Evolving Networks: Structure and Dynamics

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

---

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

John R. Hott

August 2018

APPROVAL SHEET

This Dissertation
is submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Author Signature: 

This Dissertation has been read and approved by the examining committee:

Advisor: Worthy Martin

Committee Member: Alfred Weaver

Committee Member: Gabriel Robins

Committee Member: Luther Tychonievich

Committee Member: Jeffrey Holt

Committee Member:

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

August 2018

Abstract

Network analysis, especially social network analytics, has become widespread due to the growing amount of linked data available. Many researchers have started to consider evolving networks, i.e. Time-Varying Graphs (TVGs), to begin to understand how these networks change over time.

In this dissertation, we expand on current practice in three directions: we define a new concept of "node-identity class" to describe different "lenses" over an evolving network, we develop sampling methods to produce representative static graphs over a network as it evolves, and we utilize social network metrics to produce distributions characterizing the dynamics of the network's evolution. By combining these different techniques, we uncover a change effect in metric value due to network activity across sampling methods and window sizes, and produce a differential measure $D(\mathcal{G})$ that helps signal possibly significant network evolution.

We evaluate these techniques on synthetically-generated datasets with prescribed dynamics to show their effectiveness at capturing and depicting those events. We then apply our techniques to analyze three real-world applications: the Nauvoo Marriage Project, consisting of an evolving Mormon marital network in mid-1800s Nauvoo, IL; the Social Networks and Archival Context Project's historical social-document network; and an ArXiv co-authorship network. In each case, we were able to depict the network's dynamics, highlight periods of network activity for further investigation, and guide domain-specific researchers to new insights. For the Nauvoo Marriage Project, through a comparison of the network across identity lenses, our metrics depicted an increased centrality under the patriarchal lens compared with that of the matriarchal lens. Indeed, the rapidity with which the patriarchal centrality "rebounds" suggests a desire of the Nauvoo community to form a strong patriarchal system.

To everyone who has been an influence, support, and guide.

*To my amazing wife **Patricia**, who I met on this journey. You have been an unwavering support; thank you for the countless hours spent proofreading.*

*To my advisor **Worthy Martin**, for providing wisdom and guidance. You inspire me in connecting Computer Science to so many other disciplines across the University.*

*To **Nathan Brunelle**, **Tommy Tracy**, **Samee Zahur**, and everyone who has graciously proofread my works and gave feedback in practice presentations. I am forever grateful.*

*To **Alfred Weaver**, **Gabriel Robins**, **Luther Tychonievich**, and **Jeffrey Holt**. Thank you for your feedback, suggestions, and guidance as members of my committee.*

*To **Kathleen Flake**, for the opportunity to collaborate on such an interesting motivating application and our associated theological discussions.*

*To **Daniel Pitti** and everyone in the **SNAC Cooperative**, for the opportunity to collaborate on the SNAC project, share in its development, and examine some of its intricacies.*

*To **Sarah**, **Joy**, **Shayne**, **Doug**, **Lauren**, **Robbie**, **Cindy**, **Keswick**, and everyone at **IATH**.*

*To my parents, **John** and **Debbie Hott**, who have been a constant support and encouragement.*

Contents

List of Tables

List of Figures

Chapter 1

Introduction

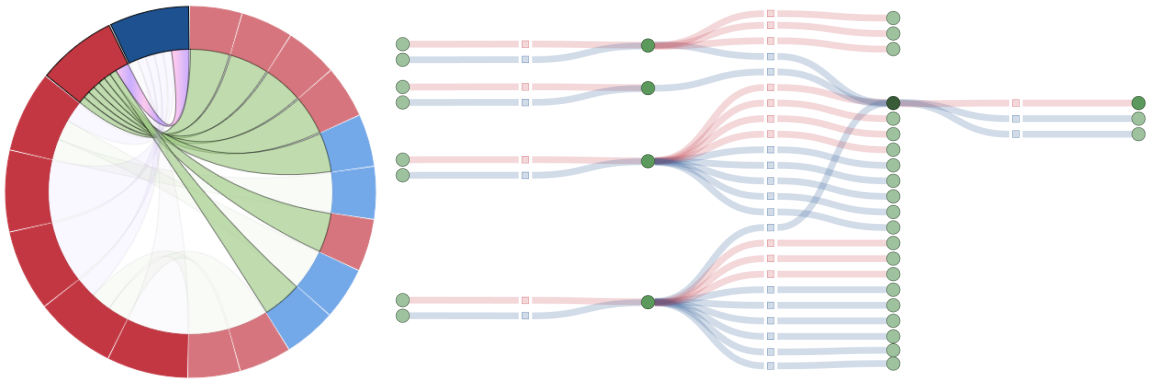
As corporations, hospitals, and research institutions try to understand and analyze the large amounts of data being gathered, Barabási [2] notes that many are realizing that their data may be represented as networks of actors and relationships. Large examples include the interconnection of web pages by links throughout the Internet, social networks such as Facebook[1] and Twitter,[2] and even doctor-patient interactions in the health sciences.

Many of the networks available have a well-defined temporal component: social graphs change as friendships come and go, emails travel through a corporate network, traffic patterns change as accidents and construction occur, and diseases spread through towns and hospitals. These examples evidence topology changes over time, such as new friends added to a social network who may then connect with their peers, as well as communications that flow through connections with a fixed topology. Prior research usually focuses on analyzing these evolving networks at evenly-spaced "snapshots," static views of the graph, either for computational or data-availability reasons [1]. However, the ways in which the evolving networks are sampled are not sufficient; we measure how graphs evolve topologically—the dynamics of their evolution. In an evolving graph, while maintaining their overall identity, the entities represented by nodes may merge, split, and change characteristics, and edges may be re-routed to alternate endpoints. Additionally, networks may have different definitions of identity conceptualized into different sets of nodes and edges that can be compared and

[1] http://www.facebook.com
[2] http://www.twitter.com

analyzed, for example institutional affiliation and individual authorship in a co-authorship network of publications.

Specifically, this project stemmed from visualizing polygamous marriages in mid-1800s Nauvoo, IL. In collaboration with Dr. Kathleen Flake, Richard L. Bushman Chair of Mormon Studies at the University of Virginia, we have defined a new highly-evolving network view of marriages. We encapsulated marriage dynamics and interrelations (intra-marriage edges) into a node, visualized as a chord diagram [3–5]. The chord diagrams depict spousal and parental relationships, with parents on the left and children on the right as seen in Figure 1.1a. These nodes evolve and change characteristics as individual participants are brought into and leave the marriage over time. We connect these nodes with directed edges (inter-marriage edges), representing the individuals participating in the marriages, from the marriage into which they were born to all marriages they enter into as an adult [3, 4, 6] as seen in Figure 5.5. This network includes temporally active and identity-changing nodes and edges, as well as various "identity classes" of nodes: pairwise binary marriages, patriarchal (husband-centric) polygamous marriages, and matriarchal (wife-centric) polygamous marriages. We provide extensions to evolving graphs that can accommodate this new identity aspect, and devise metrics to analyze and fairly compare the dynamics of multiple networks. This allowed us to address meaningful questions within the domain of Dr. Flake's project, such as how marital nodes change importance as the Mormon church expanded in Nauvoo and Utah, and how the marital lineage structure compares with the changing church hierarchy through that period.



(a) (b)

Figure 1.1: (a) Chord diagram representing a polygamous marriage with 6 wives and 11 children. Even though we consider these chord diagrams as evolving structures, we depict here all marriages and children across all time. Evolving versions may be seen at `http://nauvoo.iath.virginia.edu/viz`. (b) Sample marriage flow diagram. Nodes are matriarchal marital units, including all males married to a given woman. The darkest node depicts a polyandrous marriage, in which three men were married to the same woman.

Similarly, we also extend and apply our results to other real-world motivating applications, including

an ArXiv co-authorship graph and the Social Networks and Archival Contexts (SNAC) project. Each of these datasets provides a social network that evolves across time as more papers are published and documents are archived. They lead the researcher to ask important and detailed questions about the networks over time, such as: Which university is the most connected to others by publications at any given time? How are organizations changing their publication and archival holdings? How do the archived historical figures interact throughout SNAC's social network? Are there trends in the graphs that correlate to external forces?

Researchers need algorithms and metrics to apply over their datasets that will provide not only data points, but answers with relevant semantic meaning to their specific projects. Our aim is to define and produce these algorithms and metrics, apply them, and provide them to a variety of researchers.

> **Thesis:** *Our extensions to evolving networks and related measures, including the new concept of "node-identity function," will be effective in capturing crucial aspects of motivating applications by characterizing the structure and dynamics of those evolving networks, without and with "node-identity function" involvement.*

The effectiveness in capturing aspects of humanities-related motivating examples, specifically the Nauvoo and SNAC datasets, is measured by working directly with those researchers.

This dissertation is therefore organized as follows. We conclude this chapter with a discussion of related and prior work, ending with the formal definitions of Time Varying Graphs and three different notions of time. Our methods of analysis, including sampling methods, graph centrality measures, and dynamics measures are discussed in Chapter 2. We apply and validate those methods with synthetically-generated datasets in Chapter 3 and our real-world motivating examples in Chapter 4. Since our real-world applications led to extensions of this work, we discuss a new set of visualizations for the Nauvoo Marriage Project as an extension in Chapter 5. We conclude in Chapter 6 and discuss challenges and future work.

1.1 Related Work

Evolving networks have been defined and studied in many different forms over the last few decades [7–11]. Research in this field typically focuses on applications of evolving graphs, such as transportation networks [12], communications networks [9], ad-hoc wireless networks [13–15], and social

networks [16]. Other work has connected the temporal domain with the spatial, specifically geo-spatial networks [17].

Ferreira in 2002 [10] provides an initial definition of evolving graphs as a digraph with an ordered set of subgraphs. The ordered set is therefore the snapshot, or state of the evolving graph, at that particular point in time. He then produces a shortest path algorithm over this definition of evolving graphs. In 2004 [13], Ferreira then updates his evolving graph definition by introducing temporal paths, called "journeys," which will be adopted by Casteigts, et al [14], in their Time-Varying Graph formalism in 2012. Meanwhile, Grindrod, et al, in 2011 [11] consider the effects of time in ordered snapshots as dynamic walks on communication networks and define a temporal centrality calculation based on the effectiveness of any node to broadcast or receive messages.

Aggarwal and Subbian [8] then later summarize the state of the art in evolutionary network analysis as of 2014. They divide evolving networks into two categories: "slowly evolving networks," and "streaming networks." They argue that slowly evolving networks have little evolution and therefore may be examined offline by comparing snapshots of the network state at different times. Streaming networks, which they define as created by transient updates, include communication networks that require real-time analysis since the size of these graphs may not fit on disk. They discuss various methods for analyzing both of their types of graphs, noting the computational complexity involved in each case.

Casteigts, et al [14] provide a comprehensive summary and theoretic definition of evolving networks as Time Varying Graphs. As Aggarwal and Subbian summarized the state of the art at the time, Casteigts intended to create a formalization of evolutionary networks going forward. We will review their formalism in Section 1.2, as their work provides the foundation for our extensions. We may also note that others have begun using their definitions, such as Barjon, et al [7] in 2014. For example, Barjon uses the TVG formalism, with the network defined as an ordered set of directed snapshot graphs similar to Ferreira in 2002 [9], to devise an algorithm to test whether any vertices are connected through "ordered" journeys.

This dissertation expands on previous work in evolving networks, including that of Casteigts' Time-Varying Graphs [14]. We examine the dynamics of TVGs by utilizing new sampling methods to produce a richer representation of the TVG at each time-point, based on temporally-close changes in the network. These sampling methods allow the application of social network metrics over static

graphs to produce distributions characterizing the evolving network dynamics, which may then be compared to depict the underlying network's activity.

We also analyze evolving networks that may be transformed based on nuanced identities present in the data. Prior research does not address comparing various views of the same evolving network based on different identities present. Therefore, we define and use node-identity functions to transform an evolving network into related "views," or "classes," such as a matriarchal and patriarchal view of the Nauvoo Marriage Project's lineage network, and utilize our sampling methods and social network metrics to compare and contrast them.

1.2 Time-Varying Graphs

In 2012, Casteigts, et al, surveyed the current landscape of evolving networks and proposed a formal definition of Time-Varying Graphs (TVG) [14]. This formalism provides a framework for future work to commonly discuss and analyze time-dependent evolving networks. They define a TVG as the graph

$$\mathcal{G} = (V, E, T, \rho, \zeta, \psi, \varphi),$$

over lifetime $T \subseteq \mathbb{T}$ where

- V is the set of nodes, the representation of entities in the TVG,
- E is the set of directed, labeled edges, $E \subseteq V \times V \times L_E$, where L_E is the label alphabet,[3]
- $\rho : E \times T \to \{0, 1\}$ is the edge presence function, indicating whether a given edge is available at a specific time,
- $\zeta : E \times T \to \mathbb{T}$ is the edge latency function, indicating how long (in $\mathbb{T}$) it takes to traverse an edge given a start time,
- $\psi : V \times T \to \{0, 1\}$ is the node presence function, and
- $\varphi : V \times T \to \mathbb{T}$ is the node latency function (local processing time).

[3] For our extensions, labels are omitted.

They allow $\mathbb{T}$ to be chosen based on the desired definition of time; specifically $\mathbb{T} = \mathbb{N}$ for discrete-time systems or $\mathbb{T} = \mathbb{R}^+$ for continuous-time systems. For simplicity, let us denote G_t to be the *snapshot*, i.e., simple directed graph, of $\mathcal{G}$ at time $t \in T$, which captures the nodes and edges of $\mathcal{G}$ extant at that time point. Namely, $G_t = (V_t, E_t)$ where $V_t = \{v \in V \mid \psi(v,t) = 1\}$ and $E_t = \{e \in E \mid \rho(e,t) = 1\}$.

Path traversals under TVGs may have a temporal aspect. They may be considered to happen entirely in one G_t snapshot, or they may be traversed across time as the network evolves, i.e., traversal is constrained by the "presence" and "latency" functions of $\mathcal{G}$. The Casteigts [18] and Xuan [19] define "journeys" as paths that have a temporal component and must be traversed through time. They examine three different types of "shortest paths" with relation to journeys: foremost, fastest, and shortest. Foremost journeys are paths through time that arrive at the target node at the earliest time-point in T. Fastest journeys are paths that take the shortest amount of time from source to target. They might depart their source node later than a foremost journey and might arrive later. Shortest journeys are defined as the shortest path in the number of edge traversals, regardless of time taken.

Previous research analyzes these graphs by inspecting the snapshot view of the graph at evenly-spaced observation intervals. Ribeiro, et al, discuss this in [1], and compare the affect of interval length—the size of a Δt—between snapshots on the effectiveness of a random walker to traverse the graph via the snapshots. They found that evenly-spaced snapshots provide drastically different outcomes depending on the Δt chosen, as well as the type and dynamics of the graph. Therefore, more work is needed to compare the effects of different methods for sampling over the network, and whether metrics used on the sampled graphs adequately capture network dynamics. To begin to address time driven by events rather than Δt intervals, Butts [20] creates a relational framework for modeling social events. His model is built on an ordered series of events, in which the wall-clock time is known, but also allows for a relaxed model which requires only the ordering of events.

We extend prior work by investigating other methods to obtain a static view of the evolving graph around a time-point in $\mathbb{T}$, varying the temporal neighborhood size around that point to add additional temporal contextual information to the static view, and applying metrics to these views across time to better quantify the dynamics of the overall evolving network.

1.3 Time

Let us formally define three different notions of time necessary for discussing evolving networks and our analytic and sampling methods. We define **objective time**, **Δt-time**, and **event-driven time** as follows.

Definition 1 (Objective time)**.** *Continuous time, as in [14], such that*

$$\mathbb{T} = \mathbb{R}^+.$$

Definition 2 (Δt-time)**.** *Time measured over a regular repeating interval. Specifically,*

$$\mathcal{T} = \{t_0, t_1, ..., t_m\} \subset \mathbb{T}, \ \ s.t. \ \forall i \in [1, m], t_i = t_{i-1} + \Delta t,$$

given the initial time-point $t_0 \in \mathbb{T}$ and the interval size Δt.

Δt-time is metered in human-centric intervals, such as years, days, seconds, nanoseconds, etc, with a consistent interval size, denoted Δt. Observe that if the interval is much shorter than the network's rate of change, many time instances t_i will contain the same network without any changes. Similarly to sparse matrices, redundancy should be reduced for storage costs. Therefore, let us also define event-driven time, based on the set of events that change the state of the network.

Let $\mathcal{E} = \{\varepsilon_0, \varepsilon_1, ..., \varepsilon_n\}$ be the ordered set of state-changing events occurring in the network. Each event ε_i is defined as the set of changes applied to the TVG $\mathcal{G}$. For event sets that have associated objective time instances, we define

$$\tau : \mathcal{E} \to \mathbb{T}$$

as the one-to-one mapping from events to objective time. If multiple changes occur at the same moment in objective time, we define the set of changes as one event at that timepoint.

Definition 3 (Event-driven time)**.** *A new $t_i \in \mathbb{T}$ is assigned only when an event occurs that modifies the state the network. Specifically,*

$$T = \{t \mid \exists \varepsilon_i \in \mathcal{E} \wedge \tau(\varepsilon_i) = t\} \subseteq \mathbb{T}.$$

Event-driven time is sparse: for any two adjacent snapshots of $\mathcal{G}$, G_{t_i} and $G_{t_{i+1}}$ taken at $t_i, t_{i+1} \in T$, $G_{t_i} \neq G_{t_{i+1}}$. When events are evenly spaced in objective time, such as those events resulting from

regular metered observations of network change, $T = \mathcal{T}$ with Δt set to the observation interval. Figure 1.2 illustrates the relationships of these definitions.

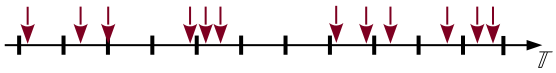


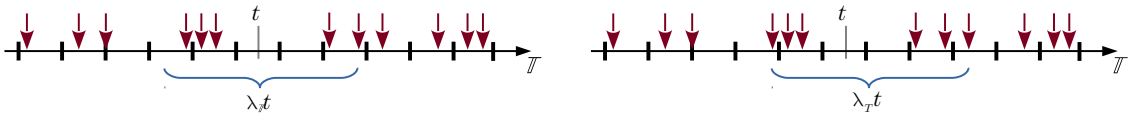
Figure 1.2: In this graph, objective time $\mathbb{T}$ is represented as the x-axis, the tick-marks denote Δt-time, $\mathcal{T}$, and the downward arrows represent events ε_i (specifically, they are event-driven time points $\tau(\varepsilon_i) = t_i \in T$ for each $\varepsilon_i \in \mathcal{E}$).

Chapter 2

Tools and Methods of Analysis

Since static network analysis tools are a well-developed and well-studied area, let us extend many of those tools to incorporate time and produce a dynamic picture of the network across its lifespan. Specifically, we apply various network centrality measures over the networks across time by utilizing different sampling methods to get pictures of the network that are both comparable and more comprehensive than simple snapshots at a time point. By comparing across time windows, sampling methods, and measures, we can get a picture of the dynamics of the network over time.

2.1 Sampling Methods



(a) (b)

Figure 2.1: (a) $\lambda_{\mathbb{T}}t$ time interval in objective time around a given point $t \in \mathbb{T}$, and (b) $\lambda_T t = 3$-event time interval in event-driven time around a given point $t \in \mathbb{T}$. Each tick mark denotes a time-point in Δt-time; the downward arrows denote the time-points of events.

Given our two different notions of time, Δt-time and event-driven time, let us define *sampling* operations that aggregate the changes of a TVG in a given λt-length time interval around any point $t \in \mathbb{T}$. Any of these operations may be performed over an interval of λt changes to the graph in

event-driven time, $\lambda_T t$, or a λt-length objective-time interval, $\lambda_{\mathbb{T}} t$. Figure 2.1 shows an example of two such intervals. As seen in the figure, choice of a λt value is important, and multiple should be considered, since any $\lambda_{\mathbb{T}} t$ may not adequately include changes to the graph and $\lambda_T t$ may include a disproportionate amount of objective time. Not only is the choice of time interval definition important, but every sampling operation of the TVG produces a semantically different version of the graph within that interval.

