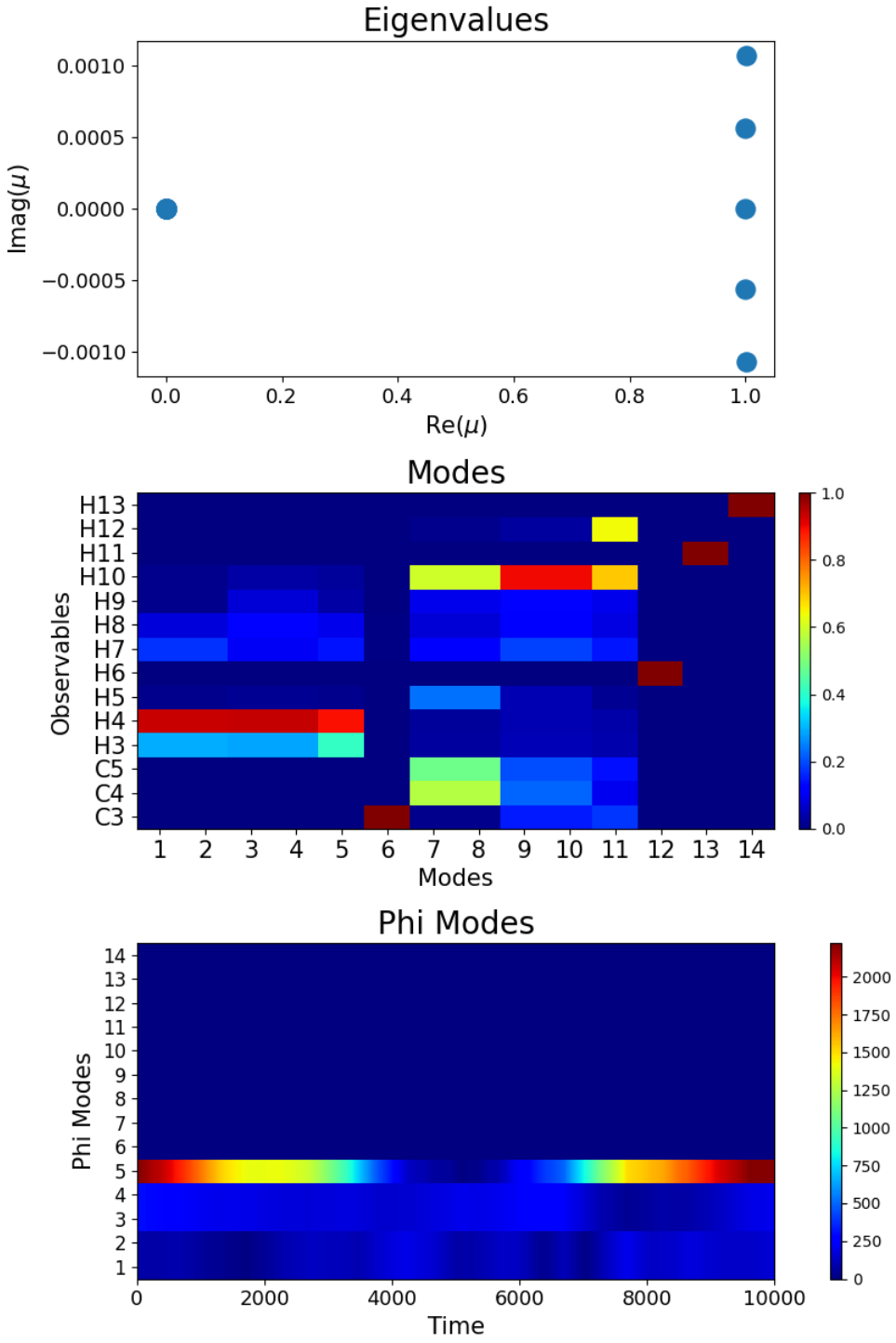

Figure 8.17. DMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.8$, $p = 0.8$.