To define the sampling methods, let $t_a = t - \lambda t$, $t_z = t + \lambda t$, and $G(t, \lambda t) = (V(t, \lambda t), E(t, \lambda t))$ be the sampling of $\mathcal{G}$ in the $2\lambda t$-sized interval surrounding t. Then, the sampling operations we produce $G(t, \lambda t)$ are:

1. **Union**: The union of all edges and nodes extant at any point during the interval. Any node or edge existing or created within one sampling interval, even if it is subsequently removed within the interval, will be included in $G(t, \lambda t)$.[1]

$$V(t, \lambda t) = \{v \mid v \in V \wedge \exists t'(t_a \leq t' \leq t_z \wedge \psi(v, t') = 1)\}$$
$$E(t, \lambda t) = \{e \mid e \in E \wedge \exists t'(t_a \leq t' \leq t_z \wedge \rho(e, t') = 1)\}$$

   b. For event-driven measurements, V and E are defined below in terms of number events from t, where $\lambda_T t \in \mathbb{N}$. Since they are directly related to their objective time counterparts, we omit the latter definitions. Let $k = \max\{i \mid \tau(\varepsilon_i) \leq t\}$, $k_a = k - (\lambda_T t + 1)$, $k_z = k + \lambda_T t$, $t_a = \tau(\varepsilon_{k_a})$, and $t_z = \tau(\varepsilon_{k_z})$.

$$V(t, \lambda_T t) = \{v \mid v \in V \wedge \exists t'(t_a \leq t' \leq t_z \wedge \psi(v, t') = 1)\}$$
$$E(t, \lambda_T t) = \{e \mid e \in E \wedge \exists t'(t_a \leq t' \leq t_z \wedge \rho(e, t') = 1)\}$$

2. **Intersection**: Only edges and nodes that exist throughout the entirety interval.

$$V(t, \lambda t) = \{v \in V \mid \forall t' \in [t_a, t_z] \Rightarrow \psi(v, t') = 1\}$$
$$E(t, \lambda t) = \{e \in E \mid \forall t' \in [t_a, t_z] \Rightarrow \rho(e, t') = 1\}$$

[1] For $t \in \mathcal{T}$ and $\lambda t = \Delta t$, this method of sampling is equivalent to the snapshots as defined by Ribeiro in [1].

3. **Begin-End**: Only edges and nodes that exist at the beginning and end of the interval. If an edge or node is added and removed within the time interval, we ignore it. Note: if an edge that exists at the beginning of the interval is removed then re-added, it will still be included in $G(t, \lambda t)$ in this sampling process.

$$V(t, \lambda t) = \{v \in V \mid \psi(v, t_a) = 1 \wedge \psi(v, t_z) = 1\}$$
$$E(t, \lambda t) = \{e \in E \mid \rho(e, t_a) = 1 \wedge \rho(e, t_z) = 1\}$$

4. **Begin**: Only edges and nodes present at the start of the interval that exist for some non-trivial time $\epsilon > 0$.[2]

$$V(t, \lambda t) = \{v \in V \mid \exists \epsilon > 0, \forall t' \in [t_a, t_a + \epsilon] \Rightarrow \psi(v, t') = 1\}$$
$$E(t, \lambda t) = \{e \in E \mid \exists \epsilon > 0, \forall t' \in [t_a, t_a + \epsilon] \Rightarrow \rho(e, t') = 1\}$$

5. **End**: Only edges and nodes present at the end of the interval that exist for some non-trivial time $\epsilon > 0$.

$$V(t, \lambda t) = \{v \in V \mid \exists \epsilon > 0, \forall t' \in [t_z - \epsilon, t_z] \Rightarrow \psi(v, t') = 1\}$$
$$E(t, \lambda t) = \{e \in E \mid \exists \epsilon > 0, \forall t' \in [t_z - \epsilon, t_z] \Rightarrow \rho(e, t') = 1\}$$

6. **Transitive**: The transitive closure of the graph within the time interval. We consider the transitive closure of any journeys or paths that may legally be taken on the graph within the λt time interval, to ensure that all temporally-ordered paths are represented in the sampling of the graph.

For the bulk of this dissertation, we will focus solely on the first three sampling methods: Union, Intersection, and Begin-End. The Begin and End sampling methods may be calculated as transformations of the Intersection sampling by setting $\lambda t' = \epsilon/2$ and shifting the sample point by $\lambda t - \epsilon/2$ in the given direction. Specifically,

$$G_{begin}(t, \lambda t) = G_{intersection}\left(t - \lambda t - \frac{\epsilon}{2}, \frac{\epsilon}{2}\right).$$

[2] If $\epsilon = 0$ and $t \in \mathcal{T}$, this method of sampling is equivalent to snapshot sampling at regular Δt intervals.

Transitive closure, on the other hand, proved to be more challenging and "messy." When the traversal cost of a journey across any edge is 0, such as in our synthetic experiments in Chapter 3, computing a transitive closure over the network will fill out the connected components in the graph. If the graph is connected during the sampled interval, or the orderings among edge presence allow for journeys connecting all nodes, the transitive closure may produce a complete graph. Since our experiments, synthetic and real-world, are not constructed with latencies, we may only usefully learn whether the network is complete from the transitive closure sampling.

2.2 Metrics

We apply preexisting social network analysis measures to sampled graphs $G(t, \lambda t)$ of $\mathcal{G}$ to obtain a distribution of the measure over time. Conceptually, $G(t, \lambda t)$ exists for any point $t \in \mathbb{T}$, however practically we will create $G(t, \lambda t)$ and compute the measures for finite subsets of $\mathbb{T}$. Let us first define and discuss the metrics under consideration.

We consider four different measures of a graph, along with some of their derivatives; they include: harmonic centrality, betweenness centrality, degree, and diameter. The first three are all considered to be measures of centrality and their origins are discussed in the works of Freeman [21–23]. When discussing his centrality measures in connection with a communication pattern study by Bavela [24] that attempted to used those metrics to determine the structure of a network based on the centrality values, he described closeness centrality, a relative of harmonic centrality, as a node's "independence from control" or "efficiency;" betweenness centrality as a node's "potential for control;" and degree (centrality) as a node's "potential for activity in the network" [23]. He continued,

> "The three measures of overall network centrality introduced by Freeman (1979) agree on assignment of extremes. They all assign the star or wheel the maximum centrality score and the circle and the complete graph the minimum score. Between these extremes, however, agreement breaks down; they differ in their relative ranking of intermediate forms.
>
> "Centrality did emerge as an important structural variable, but not the traditional kind of centrality based on closeness. Instead, the experimentally important kinds of centrality were those based on potentials for activity and for control." [23]

In this section, we discuss and define each centrality measure in terms of a static graph $G = (V, E)$ with $N = |V|$. Since each centrality metric provides a measurement of static graph G, but sampled graphs of $\mathcal{G}$ at different points in time may vary in size, we normalize all metric values between $[0, 1]$ to allow for a fair comparison. Subsequently, since each metric provides a different depiction of network change, we combine them to produce an overall depiction.

For our notation, node-centric definitions will be denoted with a superscript asterisk, such as C_H^*, while graph-wide definitions will omit this asterisk. See Appendix B for a complete list of our notations.

2.2.1 Harmonic Centrality

Closeness centrality, as defined by Sabidussi [25] and discussed by Freeman [23] and later Wasserman and Faust [26], is the inverse of distances to all other nodes. It therefore measures how close, on average, every other node is to the examined node. Freeman in 1979 viewed this measure as an indicator of either independence from control or an indicator of efficiency. Specifically, for a graph $G = (V, E)$, it is defined for node $v \in V$ as

$$C_C^*(v) = \frac{1}{\sum_{u \in V} d(u, v)}, \tag{2.1}$$

where $d(u, v)$ is the shortest distance between the two nodes measured in number of hops or weighted hops. For graphs with disconnected nodes or disjoint subgraphs, the closeness centrality is undefined. Harmonic centrality was defined as a comparable replacement for closeness centrality. Marchiori and Latora [27], Dekker [28], Rochat [29], and Boldi and Vigna [30] each define harmonic centrality to compute a measurement with a similar property that is also defined for disconnected graphs. It is defined for node v in $G = (V, E)$ as

$$C_H^{'*}(v) = \sum_{u \in V} \frac{1}{d(u, v)}. \tag{2.2}$$

We apply Wasserman and Faust's approach of normalizing closeness centrality [26] to node-centric harmonic centrality, and therefore define it officially for $G = (V, E)$ where $N = |V|$ as

$$C_H^*(v) = \frac{1}{N - 1} \sum_{u \in V} \frac{1}{d(u, v)}, \tag{2.3}$$

with $|V| < 1 \Rightarrow C^*_H(v) = 0$.

In order to obtain a graph-wide depiction of the network, we calculate some measurements over its distribution: average, minimum, and maximum node-centric harmonic centralities and the standard deviation.

$$C^*_{mH}(G) = \min_{v \in V} C^*_H(v) \tag{2.4}$$

$$C^*_{MH}(G) = \max_{v \in V} C^*_H(v) \tag{2.5}$$

$$C^*_{AH}(G) = \frac{1}{N} \sum_{v \in V} C^*_H(v) \tag{2.6}$$

$$C^*_{\sigma H}(G) = \sqrt{\frac{1}{N} \sum_{v \in V} \left(C^*_{AH}(G) - C^*_H(v)\right)^2} \tag{2.7}$$

We also calculate a graph-centric measure of harmonic centrality as an indicator of the overall network centrality. Wasserman and Faust [26] define the graph-centric measure for closeness centrality as

$$C_C(G) = \frac{2N - 3}{(N-2)(N-1)} \sum_{v \in V} \left(\max_{u \in V} C^*_C(u) - C^*_C(v) \right),$$

while Rochat defines the graph-centric harmonic index as

$$C'_H(G) = \frac{2(N-1)}{N} \sum_{v \in V} \left(\max_{u \in V} C^*_H(u) - C^*_H(v) \right).$$

In order to normalize the metric, following the methodology of Wasserman and Faust, let us formally define graph-centric Harmonic Centrality as

$$C_H(G) = \frac{2}{N-2} \sum_{v \in V} \left(\max_{u \in V} \left(C^*_H(u)\right) - C^*_H(v) \right). \tag{2.8}$$

This graph-centric measure calculates the disparity in the network between the nodes with the highest node-centric harmonic centrality and the majority of others. If all nodes have roughly the same harmonic centrality, this measure produces a value close to 0. As a few nodes become more central than others, this measure grows to a maximum of 1. Rochat noted that his harmonic centrality index's maximum value occurred when the graph was a star with a central hub and $N - 1$ spokes, even though his method did not normalize this maximal value. Our metric follows the same pattern, however the star graph produces a maximal value of 1.

2.2.2 Betweenness Centrality

Point-wise betweenness centrality, as defined by Freeman [21, 22] and discussed in Wasserman and Faust [26], measures node importance based on how many shortest paths it lies on. Freeman [23] describes it as "an index of potential for control of communication," since a node with high betweenness centrality has the most potential to become the "actor in the middle" of the network. It is formally defined for a graph $G = (V, E)$ with $N = |V|$ as

$$C_B'^*(v) = \sum_{x \neq v \neq y \in V} \frac{\sigma_{xy}(v)}{\sigma_{xy}}, \tag{2.9}$$

where σ_{xy} is the number of shortest paths between nodes x and y and $\sigma_{xy}(v)$ is the number of shortest paths between x and y that include v. Wasserman and Faust normalize the measure to allow comparison with other centrality measures. They define the measure as

$$C_B''^*(v) = \frac{2}{(N-1)(N-2)} \sum_{x \neq v \neq y \in V} \frac{\sigma_{xy}(v)}{\sigma_{xy}}, \tag{2.10}$$

which provides normalization for undirected graphs. Since we allow graphs to be directed, let us normalize the value and define point-wise betweenness centrality as

$$C_B^*(v) = \frac{2}{(N-1)(N-2)} \sum_{x \neq v \neq y \in V} \frac{\sigma_{xy}(v)}{\sigma_{xy}}. \tag{2.11}$$

Similar to harmonic centrality measures, we calculate distribution values over the network to determine a graph-wide depiction of the network under betweenness centrality: average, minimum, and maximum node-centric betweenness centralities and the standard deviation.

$$C_{mB}^*(G) = \min_{v \in V} C_B^*(v) \tag{2.12}$$

$$C_{MB}^*(G) = \max_{v \in V} C_B^*(v) \tag{2.13}$$

$$C_{AB}^*(G) = \frac{1}{N} \sum_{v \in V} C_B^*(v) \tag{2.14}$$

$$C_{\sigma B}^*(G) = \sqrt{\frac{1}{N} \sum_{v \in V} \left(C_{AB}^*(G) - C_B^*(v)\right)^2} \tag{2.15}$$

Following Wasserman and Faust [26] and Freeman [21], let us also consider a graph-centric measure of betweenness centrality as an indicator of the entire network's betweenness. Since our definition for node-centric betweenness closely aligns with their definitions, we utilize their graph-wide measure for $G = (V, E), N = |V|$,

$$C_B(G) \quad = \quad \frac{1}{N-1} \sum_{v \in V} \left(\max_{\forall u \in V} \left(C_B^*(u) \right) - C_B^*(v) \right). \tag{2.16}$$

Freeman [23] defined this graph-wide measure as a way to determine "the degree to which a network was dominated by a single point." Like the similar measure for harmonic and closeness centralities, this measure takes its greatest value of 1 when the graph topology is a star with a central "between" hub and $N-1$ spokes that must route all paths through the hub. It takes minimal value, 0, when the nodes are all equally between; either a complete, circle, or fully-disconnected graph.

2.2.3 Degree and Density

The degree of a node, sometimes referred to as degree centrality, is a measurement of the number of adjacent nodes to an inspected node. Specifically, the degree of node $v \in V$ for graph $G = (V, E)$ is typically defined as

$$C_D^{'*}(v) = deg(v) = |E'|, \forall u \in V, (u, v) \in E \cup E^{-1} \implies (u, v) \in E'.$$

For an undirected graph, $C_D^{'*}(v)$ may take the largest value of $N-1$, in which the node is connected to all others. In the directed case, the largest value is $2N-2$, since there may be at most $N-1$ in-edges and $N-1$ out-edges.

To enable a fair comparison between degree centrality and the other calculated centralities, let us normalize it and formally define degree centrality for undirected graph $G = (V, E)$ as

$$C_D^*(v) = \frac{1}{N-1} C_D^{'*}(v) = \frac{1}{N-1} deg(v). \tag{2.17}$$

This measure then produces the percentage of maximal degree for each node rather than the report of the degree itself.

A graph-centric version of degree centrality is obtained by averaging the individual degree values across all nodes in the network. It is therefore defined as

$$C_D(G) = \frac{1}{N}\sum_{v\in V} C_D^*(v) \tag{2.18}$$

$$= \frac{1}{N(N-1)}\sum_{v\in V} deg(v). \tag{2.19}$$

It is also important to note that this calculation of $C_D(G)$ produces the density of the network. Density is defined as completeness of the graph, in terms of percentage of possible edges extant in the graph. Formally for undirected graph $G = (V, E)$ with $N = |V|$, it is

$$density(G) = \frac{2|E|}{N(N-1)} \tag{2.20}$$

$$= \frac{\sum_{v\in V} deg(v)}{N(N-1)} \tag{2.21}$$

$$= C_D(G) \tag{2.22}$$

We therefore denote both density and degree centrality of the graph as $C_D(G)$.

2.2.4 Diameter

Lastly, for completeness, let us calculate the diameter of the network, defined informally as the longest shortest path existing in the network between any two nodes. This metric produces a measure of the network span; no connection need be longer than the diameter to connect any two nodes. For graph $G = (V, E)$ it is defined as

$$dia(G) = \max_{u,v\in V}(d_{min}(u, v)), \tag{2.23}$$

where $d_{min}(u, v)$ is the shortest path between u and v. In a disconnected graph, this definition of diameter does not provide an accurate measure of the network's span.

2.3 Metric Properties

Let us first uncover some properties of these metrics that will govern the behavior we will see as we utilize them to perform our temporal analysis. For our proposed synthetic experiments, we wanted to control for certain aspects of network activity, so we defined a **constrained graph** as a graph G_C that contains at least a minimum spanning star. That is, a connected graph with at least one hub node connected directly by an edge to all other nodes. Since this constraint is restrictive, we define an **unconstrained graph** as any graph G defined without this topological constraint; e.g., it may be disconnected. We will define these terms when we discuss our synthetic construction in Section 3.1. Under certain circumstances, i.e., when an unconstrained graph is connected and has a diameter of 2 or less, metric values behave similarly to those over constrained graphs.

We begin with our constrained graph construction and prove the following properties of the centrality measures in relation to edge counts. For these properties, we relax our constraint slightly, requiring that the graph G be connected and have diameter $dia(G) \leq 2$.

Theorem 2.3.1 (Proportionality of Harmonic Centrality)**.** *For any connected graph $G = (V, E)$ with diameter ≤ 2, average harmonic centrality is proportional to edge count. $C^*_{AH}(G) \propto |E|$.*

Proof. Let $G = (V, E)$ be a connected graph with diameter ≤ 2 and $|V| = N$. Then, for any node $v \in V$ with $deg(v) = m$, we can define the node's non-normalized harmonic centrality as

$$C^{'*}_H(v) = \sum_{u \in V} \frac{1}{d(v,u)} = m + (N - m - 1)\frac{1}{2} = \left(m + \frac{N - (m+1)}{2}\right), \tag{2.24}$$

since v is directly connected to m nodes by its edges with distance 1 and all other $N - m - 1$ nodes with distance 2. It is then normalized as

$$C^*_H(v) = \frac{1}{N-1} C^{'*}_H(v) = \frac{1}{N-1}\left(m + \frac{N - (m+1)}{2}\right). \tag{2.25}$$

We note here that our definition of harmonic centrality holds, such that

$$\begin{aligned} deg(v) = N - 1 \quad &\Rightarrow \quad C^{'*}_H(v) = N - 1 \text{ and } C^*_H(v) = 1 \\ deg(v) = 1 \quad &\Rightarrow \quad C^{'*}_H(v) = 1 + \frac{N-2}{2} \text{ and } C^*_H(v) = \frac{N}{2(N-1)}. \end{aligned}$$

The average harmonic centrality of G, $C^*_{AH}(G)$, is defined as

$$C^*_{AH}(G) = \frac{1}{N}\sum_{v\in V} C^*_H(v) = \frac{1}{N(N-1)}\sum_{v\in V} C'^*_H(v).$$

Since we know the formula for $C'^*_H(v)$ from Equation 2.24, we rewrite the calculation of $C^*_{AH}(G)$ as follows: Let $k_1, k_2, ..., k_{N-1} \in \mathbb{Z}$, where $\sum_i k_i = N$. Then,

$$C^*_{AH}(G) = \frac{1}{N(N-1)} \quad \left[\quad k_1\left(1+\frac{N-2}{2}\right) + k_2\left(2+\frac{N-3}{2}\right) + ... \right. \\ \left. +k_{N-2}\left(N-2+\frac{1}{2}\right) + k_{N-1}(N-1)\right] \tag{2.26}$$

with the i-th term defined as

$$k_i\left(i+\frac{N-(i+1)}{2}\right).$$

Now, let us consider $G' = (V, E')$, with one additional edge $(x, y) \notin E$. $E' = E \cup \{(x, y)\}$. Without loss of generality, let us say that in G,

$$\begin{aligned} C^*_H(x) &= j + \frac{N-(j+1)}{2}, \text{ and} \\ C^*_H(y) &= l + \frac{N-(l+1)}{2}. \end{aligned}$$

Since the diameter of G is ≤ 2, adding edge (x, y) reduces the distance between x and y from 2 to 1, but does not affect the shortest distance between any other nodes. The new path using (x, y) may be taken as an additional shortest path of length 2, or the original path of length 2 may be followed. By definition, the distance between any nodes $d(u, v) \leq 2$. Therefore, for G',

$$\begin{aligned} C^*_H(x) &= (j+1) + \frac{N-(j+2)}{2}, \text{ and} \\ C^*_H(y) &= (l+1) + \frac{N-(l+2)}{2}. \end{aligned}$$

In the computation for $C^*_{AH}(G')$, x and y's coefficients have shifted from the derivation in Equation 2.26 such that

$$k'_j = k_j - 1, \quad k'_{j+1} = k_{j+1} + 1, \quad k'_l = k_l - 1, \text{ and } \quad k'_{l+1} = k_{l+1} + 1.$$

We then compute the difference of average harmonic centralities, $C^*_{AH}(G') - C^*_{AH}(G)$ to determine

the effect of the edge addition on average harmonic centrality. Using the formula from Equation 2.26, we note that $k'_i = k_i$ except in the instances listed above, leaving us with

$$\begin{aligned} C^*_{AH}(G') - C^*_{AH}(G) &= \frac{1}{N(N-1)}[k'_j H_j - k_j H_j + k'_{j+1} H_{j+1} - k_{j+1} H_{j+1} \\ &\quad + k'_l H_l - k_l H_l + k'_{l+1} H_{l+1} - k_{l+1} H_{l+1}] \\ &= \frac{1}{N(N-1)} [H_{j+1} - H_j + H_{l+1} - H_l,] \end{aligned}$$

where

$$H_i = \left(i + \frac{N-(i+1)}{2} \right).$$

Then,

$$\begin{aligned} C^*_{AH}(G') - C^*_{AH}(G) &= \frac{1}{N(N-1)} [H_{j+1} - H_j + H_{l+1} - H_l] \\ &= \frac{1}{N(N-1)} \left[\left((j+1) + \frac{N-j-2}{2} \right) - \left((j) + \frac{N-j-1}{2} \right) \right. \\ &\quad \left. + \left((l+1) + \frac{N-l-2}{2} \right) - \left((l) + \frac{N-l-1}{2} \right) \right] \\ &= \frac{1}{N(N-1)} \left[1 + \frac{N-j-2}{2} - \frac{N-j-1}{2} + 1 + \frac{N-l-2}{2} - \frac{N-l-1}{2} \right] \\ &= \frac{1}{N(N-1)} \left[1 + \frac{N-j-1}{2} - \frac{1}{2} - \frac{N-j-1}{2} + 1 + \frac{N-l-1}{2} - \frac{1}{2} - \frac{N-l-1}{2} \right] \\ &= \frac{1}{N(N-1)} \left[1 - \frac{1}{2} + 1 - \frac{1}{2} \right] \\ &= \frac{1}{N(N-1)}. \end{aligned}$$