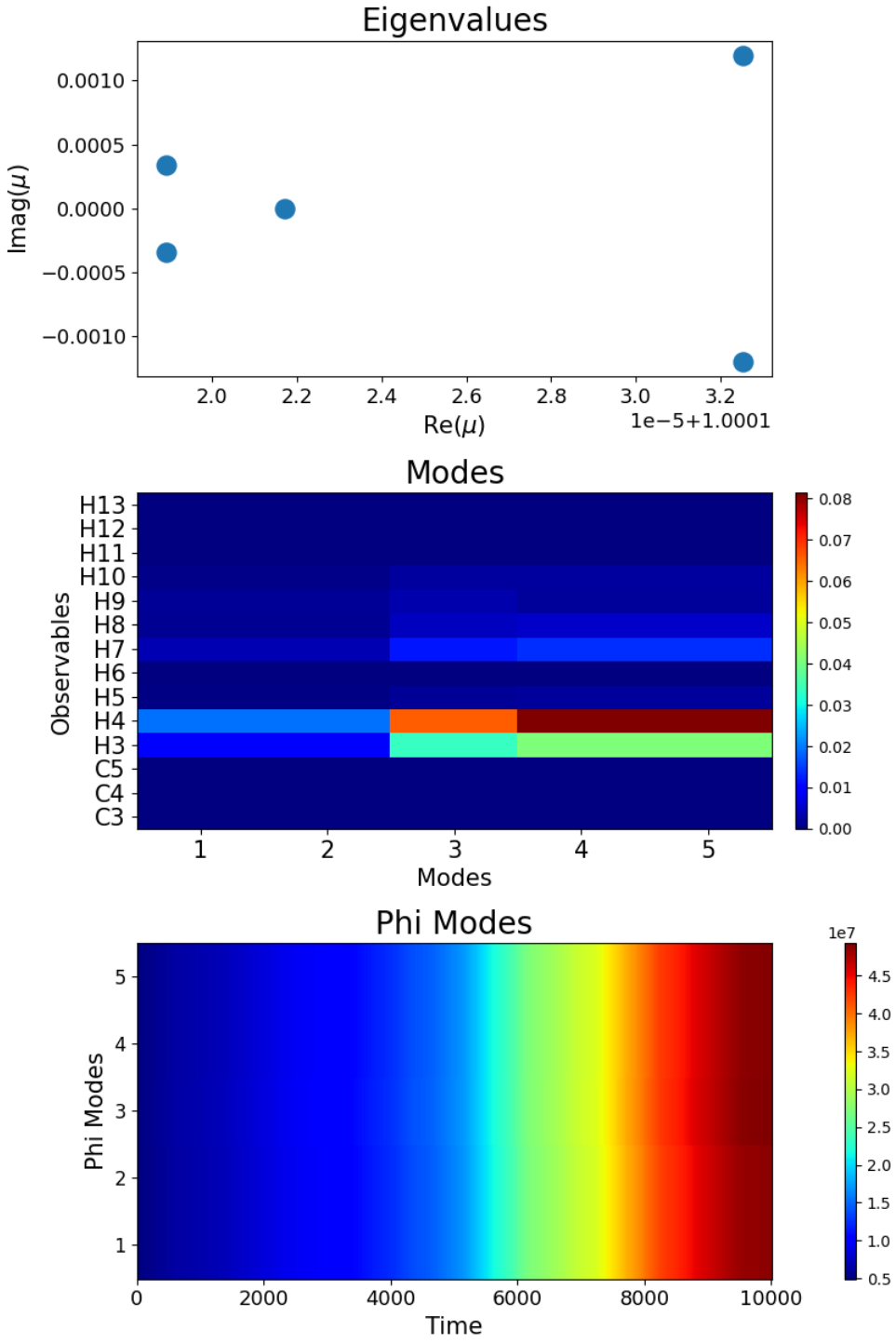

Figure 8.18. KDMD modes, eigenvalues, and phi modes for the Thij model with $\lambda = 0.8$, $p = 0.8$.

CHAPTER 9

Discussion

From the results in Chapter 8, we sift out differences in dynamic behavior by applying DMD to the motif counts. DMD and KDMD produce modes that have associated temporal patterns. For many of these simulations, the real part of the DMD eigenvalues clusters around one or zero. We only see zero eigenvalues given that a mode does not exhibit any temporal behavior such as in the $k = 1$ Barabási–Albert simulation or the $p = 0.8$ Thij simulations. The DMD and KDMD modes pick out the relationships that we see in Chapter 6. If we look to Figures 8.11 and 8.12, the DMD and KDMD modes show the H_4 count dominates the dynamics, as one would expect from Figure 4.8. For $\lambda = 0.8$, and $p = 0.2$, the dominant DMD mode shows exponential growth in the H_3 motif count. In the covariance matrix in Figure 6.5, the H_3 count variance was much higher than any other covariance or variance in the same matrix.

The DMD and KDMD modes for the Thij simulation with $\lambda = 0.2, p = 0.2$ are noticeably different. These modes reflect the strong covariances and variances of the H_7, H_8, H_9, and H_{10} counts. These motifs have subgraphs isomorphic to C_3 and C_4, which suggests a greater amount of clustering in the network. The DMD and KDMD modes are indirectly acting as measures of the prevalence of attachment mechanisms in simulations. The dominant DMD modes for this Thij simulation have structure radically different than those from other Thij simulations. The modes offer interpretations of the dynamics. These structures provide a sense of correlation and interaction between motifs. In addition, we immediately get the connection to Koopman operator theory and a way to construct a linear dynamical system approximating the temporal motif counts.

For certain simulations, we have readily available an a priori explanation of local network dynamics in the formation of induced subgraphs isomorphic to $S_k, k \gg 3$. The preferential attachment mechanism effectively acts as a positive feedback loop - the additional attachment of nodes to a node increases the likelihood of future attachment to that same node. For the Thij simulation, we have a similar preferential attachment mechanism for the message trees within a given network and the superstar mechanism for message nodes. The induced star subgraph can lead to large counts of H_4, H_7, and H_8 appearances very quickly. In Chapter 5, we determined $T2$ events can cause H_4 and

H_7 counts to grow through a combinatorial explosion, while H_8 counts grow multiplicatively.

A theory of induced star subgraphs on the network is most apt when there is a strong preferential attachment mechanism for $T2$ type events, but not so for $T3$ events. The $T3$ event opens up the possibility of adding edges between appearances of motifs. A n selection of motifs means a $\frac{n(n+1)}{2}$ number of motif joinings to consider. We briefly considered the effects of the $T3$ attachment in Chapter 5. A complete investigation would have to consider the dynamics of inter-motif and intra-motif reactions.

Examining these reactions through the covariance matrices is informative, but from the DMD and KDMD algorithms we immediately get the DMD and KDMD modes with associated dynamics. KDMD produces fewer nodes with similar SVD truncation, but overall we see better mode error and better reconstruction error from the KDMD algorithm. To improve the fit of DMD, one might use the extended dynamic mode decomposition with non-linear terms built from the motifs that we suspect interact with one another. Picking the non-linear terms judiciously could widen the span of observables, with the hope that the eigenfunctions lie within that span without making the computation intractable. In the future, a study of inner-products between modes should also provide further light on the interactions between motif appearances.

CHAPTER 10

Conclusion

The motif counts make valuable features to characterize the local structure of the network over time. The effects of the simple addition of nodes and edges can be understood through graph theory, but as the attachment mechanisms become more complex, the analysis becomes much more difficult. We have to resort to statistical tools and data-driven methods. The covariance matrices identify relationships between motifs counts. However, the Dynamic Mode Decomposition and Kernel Dynamic Mode Decomposition identify spatiotemporal coherent structures in the data. These structures, or modes, and their affiliated eigenvalues give us more than just the correlation between motifs. DMD and KDMD offer a dynamical systems perspective on motif count dynamics. The DMD algorithm is capable of producing modes of low mode error, but KDMD produces many nodes of the same magnitude of error as DMD or better. When we examine modes, we see the structure of the most dominant mode change across parameter choices of the Thij model. We are able to extract out key information about the temporal behavior of the network. DMD and KDMD are unique and effective tools to characterize the dynamics of local network structure and may lead to new methods of classifying networks by their structural dynamics.

BIBLIOGRAPHY

[1] I. Albert and R. Albert, *Conserved network motifs allow protein–protein interaction prediction*, Bioinformatics, 20 (2004), pp. 3346–3352.