Therefore,

$$C^*_{AH}(G') = C^*_{AH}(G) + \frac{1}{N(N-1)}, \tag{2.27}$$

which may be repeatedly applied for multiple transformations of the graph, i.e. edge additions. The property holds also for edge deletions if the diameter remains $dia(G) \leq 2$,

$$C^*_{AH}(G) = C^*_{AH}(G') - \frac{1}{N(N-1)}. \tag{2.28}$$

□

Corollary 2.3.2. *Given two connected graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with diameters ≤ 2 and $m = |E_2| - |E_1|$,*

$$C^*_{AH}(G_2) = C^*_{AH}(G_1) + \frac{m}{N(N-1)}. \tag{2.29}$$

Corollary 2.3.3. *Given any two connected graphs $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, with $|V_1| = |V_2|$, $|E_1| = |E_2|$, and diameters $dia(G_1) \leq 2$ and $dia(G_2) \leq 2$,*

$$C^*_{AH}(G_1) = C^*_{AH}(G_2). \tag{2.30}$$

Corollary 2.3.4. *Let $\mathcal{G}$ be a TVG, with $G_1 = (V, E(t_1))$ and $G_2 = (V, E(t_2))$ be adjacent samplings of $\mathcal{G}$, both connected graphs with diameters $dia(G_1) \leq 2$ and $dia(G_2) \leq 2$. Then the change in density of the sampled TVG across these timepoints t_1, t_2 is equal to twice the change in average harmonic centrality,*

$$\Delta C_D(\mathcal{G}) = 2\Delta C^*_{AH}(\mathcal{G}).$$

Proof. Given $\mathcal{G}$ and G_1, G_2, let $m = |E_2| - |E_1|$. Then,

$$\begin{aligned} \frac{2|E_2|}{N(N-1)} - \frac{2|E_1|}{N(N-1)} &= 2(C^*_{AH}(G_2) - C^*_{AH}(G_1)) \\ \frac{2(|E_2| - |E_1|}{N(N-1)} &= 2\left(\frac{m}{N(N-1)}\right) \\ 1 &= 1. \end{aligned}$$

□

For our other metrics in the constrained case, we make strong statements about the proportionality between the centrality measures and edge counts of our constrained graph $G_C = (V, E)$. Additional formal proofs are beyond the scope of this dissertation.

Theorem 2.3.5. *For constrained graph $G_C = (V, E)$, harmonic centrality $C_H(G)$ is inversely proportional to the number of edges.*

Proof. The graph's hub, $v \in V$, necessarily has maximum node-centric harmonic centrality, since $\forall w \neq v \in V, d(v, w) = 1$. Adding an edge $x \rightsquigarrow y$, $x \neq y \neq v$, increases the harmonic centrality of the endpoints x and y, which decreases the distance between those nodes' harmonic centrality and the

maximal harmonic centrality, $C_H^*(v,G) - C_H^*(x,G)$ and $C_H^*(v,G) - C_H^*(y,G)$. Therefore the average distance from the maximum decreases. □

Theorem 2.3.6. *For constrained graph $G_C = (V,E)$, the average betweenness centrality is inversely proportional to the number of edges.*

Proof. Let $G = (V,E)$ be a graph that contains a minimal star,

$$\exists v \in V, \forall w \neq v \in V \Rightarrow (v,w) \in E.$$

Then, a shortest path between $x, y \in V$ may either pass through hub v or another second node for $x \rightsquigarrow u \rightsquigarrow y$; or consist of a direct edge $x \rightsquigarrow y$. Hub v will therefore have the maximum betweenness value unless another hub exists, which will share maximal value.

For any added direct edge $(w,z) \in E$, the other shortest paths of length 2 will be bypassed for the direct edge, reducing the betweenness centrality of the hub and any other bypassed nodes. z may now lie on shortest paths to w, and vice versa, but they will be on no more shortest paths than hub v. Their betweenness will not increase more than the other nodes, whose betweenness dropped due to the new edge. Therefore, the average betweenness centrality over all nodes in the graph decreases. □

Corollary 2.3.7. *For constrained graph $G_C = (V,E)$, betweenness centrality is inversely proportional to the number of edges.*

Proof. Each added direct edge $w \rightsquigarrow z$ reduces the betweenness centrality of the hub, and potentially any secondary hub. The betweenness centrality of the endpoints w and z may increase; although $C_B^*(w,G) \leq C_B^*(v,G)$ and $C_B^*(z,G) \leq C_B^*(v,G)$. Therefore, the difference between the maximum betweenness centrality and other nodes decreases, decreasing the betweenness centrality of the graph. □

Relaxing our constrained graph constraint, we can reason similarly about any unconstrained graph G. Indeed, from empirical studies in Chapter 3, we find that removing the minimal spanning star constraint produces a similar outcome.

Theorem 2.3.8. *For any graph $G = (V, E)$, average harmonic centrality $C^*_{AH}(G)$ monotonically increases as the number of edges increases and monotonically decreases as the number of edges decreases.*

Proof. Since node-centric harmonic centrality is defined as

$$C^*_H(v) = \frac{1}{N-1} \sum_{u \in V} \frac{1}{d(u,v)},$$

it is determined by the length of the shortest paths in G. Adding an edge to G will not increase any shortest path's length. Likewise, deleting an edge from G will not decrease any shortest path's length. Therefore, average harmonic centrality will not decrease as edges are added nor increase as edges are removed. □

We may inversely extend the monotonicity property of average harmonic centrality to our graph-wide harmonic centrality $C_H(G)$, however it only holds when an added or deleted edge does not affect the node(s) with maximal harmonic centrality. Since $C_H(G)$ is defined as the average of differences from the maximal node-centric harmonic centrality, an edge addition may increase the harmonic centrality of the maximal node and increase the overall harmonic centrality. Likewise, an edge deletion may decrease the maximal node's harmonic centrality and decrease the overall metric's value. In cases where an edge addition is not adjacent to the node with maximal harmonic centrality, $C_H(G)$ monotonically decreases; when an edge deletion is not adjacent to the maximal node, $C_H(G)$ monotonically increases.

As we will see empirically in Chapter 3, there is likely a correlation between average betweenness centrality, betweenness centrality, and edge count. Our arguments for monotonicity based on shortest path distance do not apply, and in fact adding an edge may increase the number of shortest paths between some nodes while decreasing the number of shortest paths between others. Like $C_H(G)$, since C_B is also defined as the average of differences from the maximal node-centric betweenness centrality, any changes to the network that affect the number of shortest paths through the maximal node(s) may break monotonicity.

Our discussion of metric properties has not included degree centrality $C_D(G)$. In Equation 2.22 we proved that degree centrality is equal to the density of the graph and therefore proportional to the number of edges.

2.4 Identity Functions

As an expansion to TVGs and evolving networks, we note that there may be various definitions of node-identity within a given graph. For example, in the Nauvoo dataset of polygamous marriages, a marital unit may be defined as a binary relationship between individuals, the set of all individuals married to the same head, or the set of all individuals related by marriage or blood. Similarly, in a co-authorship graph, an author may be identified by the person, department, or institution which produced the work. This simple expansion opens the door to cross-identity metric comparison and examination of the graph at various levels of granularity of identity.

Let us therefore define *node-identity classes*, $V[i]$, which are transformations of the nodes in a TVG based on a *node-identity function*

$$f : I \times V \to \mathcal{P}(V),$$

where I is the set of possible identities. For example, in the Nauvoo dataset, given the binary marital units as nodes, one may apply a patriarchal identity function, producing $V_{patriarchal}$, to collapse those nodes to their patriarchal counterparts by combining the marriages with the same husband into one node. These new patriarchal nodes will define one node-identity class. A similar matriarchal identity function, producing $V_{matriarchal}$, collapses nodes around the women and their marriages. These two identity classes, although related, are not duals, but may provide useful insight when comparing their dynamics. The identity classes are not strict equivalence classes, since for example, considering an institutional identity function in the ArXiv co-citation graph, an individual author may be affiliated with multiple institutions.

We define the Time/Identity-Varying Graph transformation function Ω which, given a TVG $\mathcal{G}$, the identity function f, and an identity $i \in I$, produces the related TVG $\mathcal{G}_i$ of $\mathcal{G}$ under identity i. $\mathcal{G}_i$ is called, therefore, a Time/Identity-Varying Graph of $\mathcal{G}$. Specifically,

$$\mathcal{G}_i = (V[i], E[i], T, \rho[i], \zeta[i], \psi[i], \varphi[i]) = \Omega(\mathcal{G}, i, f),$$

where

- $V[i] = f(i, V)$, the set of nodes after applying the node-identity transform function f with identity $i \in I$ to the set of nodes V in the TVG $\mathcal{G}$,

- $E[i] = \{(u, v) | \exists u' \in u, v' \in v \text{ s.t. } (u', v') \in E\}$, the set of edges after applying the node-identity transform function f with identity $i \in I$ to the nodes V in the TVG $\mathcal{G}$. $E[i] \subseteq V[i] \times V[i] \times \mathcal{P}(L_E)$.

- $\rho[i](e, t) = \{\exists u' \in u, v' \in v \text{ s.t. } \rho((u', v'), t)\}$, such that an edge exists in $\mathcal{G}_i$ if any edge mapped to it from $\mathcal{G}$ exists,

- $\psi[i](v, t) = \{\exists v' \in v \text{ s.t. } \psi(v', t)\}$, such that a node exists if any node mapped to it from $\mathcal{G}$ exists.

It is important to note that in some applications, a node's identity may change and evolve over time under the identity function. This phenomenon is most evident in corporate mergers and splits, in which a corporation may change fundamental characteristics, such as ownership, in an identity shift. Consider the simple example of American circuses, as depicted in Figure 2.2. If we consider corporate identity as the identity function, the node defining "Cooper and Bailey" will map to "Barnum and Bailey" starting in 1881, "Barnum and Bailey owned by Ringling Bros" in 1907, and "Ringling Bros and Barnum & Bailey" in 1919. Each of these nodes across time define the same circus, but with different identity characteristics, as the original circus was bought, merged, and split.

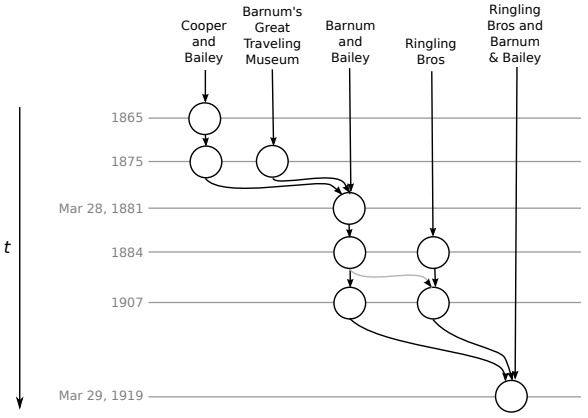


Figure 2.2: Sample set of circuses, which evolve both in name and ownership over time. In this example, the entire state of the network is shown (horizontal lines) as it evolves over time $t \in \mathbb{T}$. Vertical arrows link common identities across time.

Each of the metrics defined in Section 2.2 can therefore be used to analyze and compare the TIVGs under each defined fixed-lens identity function. By computing metrics on each variant TIVG, competing or affirming depictions of the dynamics of the overall TVG should emerge. Comparing these depictions should lead to a better understanding of the evolving network and its application.

2.5 Application to Evolving Networks

Now let us turn to applying the techniques and static-graph metrics to evolving networks, $\mathcal{G}$. First, let $G(t, \lambda t) = (V(t, \lambda t), E(t, \lambda t))$ be the sampling of $\mathcal{G}$ in the λt interval around t using one of the methods in Section 2.1. Then, applying them to the centrality metrics in Section 2.2, we arrive at sampling-based formulations of the centrality over TVG $\mathcal{G}$ in the λt-intervals around any given t as

$$C_H(\mathcal{G}, t, \lambda t) = \frac{2}{|V(t, \lambda t)| - 2} \sum_{v \in V(t, \lambda t)} \left(\max_{u \in V(t, \lambda t)} (C_H^*(u)) - C_H^*(v) \right), \tag{2.31}$$

$$C_B(\mathcal{G}, t, \lambda t) = \frac{1}{|V(t, \lambda t)| - 1} \sum_{v \in V(t, \lambda t)} \left(\max_{\forall u \in V(t, \lambda t)} (C_B^*(u)) - C_B^*(v) \right), \text{ and} \tag{2.32}$$

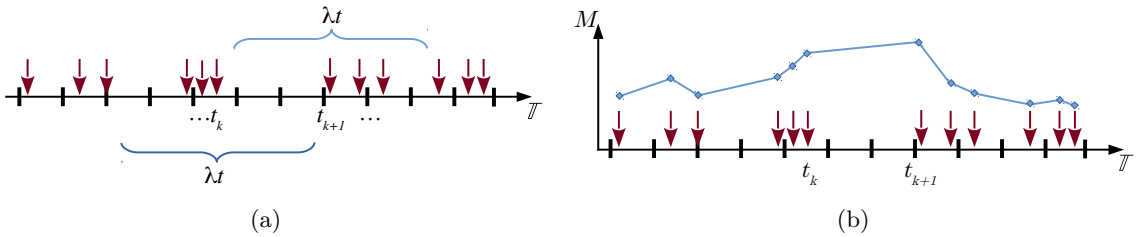
$$C_D(\mathcal{G}, t, \lambda t) = \frac{1}{|V(t, \lambda t)|(|V(t, \lambda t)| - 1)} \sum_{v \in V(t, \lambda t)} deg(v), \tag{2.33}$$

where each of the point-wise metrics are defined as before, substituting the sampled graph $G(t, \lambda t) = (V(t, \lambda t), E(t, \lambda t))$ for the internal computations of distance, degree, and shortest paths. The sampling method chosen dictates what paths are possible over the given λt time interval, which are evidenced by comparing between samplings.

By computing these measures for all $t \in T$ using a specific sampling method and λt interval, this approach produces a distribution of the measures across time for our graph $\mathcal{G}$, such as the example in Figure 2.3. After computation of this distribution, we describe two different methods for determining the dynamics of the underlying TVG: metric change due to sampling size and metric derivatives. Different λt values over the same TVG, as seen in Figure 2.4, as well as different sampling methods produce distributions at differing granularities that may then be compared to evidence metric changes indicative of change in the underlying network. Comparatively, calculating the rate of change in each metric's distribution pattern provides an additional depiction of the network's change over time. Let us examine both of these approaches for their applicability and usefulness.

2.5.1 Metric Change Due to Sampling Size as a Measure of Change

An effect in which sampling across larger λt intervals produces an increase or decrease in the metric distribution is one strong indicator of network change. Two factors in analysis impact the effect of sampling on the observed network: sampling method and λt window size. As the sampling window



(a) (b)

Figure 2.3: (a) Evolving metrics computed within a $\lambda_{\mathbb{T}}t$ interval around each graph event at times t_k, (b) which may produce a distribution over time of the metric M, similar to the one shown here.

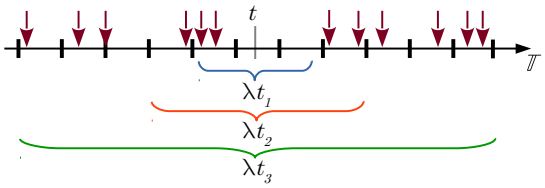


Figure 2.4: Varying the size of λt around each given point $t \in \mathbb{T}$ may capture differing degrees of the dynamics of the graph. In this timeline, λt_3 would sample all events in the network.

increases, a larger portion of the underlying evolving network is factored into the sampling method's application and ultimately dictates the sampled graph. For the additive sampling methods, Union and Transitive Closure, the additional edges and nodes present as λt increases will be included in the produced $G(t, \lambda t)$ and therefore increase potential for higher centrality measures. For the other sampling methods, only edges that persist for a predetermined amount of the window are factored into the sampled graph, reducing the potential for centrality in the sampled network. By comparing the magnitude of change between interval sizes along with the metric values, we may be able to uncover some dynamics of the network.

For TVGs with nodes that persist through their lifespan or whose nodes are relatively stable across time, this approach evidences the amount of change in the network. In this case, a highly active network is one whose edges have significant turnover. Under the Union sampling, more edges will be included in the sampled network at each point, artificially increasing the metric value for larger sampling windows compared with the distribution over snapshots. Similarly, under the Intersection of Begin-End samplings, we will see the opposite effect. Let us discuss this further in Section 3.5 when we examine the dynamics of our synthetic datasets designed to showcase these effects.

This approach, however, provides limited success in highlighting the dynamics of networks which are very sparse or in which the nodes incur a significant amount of turnover or change. When events in the network add disconnected nodes or disconnected components, even without removing any nodes or edges, including these events in an additive sampling of the network reduces the overall centrality.

Therefore, we cannot make a determined link between increased sampling window and additive sampling methods, Union and Transitive, to an increased potential for centrality; the additional disconnected components in the larger sampled graph may mask increased dynamics in the rest of the system. Likewise, the inverse applies to the other sampling methods, Intersection and Begin-End, which may remove the disconnected components from a larger sampling and therefore increase the overall centrality.

We note that this approach of determining dynamics produces a granular view of the network change, depending on the λt interval size chosen. Even though it proves effective in our synthetic results, our real-world datasets do not all evidence a persistent node phenomenon. Specifically, since real-world datasets may have significant node turnover, this approach does not work in every case. We therefore turn to a second method for uncovering the dynamics that works in the case of sparse and highly volatile networks: rate of change in centrality.

2.5.2 Metric Derivative as Measure of Change

For our second approach, we note that by definition our metrics, C_H, C_B, and C_D, give us three different depictions of the state within the network. Measuring the change in metric values may evidence different notions of network activity. The latter, C_D, provides a measure of density and will fluctuate with completeness of the graph. It will give us an indication if the edge count increases or decreases as a proportion of the size of the network. It is important to note that since we have also shown a proportionality relationship between the average centralities, C^*_{AH} and C^*_{AB}, and the density, we need not include those measures into an overall dynamic index measure. The other graph-wide centralities, C_H and C_B, each provide a rough indicator of graph topology, denoting the disparity between the most central node and the rest of the network. Combined with C_D, they provide an indicator of network change in addition to edge turnover. We therefore define an overall dynamic index $D(\mathcal{G})$ over a TVG $\mathcal{G}$ as

$$D(\mathcal{G}, \lambda t) = \frac{\sqrt{\left(\Delta C_H(\mathcal{G}, \lambda t)\right)^2 + \left(\Delta C_B(\mathcal{G}, \lambda t)\right)^2 + \left(\Delta C_D(\mathcal{G}, \lambda t)\right)^2}}{(t_i - t_{i-1})/\min(\Delta t)}, \tag{2.34}$$

where

$$\begin{aligned}
\Delta C_H(\mathcal{G}, \lambda t) &= C_H(G(t_i, \lambda t)) - C_H(G(t_{i-1}, \lambda t)), \\
\Delta C_B(\mathcal{G}, \lambda t) &= C_B(G(t_i, \lambda t)) - C_B(G(t_{i-1}, \lambda t)), \text{ and} \\
\Delta C_D(\mathcal{G}, \lambda t) &= C_D(G(t_i, \lambda t)) - C_D(G(t_{i-1}, \lambda t)).
\end{aligned}$$

Since this approach produces a dynamic index over over time, we apply it to the finest-grained sampling of $\mathcal{G}$ to produce the most precise characterization possible. When available, we apply it to TVGs analyzed using event-driven time with the smallest possible λt time window to evidence the exact points of change in the dynamical system. Therefore, let us simplify the notation in the case where $\lambda t = 0$ as

$$D(\mathcal{G}) = \frac{\sqrt{\left(\Delta C_H(\mathcal{G}, 0)\right)^2 + \left(\Delta C_B(\mathcal{G}, 0)\right)^2 + \left(\Delta C_D(\mathcal{G}, 0)\right)^2}}{(t_i - t_{i-1}) / \min(\Delta t)}. \tag{2.35}$$

Changes in graph-wide harmonic and betweenness centralities indicate changes in the topology of the network, but do not provide an indication to the growing or shrinking nature of the system. Density, or average degree centrality, provides us with an indication of edge count change. Equation 2.34, the L^2 norm of the combined values, then accounts for the topology changes in combination with the completeness changes. As we will see in Chapter 3, a jittering effect in the network is evidenced in the centrality measures even when the density is constant.

Therefore, computing $D(\mathcal{G})$ will result in a dynamic profile, highlighting the dynamic timepoints over $\mathcal{G}$'s lifespan. The resulting profile must be examined to determine a threshold by with to separate the highly dynamic points from the "noise" in the graph. This threshold depends on the overall activity of the network and a trade-off the researcher must make to limit the inspection timepoints. It must also be examined for sustained periods of change, even if they fall below this threshold, as they may provide an additional indication of significant change. For each use of this method, we present the dynamic profile as a plot and choose a threshold to remove most of the timepoints based on human intuition.

2.5.3 Takeaways

We have discussed two approaches for measuring periods of network change: comparing metric distributions across sampling window sizes and computing $D(\mathcal{G})$.

Comparing metric distributions across different samplings provides a good picture of the highly dynamic timepoints to be examined. Changes in the metric distributions under different sampling sizes are evidence of network change. However, the converse is not true; periods of no metric change between different sampling sizes is not evidence of the absence of network change. Any node presence fluctuation across samplings may offset the observed metric change.

Computing the metric derivative measure, $D(\mathcal{G})$, provides a dynamics profile of the network from which we can observe periods of significant and sustained change. The largest local maxima in $D(\mathcal{G})$ denote periods of significant change, while periods of sustained change are evidenced by time windows for which $D(\mathcal{G}) \neq 0$. This approach uncovers periods of network change even when the first approach does not.

We therefore recommend an application of our approaches based on desired reporting of the analysis:

- If a granular dynamic profile with false negatives is acceptable, or event-driven time analysis is too costly, compute centralities $C_H(\mathcal{G}), C_B(\mathcal{G})$, and $C_D(\mathcal{G})$ over multiple λt sampling windows and compare.
- Otherwise, compute $D(\mathcal{G})$ over event-driven time or the finest Δt possible. Utilize centrality components $\Delta C_H, \Delta C_B$, and ΔC_D to determine the type of change in the network at each dynamic point.

2.6 Implementation

To perform our analysis, we implemented our techniques above into a cross-platform tool, *TVGAnalyze*, written in Java. The foundation of the tool is a new data model to store TVGs. There are many possible design choices when defining a data structure to store TVGs. Our implementation utilizes an adjacency list for the nodes and edges of the network to facilitate the storing of large, potentially

sparse, networks. We also store the presence functions according to the definition of the TVG: a function stored with each graph element. To efficiently store presence, we use a tree map to store presence changes as a result of the node- and edge-presence functions ordered by timepoint. Given a smallest discernible window, such as Δt for Δt-time, we store only the points in time where the element changes its presence. By defining presence as a timeline of change-points, we can therefore determine the presence at any point in time by examining the closest previous change-point. This scheme may be easily extended to store weight or latency rather than just presence by either adding a parameter to the timeline or replacing the presence boolean with another important metric; the only requirement being a specific value denoting "not present." The full implementation of the data model is available in Appendix E.

The sampling methods were implemented as an extension to this TVG data model. For sampling methods that only require filtering the temporal edges and nodes present in the TVG, including all our sampling methods except Transitive, we add a `getPresenceOver()` extension to each node and edge that enacts a given sampling method definition over a subset of the element's timeline to determine if it exists during that window. If it returns `true`, that element can safely be added to the static graph as a result of sampling over the interval. Sampling methods that require generating edges that do not exist in the TVG, such as our Transitive sampling, necessitate creating a new graph rather than running a filter over the TVG. This led to a secondary implementation which traverses the TVG separately.

After implementing the data model and sampling methods, we are left with computing analysis over static graphs. This analysis is performed in parallel using a shared memory model, through a multi-threaded design, since each sampling is an independent representation of the TVG at a specific timepoint. Each thread shares the read-only version of the full TVG and extracts the independent sampled graph as a GraphStream [31] graph representation and performs the analysis over that sample. GraphStream is a Java framework for storing graphs and includes algorithms for traditional graph measures. It was initially built as a method for storing and analyzing dynamic graphs, but a full TVG representation has not been completed[3]. GraphStream provides a model for dynamically creating graphs with methods to add and remove edges and nodes, but computes its analysis over the static graph as it exists at the current point. We therefore use it as a representation of the static

[3]An evolving network implementation began in 2016, after we started this work, however it was abandoned shortly thereafter. Their code is currently available at https://github.com/graphstream/gs-temporal-networks. We expect this dissertation and implementation to supersede their work and supplement the GraphStream implementation to include TVGs.

graph to make use of their implementations of Dijkstra's algorithm [32] for computing shortest paths and Brandes' algorithm [33] for node-centric betweenness centrality.

This implementation, while flexible for cross-platform usage, did provide some challenges. First, by using the Java language and Java Collections framework, we found some storage limitations of the language: it uses a 32-bit integer to index arrays, even in 64-bit mode, which limits the storage capacity of all Java Collections data structures. We were therefore unable to efficiently store and access large, dense graphs with thousands of nodes, for analysis. Our implementation therefore works for networks of less than 2^{32} nodes, edges, or timepoints.

2.6.1 Time Complexity

Given a TVG $\mathcal{G} = (V, E, T, \rho, \zeta, \psi, \varphi)$, we can compute the time complexity of each of our algorithms. For our implementation, we use a TreeMap[4] to store the presence functions for each element in the network, guaranteeing a $O(\log |T|)$ computation of presence. Therefore, we can compute the sampled graph for any point $t \in T$ in $O((|V| + |E|) \log |T|)$.

To compute betweenness centrality, we make use of Brandes' algorithm, with complexity $O(|V||E|)$, resulting in an overall computational complexity for betweenness over $\mathcal{G}$ as

$$O(|T||V|^2|E| \log |T|).$$

Our implementation of harmonic centrality over $\mathcal{G}$ utilizes Dijkstra's algorithm, with complexity $O(|V| \log |V| + |E|)$, to compute shortest paths. Therefore, we have an overall complexity of

$$O(|T|(|V||E| + |V|^2 \log |V| + (|V| + |E|) \log |T|)).$$

In each case, $|T|$ is defined by either the number of events, $|\mathcal{E}|$, or the number of sample points in the network, $|\mathcal{T}|$.

[4] `https://docs.oracle.com/javase/8/docs/api/java/util/TreeMap.html`

Chapter 3

Synthetic Experiments

We conducted a two-pronged experimental exercise in order to test our approach's extensions and metrics sensitivity, scalability, and robustness across three spectrums: scalability in time and number of events, scalability in network size, and sensitivity in detecting important dynamics within the networks. In this chapter, we will discuss our synthetic experimental design, analyze our results in connection to our proposed experiments, and draw conclusions we may then apply to real-world networks. In the next chapter, we'll apply our approaches to three real-world datasets to determine their effectiveness in real conditions.

3.1 Experimental Design

Since centrality is dependent on the topology of the network, for our proposed experimental design we wanted to apply our approaches under a fixed minimal topology to control for certain aspects of network activity. To achieve this construction, we create a TVG from the **constrained network** of Chapter 2 which contains a minimal spanning star over its entire lifespan. The constrained network produces a baseline for the comparison of centrality measures across time where. As we have seen, with a star graph the central hub has the highest harmonic centrality, $C_H^*(v) = 1$, taking $d(x, y)$ to be the number of edges on the shortest path from x to y in G, compared to all other nodes with $C_H^*(u) = \frac{|V|}{2(|V|-1)}$. As more edges are added, the network becomes increasingly connected until it is complete (one clique) and every node has $C_H^*(u) = 1$.

The initial constrained network construction was too restrictive, since it did not approximate real-world graphs. After determining the effectiveness on the constrained networks, we removed the minimal star graph constraint and repeated all synthetic experiments in this chapter for **unconstrained networks** as defined in Chapter 2. The unconstrained networks in our synthetic experiments are constructed identically to their constrained counterparts, however the $|V| - 1$ edges of the star graph are not fixed. We will see similar behavior under both constructions in our synthetic experiments, until the volatility in the unconstrained networks cause the TVGs to break the assumptions of our metric properties in Section 2.3.

To create a synthetic TVG, let us begin with an initial TVG with a single time point. If this TVG follows the constrained network construction, we initialize it to contain a star graph at that time. This TVG is then extended to a full lifespan by systematically creating a series of events, with each event extending the TVG by one time unit. Each of the experiments below will then analyze sets of such randomized TVGs.

To more fully specify this experimental design, let us define our parameters:

- Γ, the construction assumption: *constrained* or *unconstrained*;
- $G_s = (V, E_s)$, the star-graph of G;
- $G_c = (V, E_c)$, the complete graph (clique) of G;
- $N = |V|$, the number of vertices present in the graph;
- γ_0, the initial density, or "completeness" factor of the TVG, $0 \leq \gamma_0 \leq 1$, with $\gamma_0 = 0 \Rightarrow$ only those edges required for Γ construction, and $\gamma_0 = 1 \Rightarrow$ a complete graph, G_c;
- $\mu(t)$, the "change over time" factor, with

$$\mu(t) = 1 \Rightarrow \text{no change},$$
$$\mu(t) > 1 \Rightarrow \text{graph grows in terms of edges},$$
$$\mu(t) < 1 \Rightarrow \text{graph shrinks in terms of edges; and}$$

- η, the degree of "jitter," the magnitude of edge activity within a given overall edge-set size.

We allow $\mu(t)$ to change over time to produce a network with changing density and topology. "Jitter," or the amount of activity in each event is set through parameter η. On an add/delete plane, μ defines the angle of the test vector while η defines the magnitude, as seen in Figure 3.1. If $\mu(t) = 1$, we anticipate that γ will fluctuate around the value of γ_0 throughout $\mathbb{T}$. However, for $\mu(t) > 1$, the graph should approach a complete graph, and likewise a star for $\mu(t) < 1$. $\mu(t)$ therefore is the velocity of γ.

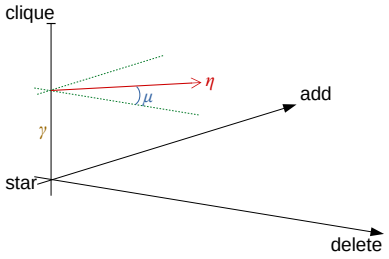


Figure 3.1: Fixed values for γ_0 and η, with $\mu = 1$, result in defining a vector in the add/delete plane. As η increases while μ remains constant, the jitter in the temporal network increases.

For simplicity of the synthetic data, let us define the event set $\mathcal{E}$ over $T = \mathcal{T} = \{1, 2, ..., t_{end}\}$ such that $\tau(\varepsilon_i) = i \in T$ and $\Delta t = 1$. In this definition, time is discretized and each timepoint contains one event. The size of each event is determined by $\mu(t)$ and η.

Algorithm 1 prescribes the overall experimental setup. For each parameter set, we will generate a TVG with the given properties and analyze it using our *TVGAnalyze* tool for a range of λt sampling sizes. In order to generate the graphs, we define generation algorithm *GenerateGraph* in Algorithm 2. We begin by creating a set of N vertices and an initial set of edges. If the TVG is constrained, $Initialize\_Edges$ first adds $N-1$ fixed edges to the set, defining the minimal star, which are inaccessible to the later $Uniform\_Select$ function. $Initialize\_Edges$ then uniformly randomly selects edges to add to the edge set until density γ_0 is reached. The TVG $\mathcal{G}$ is then built from this vertex and edge set. The vertices are set to be present for the entirety of the lifespan of $\mathcal{G}$. If the network is constrained, the constrained $N-1$ edges are also set as present for the entirety of $\mathcal{G}$'s lifespan. All other edges are set to begin presence at time $t = 0$. The generator then steps through time determining edges to be added and deleted at time t_i based on our parameters. $Select\_Add$ combines μ and η to determine the number of edges to add based on the following formula,

$$add\_size \leftarrow \lfloor \mu \cdot \eta \rfloor.$$

Algorithm 1 Synthetic Dataset Experimental Design

function *Run_Experiment*
  for $\Gamma \in \{constrained, unconstrained\}$ **do**
    for all $N \in Vertex\_Size\_Set$ **do**
      for all $t_{end} \in Lifespan\_Set$ **do**
        for all $\gamma_0 \in Initial\_Completeness\_Set$ **do**
          for all $\eta \in Jitter\_Magnitude\_Set$ **do**
            for all $\mu \in Change\_Set$ **do**
              $\mathcal{G} \leftarrow GenerateGraph(\Gamma, N, t_{end}, \gamma_0, \eta, \mu)$
              for all $\lambda t \in Sampling\_Size$ **do**
                $AnalyzeGraph(\mathcal{G}, \lambda t)$

Algorithm 2 Initial Synthetic Graph Generation

function $GenerateGraph(\Gamma, N, t_{end}, \gamma_0, \eta, \mu)$
  $V \leftarrow Initialize\_Nodes(N)$
  $E_0 \leftarrow Initialize\_Edges(V, \gamma_0, \Gamma)$
  $\mathcal{G} \leftarrow Initialize\_TVG(V, E_0, \Gamma)$
  for $t = 1...t_{end}$ **do**
    $add\_size \leftarrow Select\_Add(\mu(t), \eta)$
    $delete\_size \leftarrow Select\_Delete(\mu(t), \eta)$
    $E_t^+ \leftarrow Uniform\_Select((E_c \setminus E_{t-1}), add\_size, \Gamma)$
    $E_t^- \leftarrow Uniform\_Select((E_{t-1} \cup E_t^+), delete\_size, \Gamma)$
    $E_t \leftarrow (E_{t-1} \cup E_t^+) \setminus E_t^-$
    $\varepsilon_t = \{E_t^+, E_t^-\}$
    $\mathcal{G} \leftarrow Augment\_TVG(\mathcal{G}, \varepsilon_t, t)$
  return $\mathcal{G}$

Similarly, *Select_Delete* computes the number of edges to remove as

$$delete\_size \leftarrow \left\lfloor \frac{1}{\mu} \cdot \eta \right\rfloor.$$

With these formulas, if $\mu > 1$, the network will grow more dense; likewise, if $\mu < 1$, the network will become less dense. Once the amount of change is determined, $Uniform\_Select$ uniform randomly chooses *add_size* edges out of the absent edges at t_{i-1} to be added at this timepoint t_i. If the number of absent edges is less than *add_size*, all absent edges are added. Then $Uniform\_Select$ is used to choose *delete_size* edges to be removed from the present edges at t_{i-1} plus any newly added edges, ignoring any edges that are fixed due to the constraint Γ. *GenerateGraph* then applies those changes as one event to the TVG $\mathcal{G}$ by setting the presence functions for any newly created or deleted edges. If an edge is both created and deleted in one t_i, it is not considered as part of the event.

3.2 Experiments

The following synthetic experiments were performed. Each experiment follows the pseudo-code in Algorithms 1 and 2 to create the synthetic TVGs, exercising all combinations of parameters defined for that experiment. The generated TVGs from each experiment were then analyzed using each of our metrics and sampling methods for $\lambda t = \pm 0...5$ time-steps in $\mathcal{T}$, defined by $Sampling\_Size = \{0, 1, 2, 3, 4, 5\}$, for each $t \in T$ noting that $\lambda_{\mathbb{T}} t = \lambda_T t$.

Experiment 1 (Baseline)**.** The dataset generated from this experiment should produce a baseline for comparison to other datasets, since it includes evenly-spaced initial densities and little change over time. We hypothesize that the metrics C_H and C_B should remain relatively constant for the minimally changing graph with $\eta = 1$.

$$\begin{aligned} Vertex\_Size\_Set &= \{6, 10, 50, 100, 500\} \\ Lifespan\_Set &= \{100, 1000\} \\ Initial\_Completeness\_Set &= \{0, 0.25, 0.5, 0.75, 1\} \\ Jitter\_Magnitude\_Set &= \{1\} \\ Change\_Set &= \{1\} \end{aligned}$$

Experiment 2 (Stable Jitter Sensitivity)**.** Building on the baseline experiment, the datasets generated from this experiment test the sensitivity of the metrics C_H and C_B to jitter on the network over time. We choose multiple magnitudes of jitter based on the vertex count. We hypothesize that the metric values will fluctuate more with increased jitter for smaller λt values, but that the best-fit line will be a constant around γ_0.

$$\begin{aligned} Vertex\_Size\_Set &= \{6, 10, 50, 100, 500\} \\ Lifespan\_Set &= \{100, 1000\} \\ Initial\_Completeness\_Set &= \{0, 0.25, 0.5, 0.75, 1\} \\ Jitter\_Magnitude\_Set &= \{\lfloor N/2 \rfloor, N, 2N, 3N, \lfloor N^2/4 \rfloor, \lfloor N^2/2 \rfloor, N^2 - N + 1\} \\ Change\_Set &= \{1\} \end{aligned}$$

Experiment 3 (Constant Change)**.** This experiment's datasets test the metrics' sensitivity to changing graph density. With only one gained or removed edge per time point, we create networks

that tend toward a star and others tending to a complete graph. We anticipate that the metrics over each generated dataset will track γ as $\mu(t)$ influences the state of the network.

$$\begin{aligned} Vertex\_Size\_Set &= \{6, 10, 50, 100, 500\} \\ Lifespan\_Set &= \{100, 1000\} \\ Initial\_Completeness\_Set &= \{0, 0.25, 0.5, 0.75, 1\} \\ Jitter\_Magnitude\_Set &= \{1\} \\ Change\_Set &= \{1/2, 2\} \end{aligned}$$

Experiment 4 (Jitter Change)**.** In this experiment, datasets mirror those of Experiment 3, but include the amounts of jitter as generated in Experiment 2 to test that the metric is sensitive to changing density even with network fluctuations. We hypothesize that the metrics will track the changing values of γ, even with the increased level of jitter.

$$\begin{aligned} Vertex\_Size\_Set &= \{6, 10, 50, 100, 500\} \\ Lifespan\_Set &= \{100, 1000\} \\ Initial\_Completeness\_Set &= \{0, 0.25, 0.5, 0.75, 1\} \\ Jitter\_Magnitude\_Set &= \{\lfloor N/2 \rfloor, N, 2N, 3N, \lfloor N^2/4 \rfloor, \lfloor N^2/2 \rfloor, N^2 - N + 1\} \\ Change\_Set &= \{1/2, 2\} \end{aligned}$$

The next few experiments test the sensitivity of the metric to $\mu(t)$ as a function of time to mimic known social network phenomena. Hendrickson [34] stated that when measuring a specific hashtag's tweet volume, trends emerge. An expected real-world occurrence, such as a hurricane, has both a slow buildup and decay in hashtag tweet volume. However, an unexpected occurrence, such as an earthquake, creates a quick exponential buildup and a slow decay. To simulate increased activity in the graph, let us build $\mu(t)$ piece-wise with sin functions to affect smooth changes in γ over time.

Experiment 5 (Expected Occurrence)**.** To simulate an expected occurrence, we produce an equal increase and decrease in γ over the lifetime $|T|$. We hypothesize that in all cases, the metrics will be able to track the change exhibited in the generated dataset. However, in analysis of these datasets, we anticipate that the Begin and End sampling methods will shift the peak in the metrics by λt since

the edges of the sampling window will experience the peak at different times.

$$
\begin{aligned}
Vertex\_Size\_Set &= \{6, 10, 50, 100, 500\} \\
Lifespan\_Set &= \{100, 1000\} \\
Initial\_Completeness\_Set &= \{0, 0.25, 0.5, 0.75, 1\} \\
Jitter\_Magnitude\_Set &= \{1, \lfloor N/2 \rfloor, N, 2N, 3N, \lfloor N^2/4 \rfloor, \lfloor N^2/2 \rfloor, N^2 - N + 1\} \\
Change\_Set &= \mu(t) = \begin{cases} \sin(\frac{2\pi}{|T|}t) + 1 & t \leq t_{end}/2 \\ \sin(\frac{2\pi}{|T|}t)/2 + 1 & t > t_{end}/2 \end{cases}
\end{aligned}
$$

Experiment 6 (Unexpected Occurrence)**.** An unexpected occurrence will be modeled with a faster rise in γ until $|T|/2$ followed by a slower descent, such that $\gamma_{t_{end}} > \gamma_0$. We expect similar results to Experiment 5.

$$
\begin{aligned}
Vertex\_Size\_Set &= \{6, 10, 50, 100, 500\} \\
Lifespan\_Set &= \{100, 1000\} \\
Initial\_Completeness\_Set &= \{0, 0.25, 0.5, 0.75, 1\} \\
Jitter\_Magnitude\_Set &= \{1, \lfloor N/2 \rfloor, N, 2N, 3N, \lfloor N^2/4 \rfloor, \lfloor N^2/2 \rfloor, N^2 - N + 1\} \\
Change\_Set &= \mu(t) = \begin{cases} \sin(3\frac{2\pi}{|T|}t) + 1 & t \leq t_{end}/2 \\ \sin(\frac{2\pi}{|T|}t)/2 + 1 & t > t_{end}/2 \end{cases}
\end{aligned}
$$

3.3 Modified Synthetic Analysis

The previous experiments, as proposed, combined the effects of μ and η on the synthetically-generated TVG; namely, *Select_Add* and *Select_Delete* computed $\mu \cdot \eta$ and η/μ respectively. In that case their effects are not separably examinable. Therefore, let us create a secondary set of experiments based on applying μ and η separately with a new algorithm, *GenerateGraphSeparate*.