[2] N. Alon, R. Yuster, and U. Zwick, *Finding and counting given length cycles*, in Algorithms — ESA '94, J. van Leeuwen, ed., Berlin, Heidelberg, 1994, Springer Berlin Heidelberg, pp. 354–364.

[3] S. Aparicio, J. Villazón-Terrazas, and G. Álvarez, *A model for scale-free networks: Application to twitter*, Entropy, 17 (2015), pp. 5848–5867.

[4] A.-L. Barabási and M. Pósfai, *Network science*, Cambridge University Press, Cambridge, 2016.

[5] A. Bessi, F. Petroni, M. Del Vicario, F. Zollo, A. Anagnostopoulos, A. Scala, G. Caldarelli, and W. Quattrociocchi, *Homophily and polarization in the age of misinformation*, The European Physical Journal Special Topics, 225 (2016).

[6] S. Bhamidi, J. M. Steele, and T. Zaman, *Twitter event networks and the superstar model*, The Annals of Applied Probability, 25 (2015).

[7] J. Bollen, H. Mao, and X. Zeng, *Twitter mood predicts the stock market*, Journal of Computational Science, 2 (2011), pp. 1–8.

[8] R. Browne, *Bitcoin spikes 20% after elon musk adds #bitcoin to his twitter bio.* "https://www.cnbc.com/2021/01/29/bitcoin-spikes-20percent-after-elon-musk-adds-bitcoin-to-his-twitter-bio.html", March 2020. [Online; accessed 15-March-2021].

[9] R. Browne, *Tweets from elon musk and other celebrities send dogecoin to a record high.* "https://www.cnbc.com/2021/02/08/tweets-from-elon-musk-and-celebrities-send-dogecoin-to-a-record-high.html", February 2020. [Online; accessed 19-March-2021].

[10] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, *Modern koopman theory for dynamical systems*, 2021.

[11] J. Bursztynsky, *Tesla shares tank after elon musk tweets the stock price is 'too high'.* "https://www.cnbc.com/2020/05/01/tesla-ceo-elon-musk-says-stock-price-is-too-high-shares-fall.html", March 2020. [Online; accessed 15-March-2021].

[12] P. Erdős and A. Rényi, *On the evolution of random graphs*, (1960), pp. 17–61.

[13] A. Jazayeri and C. C. Yang, *Motif discovery algorithms in static and temporal networks: A survey*, Journal of Complex Networks, 8 (2020). cnaa031.

[14] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor,

Dynamic Mode Decomposition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016.

[15] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, *Network motifs: Simple building blocks of complex networks*, Science, 298 (2002), pp. 824–827.

[16] M. E. J. Newman, *Networks: an introduction*, Oxford University Press, Oxford; New York, 2010.

[17] A. Paranjape, A. R. Benson, and J. Leskovec, *Motifs in temporal networks*, in Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17, New York, NY, USA, 2017, Association for Computing Machinery, p. 601–610.

[18] R. J. Prill, P. A. Iglesias, and A. Levchenko, *Dynamic properties of network motifs contribute to biological network organization*, PLOS Biology, 3 (2005), p. null.

[19] N. Przulj and I. Jurisica, *Modeling interactome: Scale-free or geometric?*, Bioinformatics (Oxford, England), 20 (2005), pp. 3508–15.

[20] Y. Ruan, A. Durresi, and L. Alfantoukh, *Using twitter trust network for stock market analysis*, Knowledge-Based Systems, 145 (2018), pp. 207–218.

[21] D. Shen, A. Urquhart, and P. Wang, *Does twitter predict bitcoin?*, Economics Letters, 174 (2019), pp. 118–122.

[22] M. ten Thij, T. Ouboter, D. Worm, N. Litvak, H. Berg, and S. Bhulai, *Modelling of trends in twitter using retweet graph dynamics*, 01 2015.

[23] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, *A kernel-based approach to data-driven koopman spectral analysis*, 2015.

[24] S. Wojcik and A. Hughes, *Sizing up twitter users*, PEW research center, 24 (2019).

[25] H. Zhang, S. T. M. Dawson, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, *Evaluating the accuracy of the dynamic mode decomposition*, 2017.