In this synthetic TVG generator, *Select_Add*() and *Select_Delete*() are defined solely in terms of μ: either μ or $1/\mu$. To affect μ's change on the network at time t_i, $Uniform\_Select$ is used as in Algorithm 2 to uniformly select $\mu(t)$ edges to add from those absent in t_{i-1}, followed by $1/\mu(t)$ edges to remove from the present edges in t_{i-1} or the newly created edges. After the edge changes based on μ, the jitter based on η is subsequently applied to the set of present and newly created edges:

Algorithm 3 Modified Synthetic Graph Generation

function $GenerateGraphSeparate(N, t_{end}, \gamma_0, \eta, \mu, \Gamma)$
  $V \leftarrow Initialize\_Nodes(N)$
  $E_0 \leftarrow Initialize\_Edges(V, \gamma_0, \Gamma)$
  $\mathcal{G} \leftarrow Initialize\_TVG(V, E_0)$
  for $t = 1...t_{end}$ **do**
    $add\_size \leftarrow Select\_Add(\mu(t))$
    $delete\_size \leftarrow Select\_Delete(\mu(t))$
    $E_t^+ \leftarrow Uniform\_Select((E_c \setminus E_{t-1}), add\_size, \Gamma)$
    $E_t^- \leftarrow Uniform\_Select((E_{t-1} \cup E_t^+), delete\_size, \Gamma)$
    $muE_t \leftarrow (E_{t-1} \cup E_t^+) \setminus E_t^-$
    $jitterE_t^+ \leftarrow Uniform\_Select((E_c \setminus muE_t), \eta, \Gamma)$
    $jitterE_t^- \leftarrow Uniform\_Select((muE_t \cup jitterE_t^+), \eta, \Gamma)$
    $E_t \leftarrow (muE_t \cup jitterE_t^+) \setminus jitterE_t^-$
    $\varepsilon_t = \{E_t^+, E_t^-, jitterE_t^+, jitterE_t^-\}$
    $\mathcal{G} \leftarrow Augment\_TVG(\mathcal{G}, \varepsilon_t, t)$
  return $\mathcal{G}$

η edges are added and η edges are removed by an additional use of $Uniform\_Select$. Once both operations have been calculated, the final change in edges is applied to the network, ignoring any edges that were both created and deleted in the same timestep. Due to these changes, both μ and η must individually be defined in terms of N to produce similar results to those of other synthetic experiments.

Using this new generator, we performed the following modified experiments to observe the metrics' sensitivity to graph change, μ, independently and in spite of a larger jitter η.

Experiment 7 (Jitter Change)**.** In this experiment, μ is set to a constant increase or decrease as a proportion of the number of nodes. We use various levels of jitter η in an attempt to test the metrics' ability to track change even with network fluctuations. The metrics still track the changing values of graph density and edge counts, even with the increased level of jitter.

$$\begin{aligned} Vertex\_Size\_Set &= \{10, 100\} \\ Lifespan\_Set &= \{100\} \\ Initial\_Completeness\_Set &= \{0.25, 0.5, 0.75\} \\ Jitter\_Magnitude\_Set &= \{1, N, 2N, 3N\} \\ Change\_Set &= \{N, \lfloor N/2 \rfloor, \lfloor N/4 \rfloor, \lfloor 1/N \rfloor, \lfloor 2/N \rfloor\} \end{aligned}$$

Experiment 8 (Expected Occurrence with Jitter)**.** To simulate an expected occurrence, we produce an equal increase and decrease in γ over the lifetime $|T|$. We hypothesize that in all cases, the metrics

will be able to track the change exhibited in the generated dataset. However, in analysis of these datasets, we anticipate that the Begin and End sampling methods will shift the peak in the metrics by λt.

$$
\begin{aligned}
Vertex\_Size\_Set &= \{10, 100\} \\
Lifespan\_Set &= \{100\} \\
Initial\_Completeness\_Set &= \{0.25, 0.5, 0.75\} \\
Jitter\_Magnitude\_Set &= \{1, N, 2N, 3N\} \\
Change\_Set &= \mu(t) = \begin{cases} (\sin(\frac{2\pi}{|T|}t) + 1)N & t \leq t_{end}/2 \\ (\sin(\frac{2\pi}{|T|}t)/2 + 1)N & t > t_{end}/2, \end{cases} \\
&\quad\ \mu(t) = \begin{cases} (\sin(\frac{2\pi}{|T|}t) + 1)\frac{N}{4} & t \leq t_{end}/2 \\ (\sin(\frac{2\pi}{|T|}t)/2 + 1)\frac{N}{4} & t > t_{end}/2 \end{cases}
\end{aligned}
$$

Experiment 9 (Unexpected Occurrence with Jitter)**.** An unexpected occurrence will be modeled with a faster rise in γ until $|T|/2$ followed by a slower descent, such that $\gamma_{t_{end}} > \gamma_0$. We expect a similar metric behavior to Experiment 8.

$$
\begin{aligned}
Vertex\_Size\_Set &= \{10, 100\} \\
Lifespan\_Set &= \{100\} \\
Initial\_Completeness\_Set &= \{0.25, 0.5, 0.75\} \\
Jitter\_Magnitude\_Set &= \{1, N, 2N, 3N\} \\
Change\_Set &= \mu(t) = \begin{cases} (\sin(3\frac{2\pi}{|T|}t) + 1)N & t \leq t_{end}/2 \\ (\sin(\frac{2\pi}{|T|}t)/2 + 1)N & t > t_{end}/2, \end{cases} \\
&\quad\ \mu(t) = \begin{cases} (\sin(3\frac{2\pi}{|T|}t) + 1)\frac{N}{4} & t \leq t_{end}/2 \\ (\sin(\frac{2\pi}{|T|}t)/2 + 1)\frac{N}{4} & t > t_{end}/2 \end{cases}
\end{aligned}
$$

3.4 Results

Our synthetic experimental design produced an parameter space of $159,552$ possible combinations of values, each exercised with multiple random seeds. Let us, therefore, examine slices through this

Table 3.1: Baseline metric values each of the γ_0 initial densities. Metrics remain constant across time under both centrality definitions, but evenly spaced for harmonic centrality and an exponential decline for betweenness centrality.

γ_0	C_H	C_B
0	1	1
0.25	0.75	0.11545
0.5	0.5	0.01555
0.75	0.25	0.00205
1	0	4.9×10^{-324}

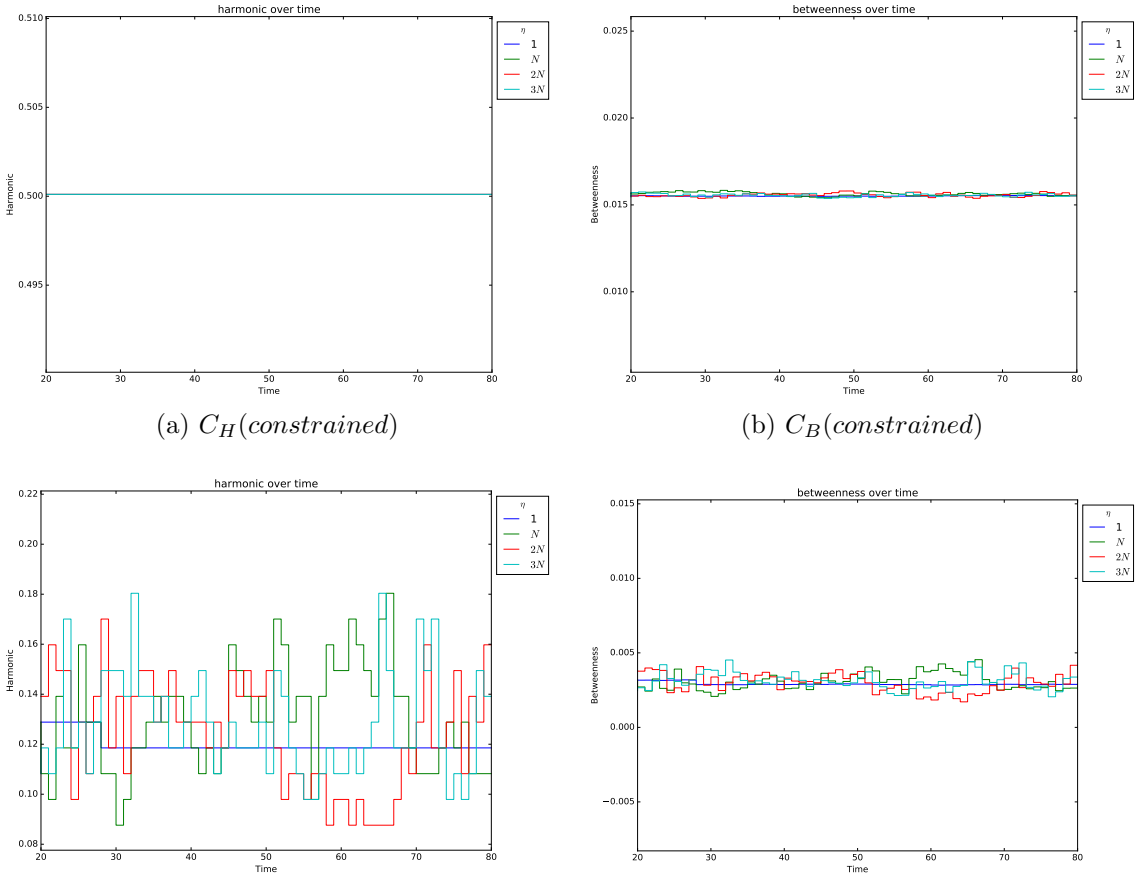
experimental space that are most helpful in verifying the usefulness of our methods and answering the experiments we set to find. We will address each experiment in order.

Experiment 1. For the baseline experiment, our results produce the intended effect: the centrality metrics are constant for a minimally changing network. This result holds for all combinations of parameters. Table 3.1 shows an example baseline measure for both C_H and C_B over a network of 500 nodes for each of the γ_0 values with $\lambda t = 0$. The properties of graph-wide centrality measures defined in Section 2.2 are observed, since the baseline $\gamma_0 = 0$ star graph produces a harmonic centrality of 1 and for each higher γ_0 value the harmonic centrality is closer 0. The change in density produces an exponential decline in betweenness centrality as density increases.

Experiment 2. For our second experiment, we increase the jitter over the network. Figure 3.2 shows the results comparing a few changes in η as the network maintains its density at $\gamma_0 = 0.5$. Under the constrained network construction, our hypothesis does not hold for C_H due to the metric properties we proved in Chapter 2 since C_H remains constant even with jitter. On the other hand, C_B evidences a small amount of change; even as η increases, the measure fluctuates around the baseline value when $\eta = 1$, supporting our hypothesis.

It is important to note here that we do not plot the results for larger η jitter values. Based on our construction in Algorithms 2 and 3, jitter is applied by adding η followed by deleting η edges. Therefore, our TVG will never be a complete graph, since the delete η step happens second. For example, let $N = 10$, which produces a graph with at most 45 edges total. Based on our construction, if $\eta = 3N = 30$, we will add and subtract 30 edges resulting in a maximum of 15 edges present at each timepoint. This will skew our synthetic results and therefore we make a trade-off and limit our analysis to smaller η that evidence the effects of jitter without hiding the effects due to network size.

Experiment 3. Figures 3.3 and 3.4 show two examples of a constrained network with 100 nodes



(a) $C_H(constrained)$ (b) $C_B(constrained)$

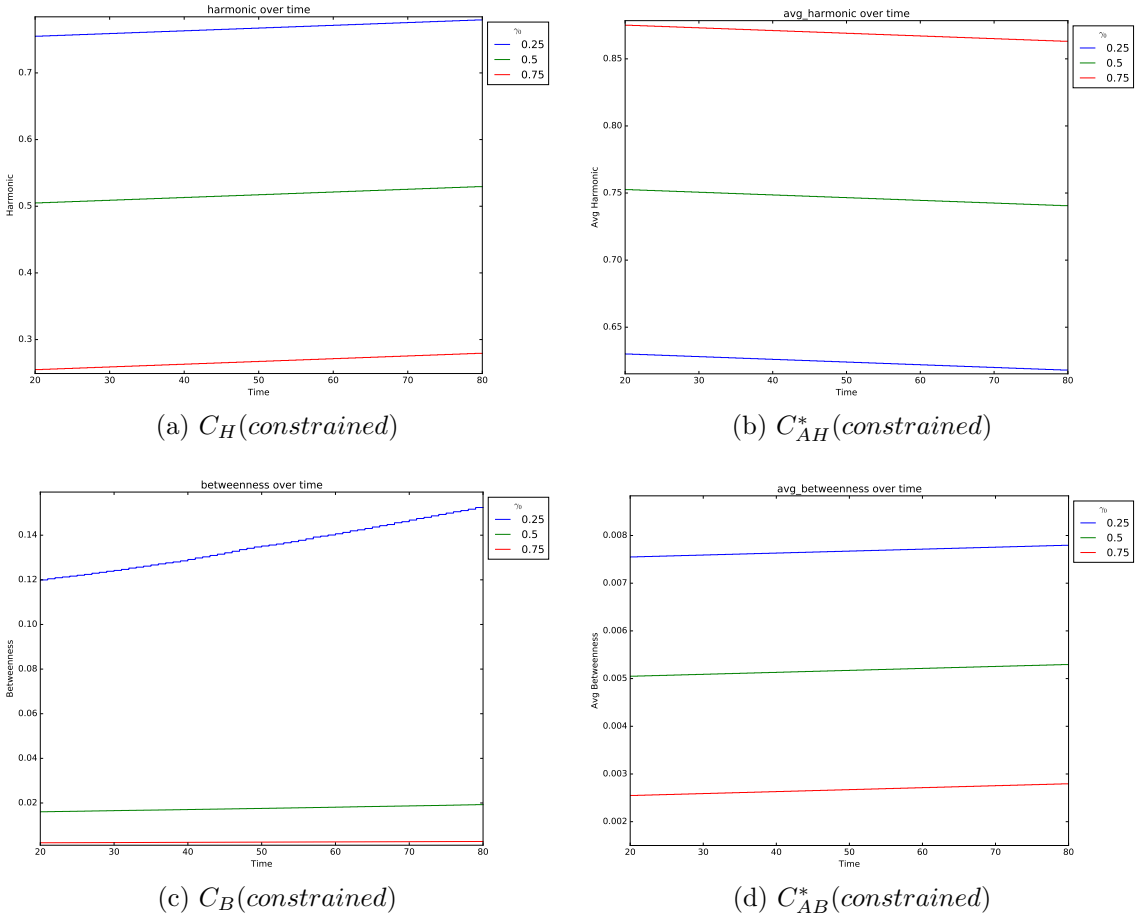
(c) $C_H(unconstrained)$ (d) $C_B(unconstrained)$

Figure 3.2: Comparing jitter over TVGs with 100 nodes. As jitter increases, our metrics prove to be stable. For the constrained construction, jitter does not affect C_H.

under both $\mu(t)$ functions starting at different γ_0 densities. In each case, one edge is added or removed in each timestep based on the value of μ. Average harmonic centrality in Figure 3.3b and Figure 3.4b is proportional to the changing density over time, validating our hypothesis. Based on our additional investigation on metric properties in Chapter 2, we confirm our hypothesis with the inverse proportionality relationship of average betweenness and harmonic centrality. Figure 3.5 extends the lifespan of the TVG in Figure 3.3, displaying changes in metric values for 1000 timesteps.

The results are not as clean when we look at the unconstrained network; the properties that hold for connected graphs with minimal stars do not hold for networks without the guaranteed minimal topology. Figures 3.6 and 3.7 show similar results for a 100-node unconstrained network. Our hypothesis holds for the average centrality metrics, following our metric properties in Chapter 2. Since the maximal node centrality is not guaranteed for unconstrained networks, as it is for the hub in constrained networks, C_H and C_B track the maximal node's centrality.

For both the constrained and unconstrained networks, the betweenness centrality metric exemplifies an inverse proportionality to the change in density of the network, however its magnitude is dependent on overall network density. Specifically, for a denser network, such as $\gamma_0 = 0.75$, or 75% complete, we see a less pronounced increase in betweenness compared with a network that is only 25% complete, $\gamma_0 = 0.25$.

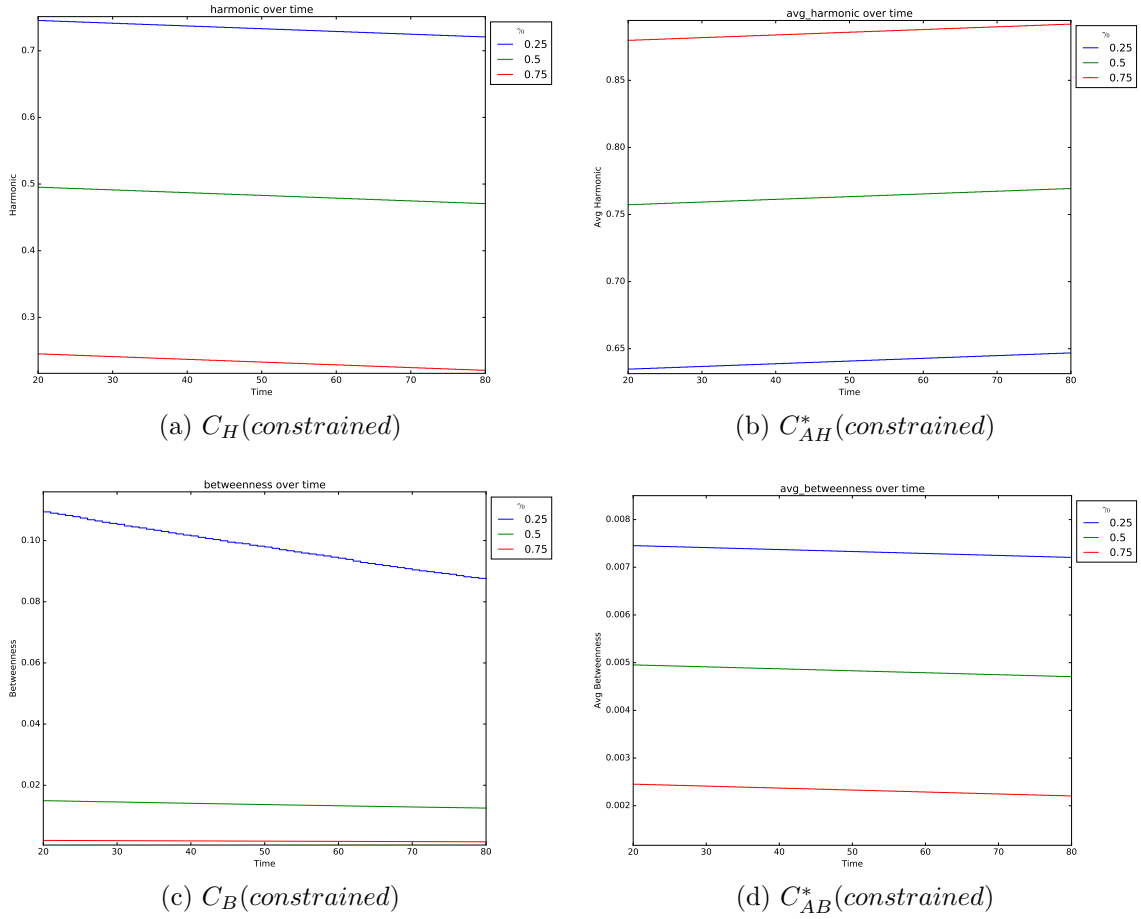


(a) $C_H(constrained)$ (b) $C^*_{AH}(constrained)$

(c) $C_B(constrained)$ (d) $C^*_{AB}(constrained)$

Figure 3.3: Centrality values for a constrained network of 100 nodes where $\mu = 1/2$ and density is decreasing. We see a proportionality relationship between average harmonic centrality and density, and an inverse proportionality with harmonic and average betweenness centrality.

Experiment 4. Since we have seen in Experiment 3 a proportionality without jitter, let us focus on the case where $\mu = 2$. We see the opposite effect when $\mu = 1/2$. Figure 3.8 depicts the change of centrality across the four different metrics, C_H, C_B, C^*_{AH}, and C^*_{AB}. Like the previous experiments, the metric values are proportional to the increase in the density until the network becomes functionally complete. Figure 3.9 shows a similar effect for the unconstrained network. The networks do not become complete due to the factor of η edges added and removed at each point. With our $\mu\eta$ construction, the calculation removes η/μ edges, leaving a maximum $N(N-1)/2 - \eta/2$ edges at any



(a) $C_H(constrained)$ (b) $C^*_{AH}(constrained)$

(c) $C_B(constrained)$ (d) $C^*_{AB}(constrained)$

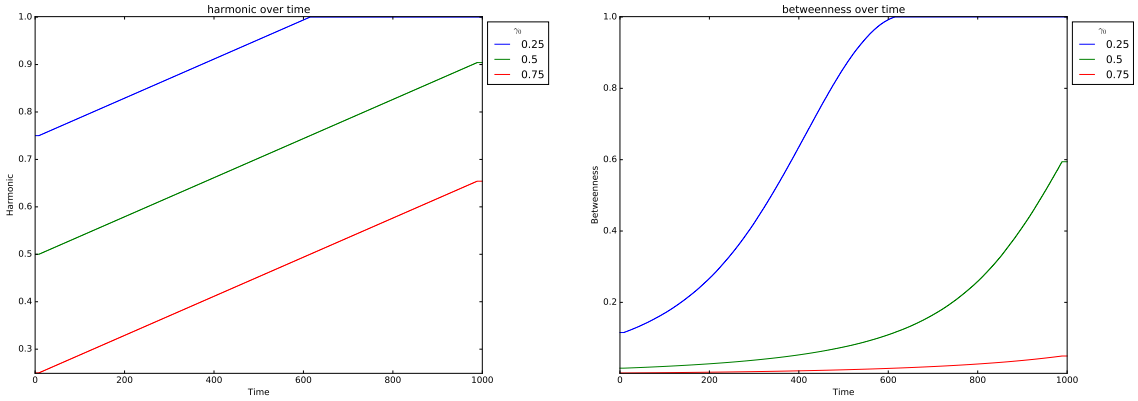
Figure 3.4: Centrality values for a constrained network of 100 nodes where $\mu = 2$ and density is increasing. Like Figure 3.3, we see a proportionality relationship between average harmonic centrality and density, and an inverse proportionality with harmonic and average betweenness centrality.

point in time.

We must address the unanticipated slopes on our networks: a major shortcoming of this experiment and the next two, due to our original experimental design. Thus, our reasoning for re-configuring the TVG generation and considering Experiments 7, 8, and 9, is the inter-dependency between μ and η in Algorithm 2. Specifically, as seen in Figure 3.8, the magnitude of jitter affects the slope of the change in density of the graph due to μ. We would like to understand the effect of jitter on the metric values without the consequential effect on the density.

Experiment 5 and 6. Our expected and unexpected change experiments performed much like our constant μ with jitter. Therefore, we postpone their discussion to our revised experimental setup in Experiments 8 and 9.

Experiment 7. With an ability to separate jitter and prescribed network density change, we can



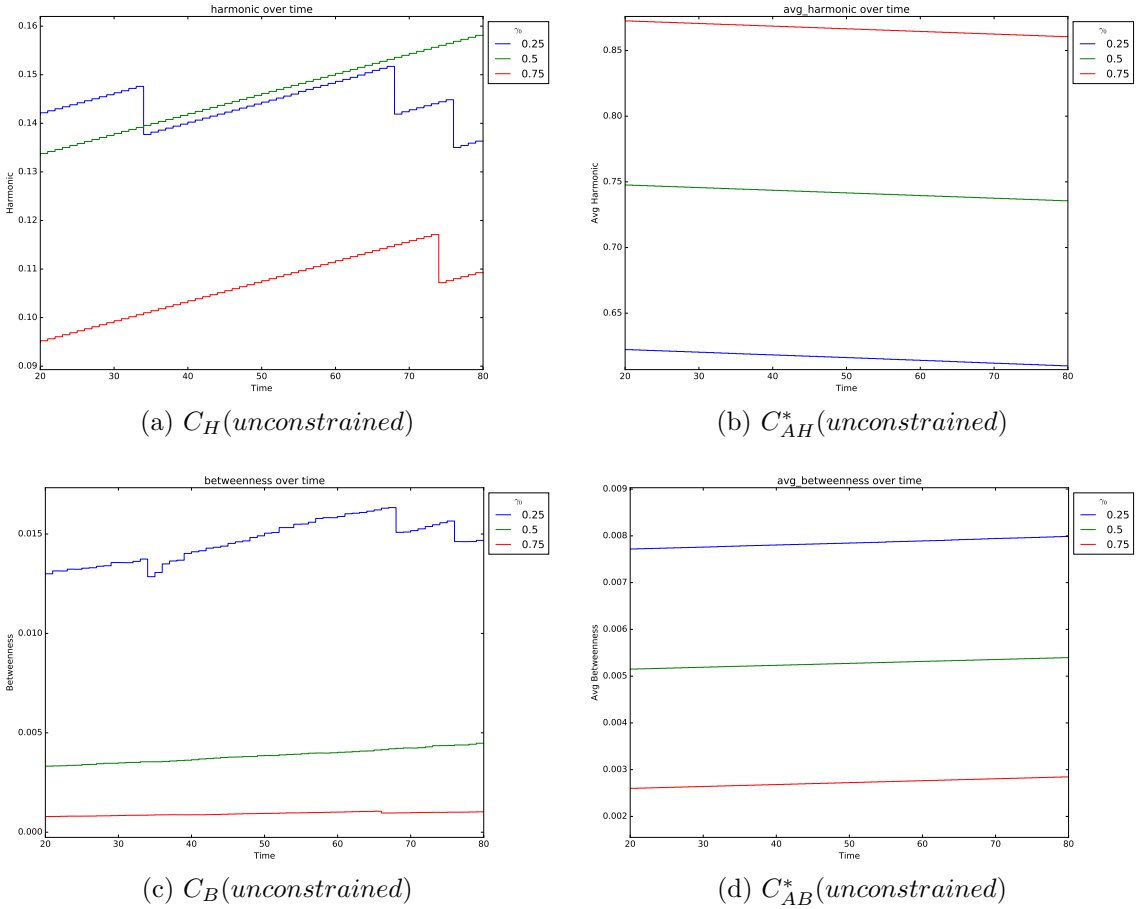
(a) $C_H(constrained)$ (b) $C_B(constrained)$

Figure 3.5: Extending the lifespan of the constrained network in Figure 3.3, with 100 node, $\mu = 1/2$, and density is decreasing, we see the proportionality relationship continue until the network includes only the minimal star.

clearly see that our metrics are able to track, proportionally, the change in the network even with jitter. Figure 3.10 depicts the metric values across varying amounts of jitter over a constrained TVG consisting of 100 nodes that grows by 50 edges at each timepoint.

Comparatively, Figure 3.11 depicts a similar, but unconstrained network. Most of our measures behave similarly to the constrained case: the density measurements are identical, accounting for the removal of the $N-1$ star edges, with the other average metrics maintaining proportionality. However, the harmonic and betweenness centralities are affected by the jitter in the network. Therefore it appears we can use density as an indicator of approximate average betweenness or average harmonic centrality, but the graph-wide variants capture the inner dynamics of the jitter.

Experiment 8. The purpose of this experiment was to exhibit a wave-like change in the network, simulating an expected occurrence with an equivalent increase and decrease period. Since our initial experiments conflated the calculation of μ and η, we were unable to capture the expected-behavior motion: under our previous construction, the slopes of the increase and decrease were dependent on the amount of jitter. In this experiment, we see that our metrics were able to track the network as it became increasingly connected during the event period, then returned to normal. Figure 3.12 shows the metric values for a constrained network, including the number of edges based on our μ calculation in Figure 3.12f. As experienced before, jitter does not affect the metric values, except minimally in the case of betweenness centrality. Likewise, Figure 3.13 depicts the metric values for a similar unconstrained network. In this case, the average centralities are unaffected by the jitter, while the graph-wide centrality metrics evidence a fluctuation proportional to the maximal node-centric centrality value. We note that the maximal value is fixed at the hub in the constrained case, which



(a) $C_H(unconstrained)$ (b) $C^*_{AH}(unconstrained)$

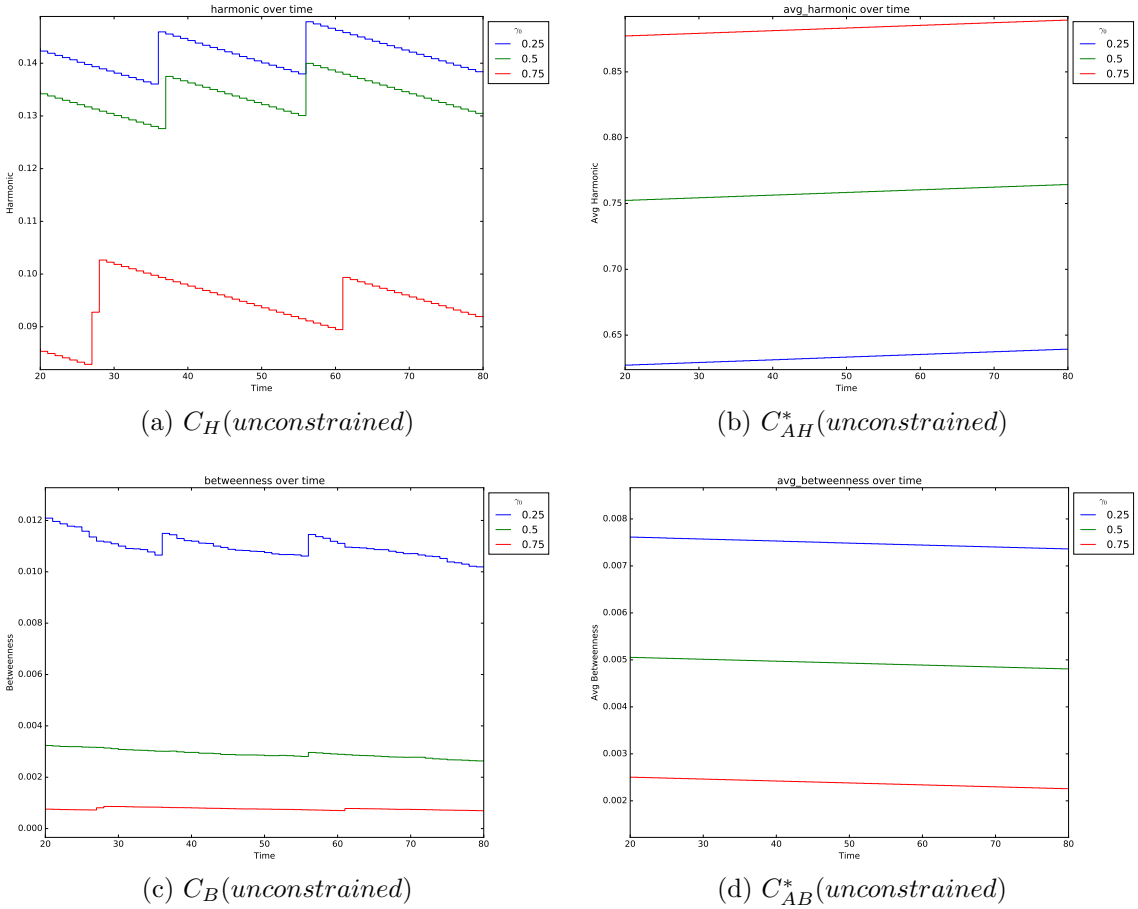
(c) $C_B(unconstrained)$ (d) $C^*_{AB}(unconstrained)$

Figure 3.6: Centrality values for an unconstrained network of 100 nodes where $\mu = 1/2$ and decreasing density. We see a proportionality relationship between average harmonic centrality and density, and an inverse proportionality with average betweenness centrality. When an edge is removed that affects the node with maximal centrality, we see a drop in the harmonic and betweenness centralities.

removes the fluctuations. From this experiment, we note that we should not rely on the graph-wide harmonic value in the unconstrained case, as its value does not approximate the no-jitter network as well as betweenness.

An important question posed during this experiment should also be addressed. Namely, in the experimental design, we speculated that the Begin and End sampling methods would shift the peak in metric centrality by our sampling window size. We have shown in Section 2.1 that the property theoretically holds and Figure 3.14 provides an empirical example using average harmonic centrality when $\lambda t = 5$. Similarly, we note that the Union sampling method extends the width of the peak while the Intersection and Begin-End sampling methods lower the peak's maximal value.

Experiment 9. In this experiment, which closely mirrors Experiment 8, our metrics exhibit a similar pattern. Figure 3.15 depicts the metric values for an example constrained network with 100 nodes

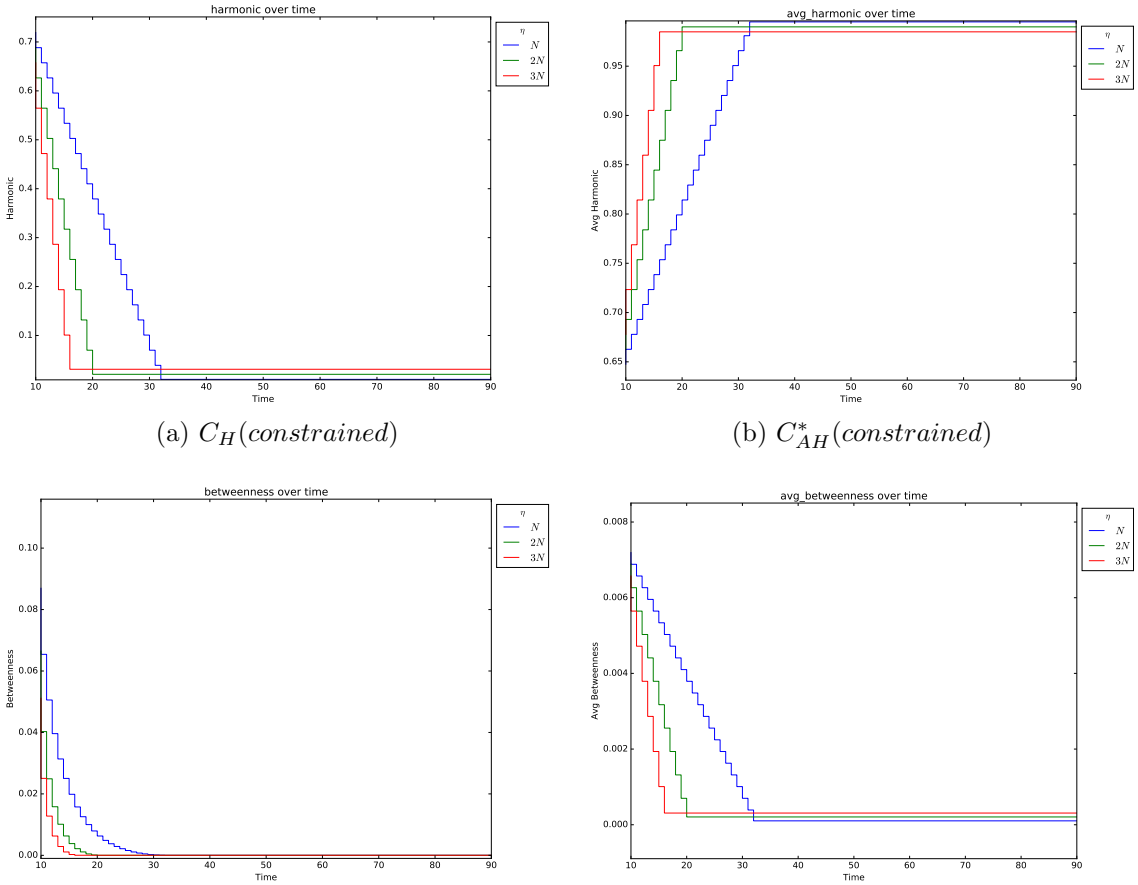


(a) $C_H(unconstrained)$ (b) $C^*_{AH}(unconstrained)$

(c) $C_B(unconstrained)$ (d) $C^*_{AB}(unconstrained)$

Figure 3.7: Centrality values for an unconstrained network of 100 nodes where $\mu = 2$ and density increasing. Here we see a similar but opposite pattern to Figure 3.6.

and varying amounts of jitter across its lifespan. We use the second value of $\mu(t)$ in this depiction to capture the full change in the network due to the unexpected occurrence without causing the network to become complete. If the network were to become functionally complete, we would see a divergence of the metrics based on jitter, since we have previously shown that the maximum number of edges is dependent on η. In each of these plots, we see a higher slope to the maximal density, followed by a slower return to normal. Figure 3.16 depicts similar results for an unconstrained network. Like our last example, we find that the harmonic centrality is the only measure that does not track the metric value for a network with little jitter.



(a) $C_H(constrained)$ (b) $C^*_{AH}(constrained)$

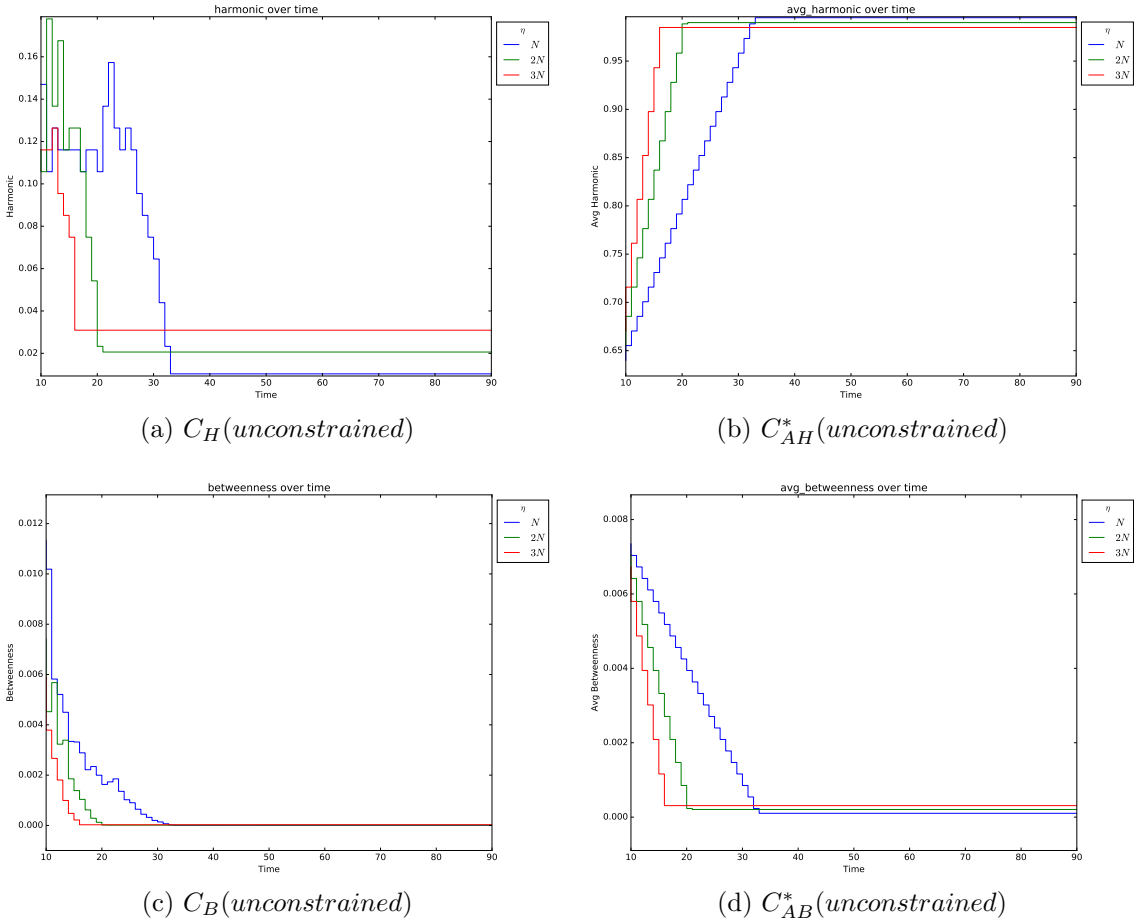
(c) $C_B(constrained)$ (d) $C^*_{AB}(constrained)$

Figure 3.8: Centrality values for a constrained network of 100 nodes where $\mu = 2$ and $\eta = N, 2N, 3N$. Since the number of added edges is $\mu \cdot \eta$, we see a slope proportional to the choice of η.

3.5 Sampling Size Effects

Our experiments were designed to show a connection between the topology of the network and the calculation of our metrics over time. Since we have been able to evidence the network change in the metrics themselves, we turn now to approaches aimed at determining the dynamics of the network. Specifically, is there significant change in the network at any point in time? And, how dynamic is the network over its lifespan?

We start by examining the effects of sampling window size and choice of sampling method performed on the calculated measures of the network over time. In fact, we see a distraction effect as sampling window size compounds with the graph's activity and jitter. This effect increases as the network becomes more active. Additionally, as the sampling window increases in size, additional edges are factored into the sampled graph and likewise affect the measure. By comparing the magnitude



(a) $C_H(unconstrained)$ (b) $C^*_{AH}(unconstrained)$

(c) $C_B(unconstrained)$ (d) $C^*_{AB}(unconstrained)$

Figure 3.9: Centrality values for an unconstrained network of 100 nodes where $\mu = 2$ and $\eta = N, 2N, 3N$. Jitter is more apparent on the harmonic (a) and betweenness centrality (b).

of change between interval sizes along with the metric values, we uncover some dynamics of the network.

Consider Figure 3.17. In the network on the left-hand side (a), we have a small change—at most one edge—at each time point. On the right (b), there are up to 200 edges coming and going at each time point ($\eta = 2N$, where $N = 100$). Here, the Union sampling is used, which shows a decrease in the graph-wide betweenness centrality as additional edges are included in the sampled graph. Even though the magnitude of the betweenness centrality metric is small, the change in the metric value for increasing larger window sizes is significantly smaller than the network with higher activity (jitter). This change is evidenced in both the constrained networks and the unconstrained networks. Since the constrained networks guarantee at least one node of highest betweenness—the hub—which factors into the calculation of C_B, we expect a smaller value for the unconstrained network since that maximal node is not guaranteed. We can account for the change in metric value of the larger

sampled graph $\lambda t = 4$ compared with the $\lambda t = 0$ snapshot view since, the larger sampling window may produce a graph with at most 8 additional edges included through the Union sampling for $\eta = 1$ and an additional 1600 edges for $\eta = 2N$. Let us therefore formalize these properties.

3.5.1 Properties Due to Sampling Size in Synthetic Experiments

Since we begin to notice a trend in our synthetic results based on sampling size choice, we make some theoretic observations about the network behavior as each event happens. Let us consider the Stable Jitter Sensitivity experiment, Experiment 2, which defines a network that should maintain a consistent number of edges while allowing those edges to jitter. In generating the TVG, *GenerateGraph* in Algorithm 2 adds at most η edges and removes at most η edges per timestep with $\mu(t) = 1$. If no overlap exists in the edges chosen to be added and removed, then there will be η new edges added and η old edges removed. On the other hand, if there is complete overlap, then $E_t = E_{t-1}$. By taking the union of two time-adjacent—therefore also event-adjacent—edge sets, we find that there are at most $\gamma_0 + \eta$ edges in the union. Specifically,

$$\forall t \in T, |E_t| = \gamma_0 \text{ and } 0 \leq |E_t \cup E_{t-1}| - |E_{t-1}| \leq \eta.$$

Conversely, we may also consider the intersection of two time-adjacent edge sets. We find that

$$\forall t \in T, |E_t| = \gamma_0 \text{ and } 0 \leq |E_t| - |E_t \cap E_{t-1}| \leq \eta.$$

These findings may be expanded to include multiple time-adjacent edge sets, or time windows. For simplicity, consider the union over a λt-sized interval starting at time t and including $\lambda t + 1$ time-adjacent edge sets.

$$\forall t \in T, \lambda t \in \mathbb{N}, 0 \leq \left| \bigcup_{w=t}^{t+\lambda t} E_w \right| - |E_t| \leq \lambda t \cdot \eta \leq |E_c|.$$

Including the λt time-adjacent edge sets before t, we arrive at the maximum number of edges under the Union sampling with η jitter:

$$\forall t \in T, \lambda t \in \mathbb{N}, 0 \leq \left| \bigcup_{w=t-\lambda t}^{t+\lambda t} E_w \right| - |E_t| \leq 2\lambda t \cdot \eta \leq |E_c|.$$

By examining the empirical edge counts over multiple runs, not only do we find that the edge count exhibits this behavior, but we also discover that the number of edges is a lower percentage of $2\lambda t\eta$ as λt increases. As jitter increases the percentage of the graph's total edges that are candidates for changing, the proportion of sustained increase in edges drops. Let us start by considering the constrained network. Table 3.2 shows the average increase in number of edges as η and λt increase compared to the snapshot view of the network where $\lambda t = 0$. Our results are averaged over 1200 time intervals for 100-node graphs that began with an initial density of 50%, or $\gamma_0 = 0.5$. For example, when $\lambda t = 5$, the sampled graph will cover 10 time units; in a network with little jitter, we find an average of 9.98 additional edges are considered in the sampling, or about 98% of the maximal 10 additional edges if there were no overlap of changes among the 10 time units. Consequently, with the $\lambda t = 5$ time window and $3N$ jitter, we find only 1670 additional edges on average, or 55.7% of our theoretic $2\lambda t \cdot \eta$ maximum. With larger jitter and uniform random selection from the available 2475 edges, we expect a higher overlap. Table 3.3 provides the list of percentage of maximum calculations for each value in Table 3.2.

Table 3.2: Union Sampling: Increase in average number of edges for each η and λt combination over the constrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0.00	0.00	0.00	0.00	0.00
1	2.00	96.98	188.17	355.46	504.29
2	3.99	190.09	361.89	659.10	903.90
3	5.99	279.53	522.33	918.38	1,220.51
4	7.98	365.47	670.25	1,139.73	1,471.36
5	9.98	447.97	806.70	1,328.70	1,669.72

Similarly, we find the property holds for the Intersection sampling,

$$\forall t \in T, \lambda t \in \mathbb{N}, 0 \leq |E_t| - \left| \bigcap_{w=t-\lambda t}^{t+\lambda t} E_w \right| \leq 2\lambda t \cdot \eta \leq |E_c|.$$

Tables 3.4, 3.5 evidence these results.

Since our network construction for both the constrained networks and unconstrained networks were

Table 3.3: Union Sampling: Average sustained percentage of the additional $2\lambda t \cdot \eta$ edges for each η and λt combination over the constrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0.00	0.00	0.00	0.00	0.00
1	99.92	96.98	94.08	88.87	84.05
2	99.86	95.05	90.47	82.39	75.33
3	99.82	93.18	87.05	76.53	67.81
4	99.81	91.37	83.78	71.23	61.31
5	99.79	89.59	80.67	66.44	55.66

Table 3.4: Intersection Sampling: Decrease in average number of edges for each η and λt combination under the constrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0.00	0.00	0.00	0.00	0.00
1	-2.00	-96.96	-188.28	-355.53	-504.55
2	-4.00	-189.97	-362.12	-659.29	-904.05
3	-5.99	-279.22	-522.36	-918.31	-1,220.30
4	-7.99	-364.85	-670.03	-1,139.24	-1,470.93
5	-9.98	-447.06	-806.34	-1,327.79	-1,669.73

Table 3.5: Intersection Sampling: Average sustained percentage of the $2\lambda t\eta$ less edges for each η and λt combination over the constrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0	0	0	0	0
1	99.92	96.96	94.14	88.88	84.09
2	99.92	94.99	90.53	82.41	75.34
3	99.89	93.07	87.06	76.53	67.79
4	99.87	91.21	83.75	71.2	61.29
5	99.84	89.41	80.63	66.39	55.66

identical, aside from the minimal-star requirement, we find empirically that they evidence the same results. We leave those results for Appendix D.

3.5.2 Aliasing Effect Based on Sampling Size

By combining the metric properties and effects due to λt size, we can discuss an aliasing effect on metric measurements due to sampling. In signal processing, aliasing is an effect of a sampling in which different signals are indistinguishable. Specifically, Oppenheim and Schafer define *aliasing* as when a higher-frequency signal takes the identity of a lower-frequency signal "as a consequence of the sampling and reconstruction" [35, 36].

As found by Ribeiro in [1], aliasing may hide certain aspects of the network dynamics. He defined a polling interval, or "integration window," where the network state is reported at every Δt. Moreover, each reporting of the network state coalesces—or integrates—all components that exist in that interval. Even though our synthetic experiments utilize a discrete notion of time, we emulated his results with a λt interval and our Union sampling method. For example, let us consider two TVGs, one with $\eta = N$ jitter and one with $\eta = 2N$ jitter. We define two of Ribeiro's polling intervals $\Delta t = 2$ and $\Delta t = 4$ as our own $\lambda t = 1$ and $\lambda t = 2$, respectively. As Figure 3.21e shows, we find a relationship between sampling size and network size: the number of edges present in the sampling is greater than at any point in the underlying network, as Ribeiro discovered with his random walks. In Figure 3.21a-b, we uncover a relationship between sampling size and node-centric network activity that evidences aliasing: a less-active network may appear more active under longer sampling—or integration window—sizes. The less active network with $\eta = N$, sampled with window size 4, depicted by the red line, appears as active as the network with twice the jitter, $\eta = 2N$, sampled with window size 2 and depicted by the green line, as reported by average harmonic and betweenness centralities. In comparison, we note that the computation of the graph-level centrality measures, depicted in Figure 3.21c-d, mitigate the aliasing effect due to sampling window size and jitter, since they evidence more of the network structure at each point which is relatively consistent for these networks.

3.5.3 Dynamics Through Sampling Size Comparison

We therefore use these results to uncover dynamics in the network based on the effects of sampling size choice. Rather than sampling at larger Δt strides, we compute samplings at the same points with varying λt sampling sizes. From our analysis on constrained networks, as λt increases with Union sampling, we find that harmonic, betweenness, and average harmonic centralities will evidence an artificial increase while average betweenness centrality will be artificially low compared with that of the actual evolving network observed through snapshots. We note that for unconstrained networks, the properties hold for average centralities C^*_{AH} and C^*_{AB} as in Figure 3.21, but only on average for graph-wide measures C_H and C_B, i.e., we lose the strict monotonicity in the graph-wide measures as λt increases. Returning to Figure 3.18 shows effect of increasing interval sizes on the metric values compared with the $\lambda t = 0$ baseline measure over a constrained network with $\eta = 2N$ jitter.

Each of our sampling methods, excluding Transitive, produce a different kind of distraction as a sampling window size increases. By using our Intersection sampling, we observe the inverse result to

the Union sampling. Namely, a decrease in the edge count in each sampling and therefore a decrease in harmonic, betweenness, and average harmonic centralities with an increase in average betweenness centrality. The Begin-End sampling method, which requires any edge to exist at the beginning and end of the time window, performs similarly. Figure 3.22 compares the sampling methods across a network with $\eta = 3N$ jitter and a sampling size of $\lambda t = 3$.

We therefore have shown that a network with more activity, such as increased jitter, will report an increased average centrality measure as the window size increases under our additive sampling method, the Union sampling. Likewise, a decreased average centrality for the other sampling methods. By examining the centrality measures over various sampling windows and comparing their differences, we can therefore evidence how active the network is across time. If the magnitude of their difference is small, we expect little activity; however larger magnitude differences may indicate higher activity level.

3.6 Dynamics Through Metric Derivatives

Next, we turn to examining the effectiveness of our derivative measure, $D(\mathcal{G})$, to identify an activity profile from the network over time. In order to measure this in the absence of sampling size interference, we confine our application to $\lambda t = 0$.

For our constrained set of experiments, we have shown that jitter has little effect on the measurements of our metrics due to the properties of the network under the minimal-star assumption. We can see the same result when applying our $D(\mathcal{G})$ dynamics measure. Figure 3.23a shows a constrained example for a 100-node network with $3N$ jitter. In this case, μ remains fixed at 1 and therefore the network maintains a stable density of 0.5 without any dynamic points. When the network exhibits change, as in Figure 3.23b, the magnitude of change is evidenced. In this sub-figure, we depict $D(\mathcal{G})$ over a similar 100-node constrained network with $3N$ jitter, but this time with an expected occurrence as described in Experiment 8. We see a sustained change in the network throughout the entire event, with a slight dip as the network settles at its peak connectivity before returning to its "normal" inactivity.

In contrast, networks without the minimal-star constraint show a greater volatility with jitter. Therefore, we must examine the resulting profiles more closely and develop a set of heuristics to

analyze them. Figure 3.24 depicts $D(\mathcal{G})$ over networks with four different levels of inherent change. For each TVG, there are 100 nodes and we compare between $\eta = 1$ jitter and a consistent $3N$ jitter that occurs at every timepoint between 10 and 90. In our first plot, we start with a network that is 50% complete. Compared with the baseline, in which μ and η are both 1, the dynamics measure is more volatile with larger jitter. In this case, the graph is changing topology without changing density; the changes in the $D(\mathcal{G})$ measure are the result of changing betweenness and harmonic centrality. Some cases will evidence a significant change in the network, especially higher centrality changes. By examining each highly dynamic point, we may set a threshold by heuristic, in an attempt to remove noise from the points that define a more significant graph change. This technique is used in our later real-world experiments to uncover significant points of change in the network. Specifically, we choose a threshold to highlight the maximal values and relax it as the network is examined at the dynamic timepoints for importance, allowing our choice of threshold to be determined by domain-specific network properties.

The other three plots in Figure 3.24 evidence a similar characteristic to their constrained counterparts: sustained dynamics across density change. In Figure 3.24b, we observe a network that begins with 25% density and has 100 edges randomly added at each time point, beginning at $t = 10$. By $t = 35$, the network has reached its maximal density, which based on our experimental setup, is η edges less than a complete graph. After observing this sustained dynamics value in the profile plot, we may verify the positive increase in the connectedness of the graph by checking the components of $D(\mathcal{G})$. Let us set a threshold of 0.02 to filter for the sustained values and compare the components for $\eta = 1$ in Table 3.6. Here we see a consistently negative change in the harmonic and betweenness centralities, denoting a drop in centrality disparity, coupled with an increase in density: the network is becoming more complete. Or more specifically, as the network is becoming more dense, the nodes are becoming more equally central.

Figure 3.24c follows a similar pattern to that of Figure 3.23b. With only a small amount of jitter, $\eta = 1$, we see a nearly identical dynamic profile except when the "jittered" edge causes a change in the centrality of the network in a way not possible under our constrained construction. Jitter affects this network in the same way as the other networks in Figure 3.24, however through examining the sustained dynamics and $D(\mathcal{G})$ component ΔC_D, we can identify an overall trend in the network. It is fairly active due to the jitter as evidenced by ΔC_B and ΔC_H, but increasing in connectedness through $t = 50$ and then decreasing as evidenced by ΔC_D.

Table 3.6: $D(\mathcal{G})$ values above the 0.02 threshold and its components for a synthetic network without a guaranteed minimal star and $\mu = N$.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
10	0.0288784	-0.0206143	-0.0009453	0.0202020
11	0.0288716	-0.0206143	-0.0007049	0.0202020
12	0.0202020	0.0000000	-0.0000076	0.0202020
13	0.0202027	0.0000000	-0.0001703	0.0202020
14	0.0202031	0.0000000	-0.0002075	0.0202020
15	0.0369373	0.0309215	0.0003262	0.0202020
16	0.0226795	0.0103072	0.0000276	0.0202020
17	0.0288660	-0.0206143	-0.0004155	0.0202020
18	0.0226806	-0.0103072	-0.0002260	0.0202020
19	0.0226795	0.0103072	-0.0000533	0.0202020
20	0.0202021	0.0000000	-0.0000534	0.0202020
21	0.0288645	-0.0206143	-0.0002929	0.0202020
22	0.0202029	0.0000000	-0.0001869	0.0202020
23	0.0226795	0.0103072	0.0000394	0.0202020
24	0.0226795	0.0103072	-0.0000455	0.0202020
25	0.0288635	-0.0206143	-0.0001763	0.0202020
26	0.0226797	-0.0103072	-0.0000963	0.0202020
27	0.0226798	-0.0103072	-0.0001225	0.0202020
28	0.0226796	-0.0103072	-0.0000693	0.0202020
29	0.0202021	0.0000000	-0.0000549	0.0202020
30	0.0288630	-0.0206143	-0.0000383	0.0202020
31	0.0226795	0.0103072	-0.0000262	0.0202020
32	0.0288630	-0.0206143	-0.0000467	0.0202020
33	0.0288630	-0.0206143	-0.0000286	0.0202020
34	0.0213586	-0.0152546	-0.0000075	0.0149495

In the case of an unexpected occurrence, Figure 3.24d, the jitter obscures the rate of density change. The overall distribution follows the version with little jitter, however the change in centrality measures outweigh the density changes in our computation of $D(\mathcal{G})$. This warrants analysis of $D(\mathcal{G})$'s components as well as the network to determine the significance of the change caused by jitter. However, our changing density component, ΔC_D, still evidences the prescribed change in the network.

Our synthetic examples are designed to show significant change in terms of both jitter and network topology. We do not expect to see as much jitter in our real-world experiments, which are both less dense and less dynamic. Their important changes rely on the changing structure of the network more than the change in density that our synthetic experiments utilize. We therefore expect that the effects we see through jitter in these experiments to play a more important role in the real-world experiments. Since our $D(\mathcal{G})$ measure gives equal weight to the betweenness and harmonic centralities as it does to changing density, we see that it is effective at uncovering the highly dynamic points of those networks. In this case, we limit our analysis of the $D(\mathcal{G})$ measure to depicting the dynamic profiles and using

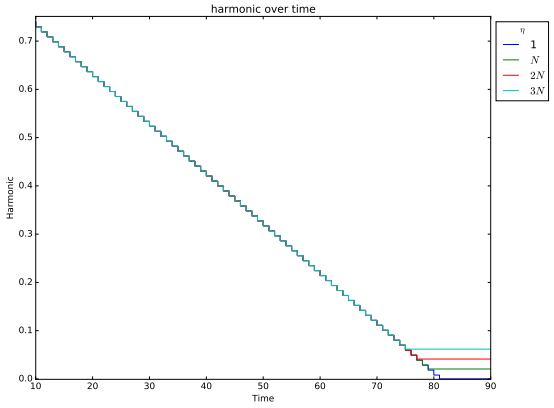
the threshold measure highlight dynamic timepoints for further examination.

3.7 Implications of Experiments

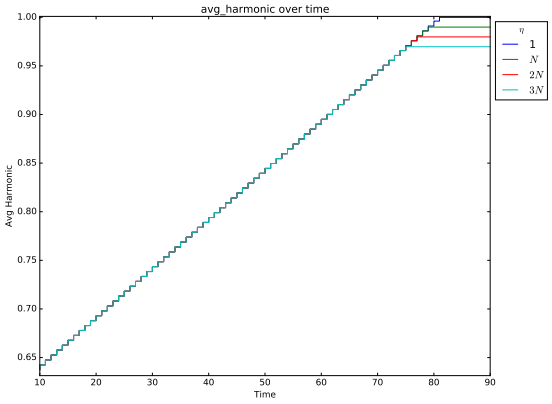
In each of our synthetic experiments, we have shown that calculating the metrics' distributions over time depicts the prescribed changes in the network. The metrics evidence the density changes even in the presence of jitter, for all centrality measures over our initial constrained networks and average centrality measures over the unconstrained variations. In the unconstrained case, we observed that the harmonic and betweenness centrality values evidence the jitter effect, with betweenness centrality's distribution approximating the jitter-free distribution. These observed jitter effects tell us about the underlying topology changes in the network and are important to capture when considering a profile of the overall dynamics.

By then comparing metric values over our sampling methods and across different-sized λt sampling windows, we uncovered a difference in metric value based on the level of network activity. In certain circumstances, an aliasing affect was observed, corroborating Ribeiro's analysis of polling window size effects [1] on a random walker's ability to traverse an evolving network. We also discovered that the shift in metric value due to sampling size may be used to depict periods of high activity. However, we will discover in the next chapter that this effect is dependent on the stability of the nodes across time and may therefore be less useful in uncovering dynamics of many real-world networks.

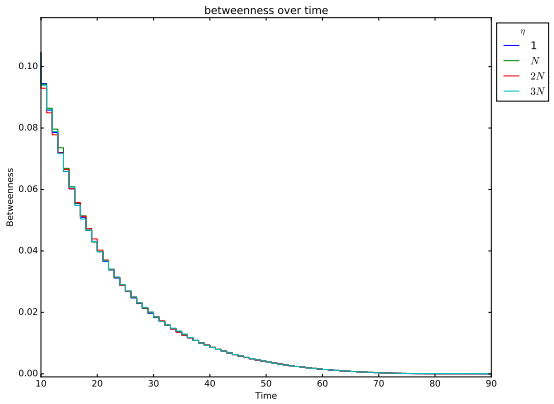
Lastly, we observed that our three centrality measures exhibited different patterns as the network changed. Degree centrality C_D reported the density of the graph, while betweenness C_B and harmonic C_H centralities changed as a result of network topology even under a constant density. Therefore, we created and utilized a measure of dynamics $D(\mathcal{G})$, as an L^2 norm combination of the three centrality values. By applying $D(\mathcal{G})$ to our synthetic data, we determined that the magnitude of the measure and sustained periods of change were both important to describing the changes in the network. We surmised that a threshold applied by heuristics highlights periods of high activity, including density changes and significant change as a result of jitter in these experiments. Through observing the sustained periods of change in $D(\mathcal{G})$ across time, we successfully captured network change with and without the effects of jitter. We use these findings about our techniques and apply them in the next chapter to make inferences from our real-world examples.



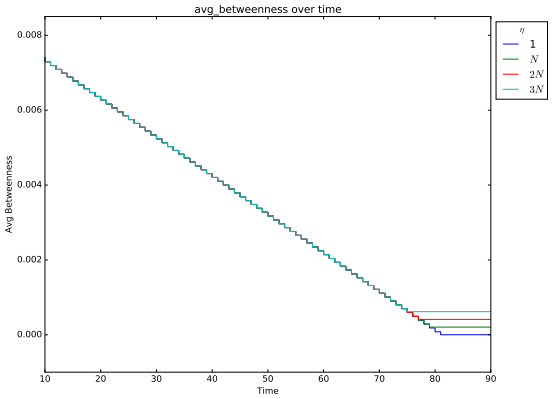
(a) $C_H(constrained)$



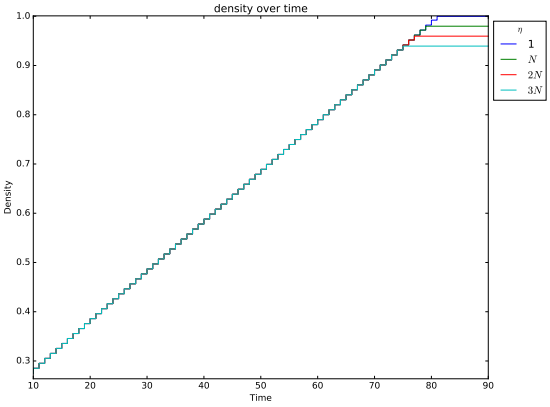
(b) $C^*_{AH}(constrained)$



(c) $C_B(constrained)$

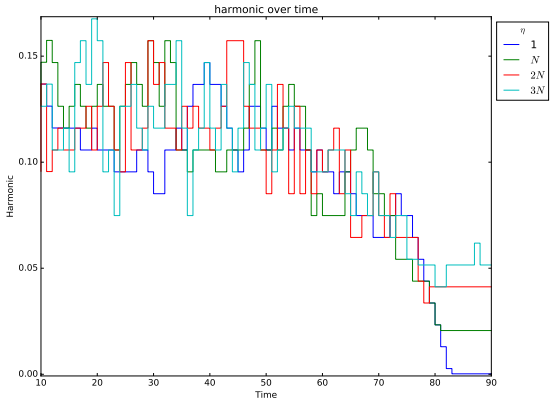


(d) $C^*_{AB}(constrained)$

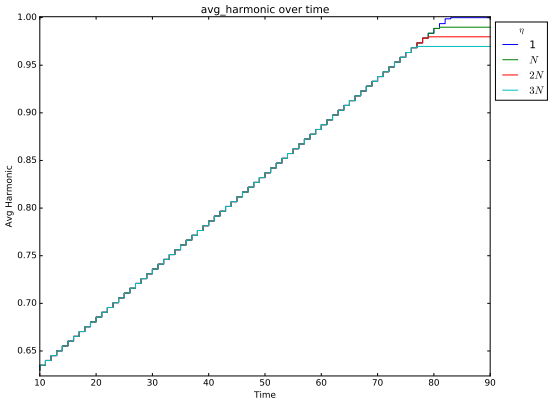


(e) $C_D(constrained)$

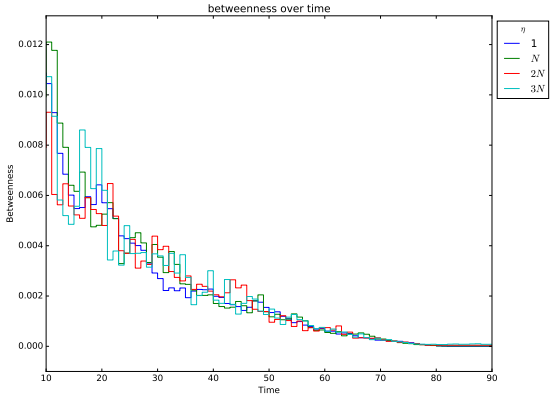
Figure 3.10: Centrality values for a constrained network of 100 nodes where $\mu = N/2$ and $\eta = 1, N, 2N, 3N$. Jitter has little effect until the graph becomes complete minus η edges.



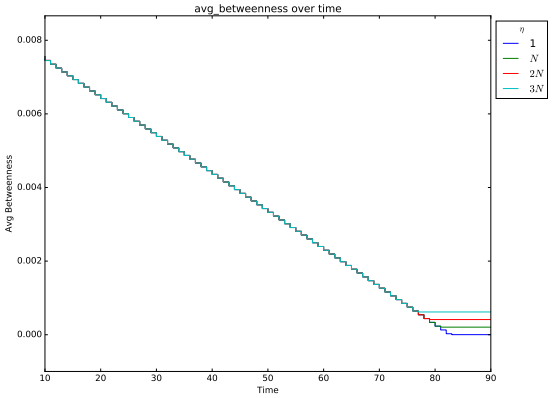
(a) $C_H(unconstrained)$



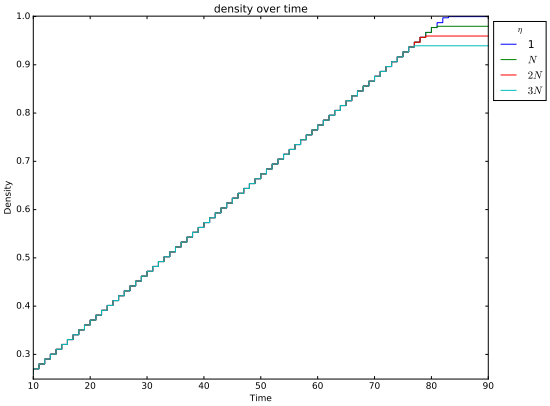
(b) $C^*_{AH}(unconstrained)$



(c) $C_B(unconstrained)$

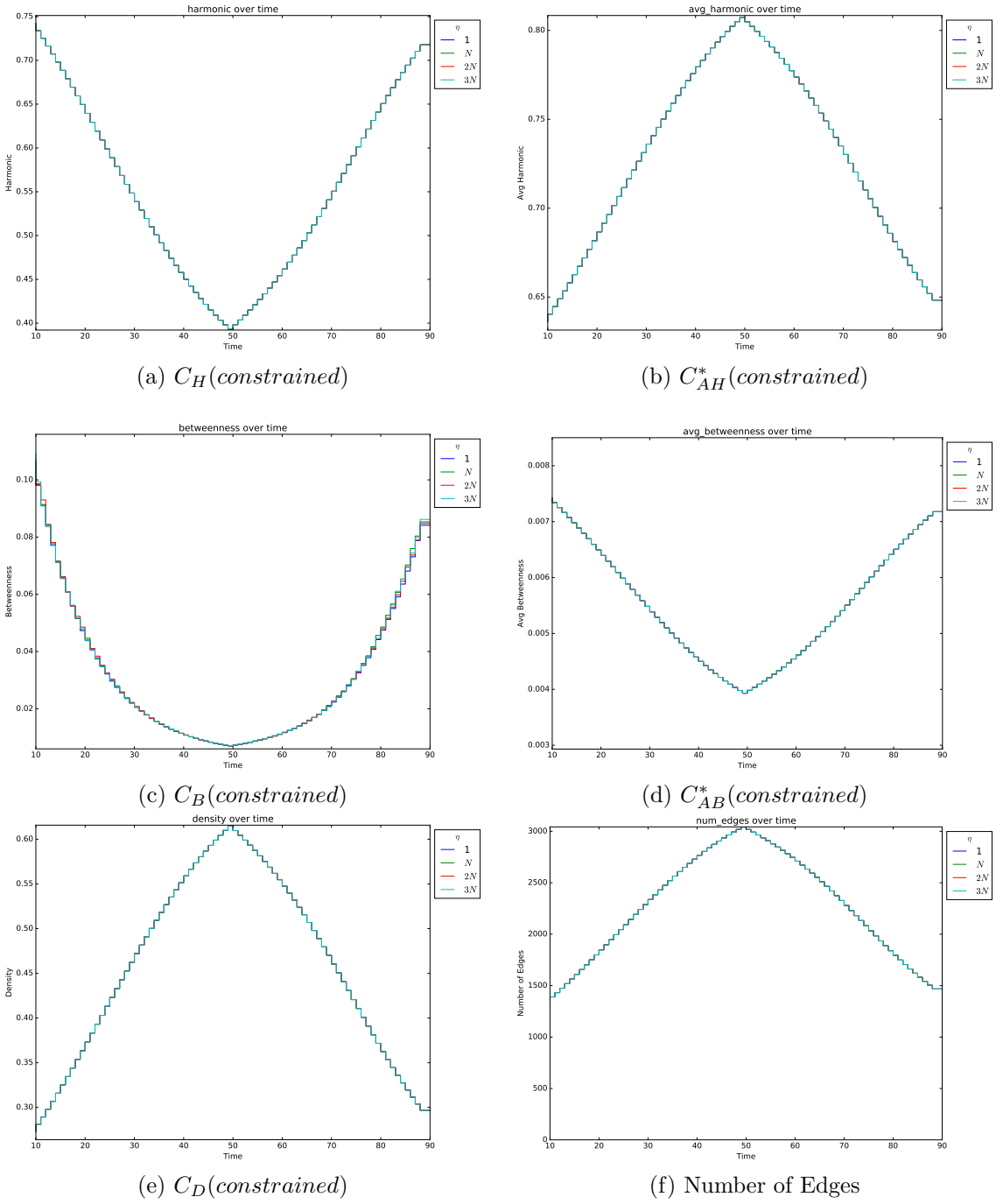


(d) $C^*_{AB}(unconstrained)$



(e) $C_D(unconstrained)$

Figure 3.11: Centrality values for an unconstrained network of 100 nodes where $\mu = N/2$ and $\eta = 1, N, 2N, 3N$. Jitter produces no effect on average centralities. Betweenness centrality (c) approximates the exponential decline of the similar constrained network in Figure 3.10.

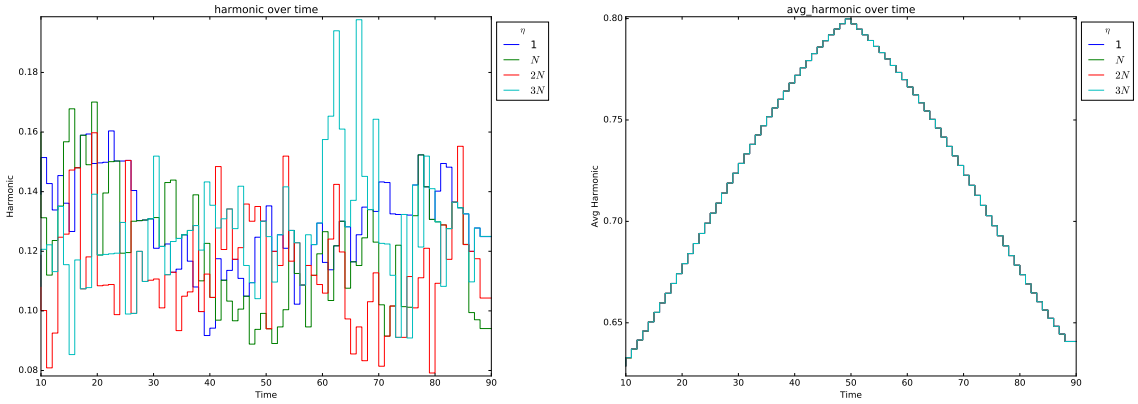


(a) $C_H(constrained)$ (b) $C^*_{AH}(constrained)$

(c) $C_B(constrained)$ (d) $C^*_{AB}(constrained)$

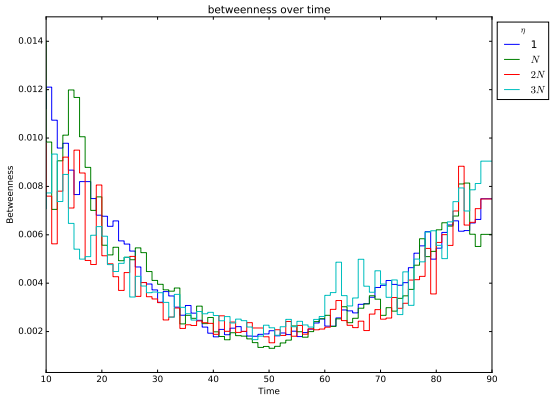
(e) $C_D(constrained)$ (f) Number of Edges

Figure 3.12: Centrality values for a constrained network of 100 nodes where μ prescribes an expected occurrence and $\eta = 1, N, 2N, 3N$. Jitter has little to no effect on any metrics.

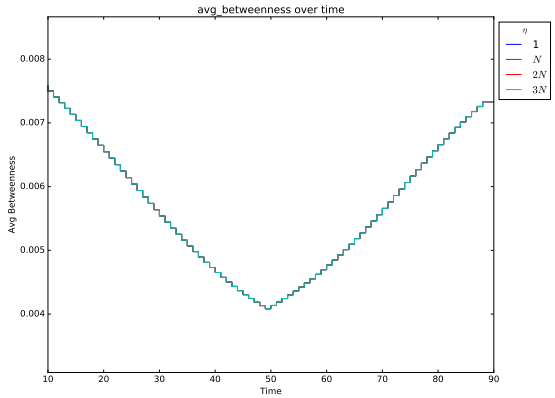


(a) $C_H(unconstrained)$

(b) $C^*_{AH}(unconstrained)$



(c) $C_B(unconstrained)$



(d) $C^*_{AB}(unconstrained)$

Figure 3.13: Centrality values for an unconstrained network of 100 nodes where μ prescribes an expected occurrence and $\eta = 1, N, 2N, 3N$. While jitter has no effect on the average centralities (b,d), betweenness centrality approximates the pattern of the measure on the constrained network and harmonic centrality (a) does not.

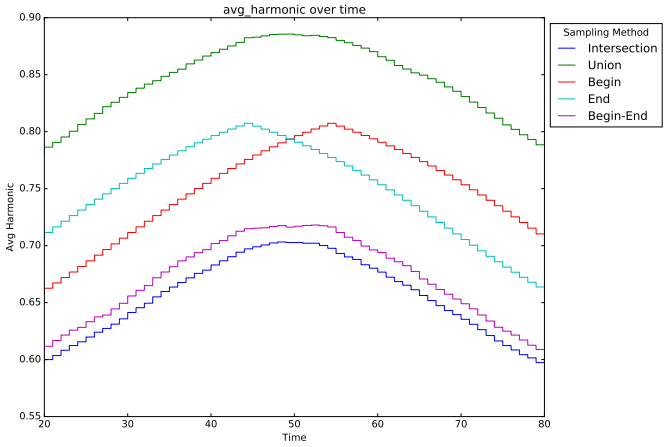
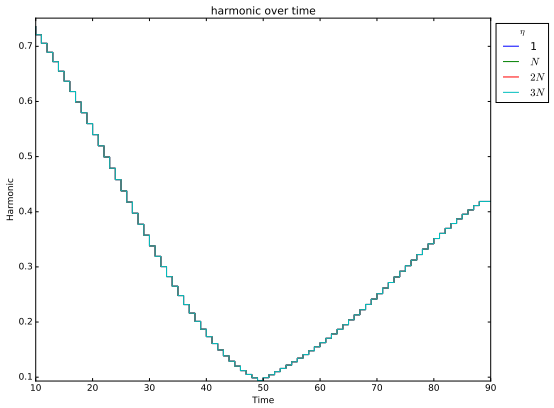
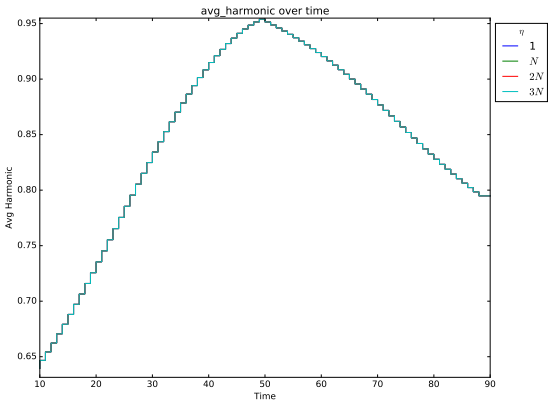


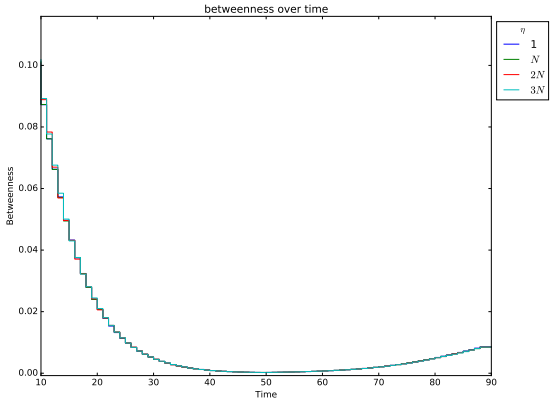
Figure 3.14: Our various sampling methods shift the peak in average harmonic centrality of the nodes during an expected occurrence. This constrained network contains $\eta = N$ jitter, 100-nodes, and is sampled with a window size of $\lambda t = 5$.



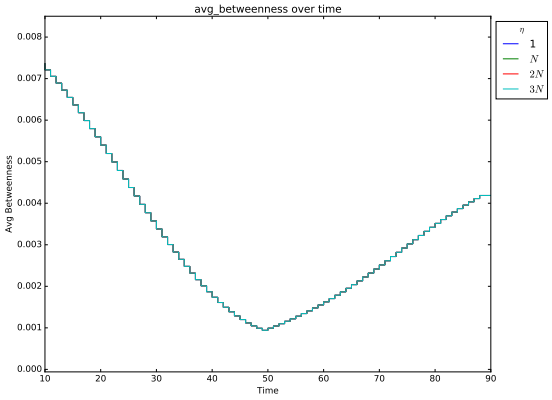
(a) $C_H(constrained)$



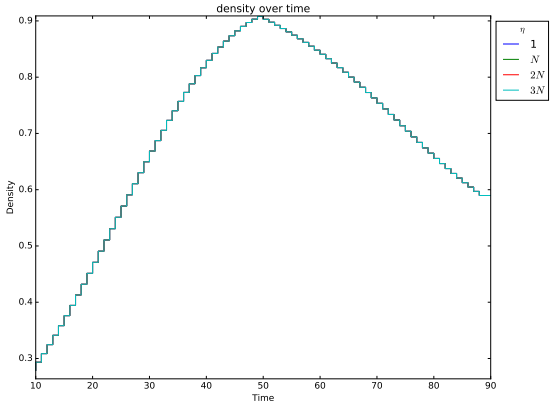
(b) $C^*_{AH}(constrained)$



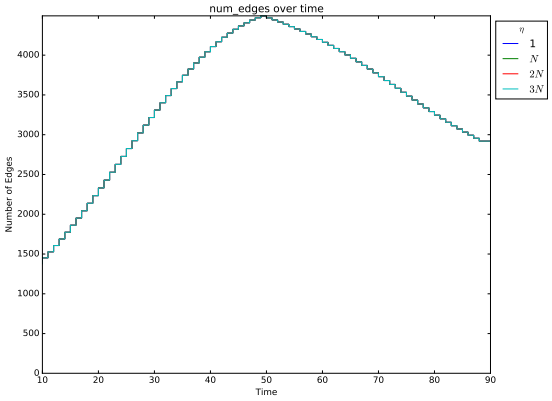
(c) $C_B(constrained)$



(d) $C^*_{AB}(constrained)$

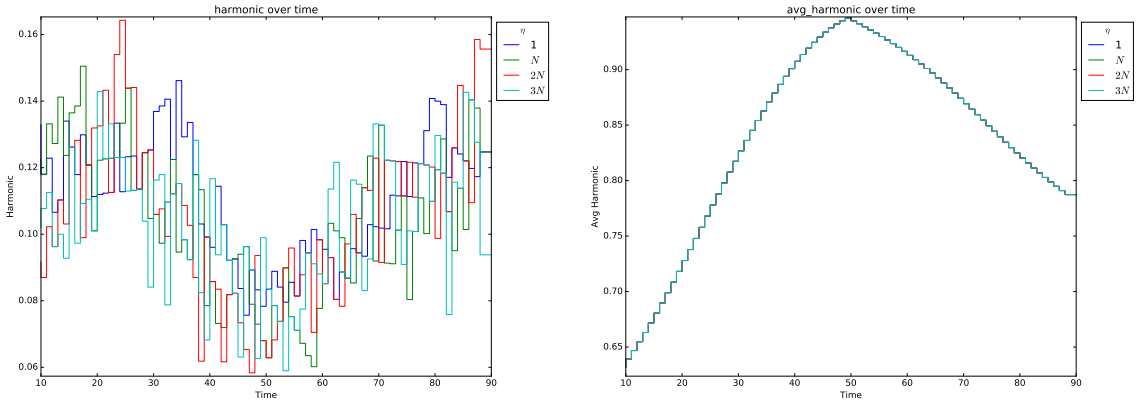


(e) $C_D(constrained)$



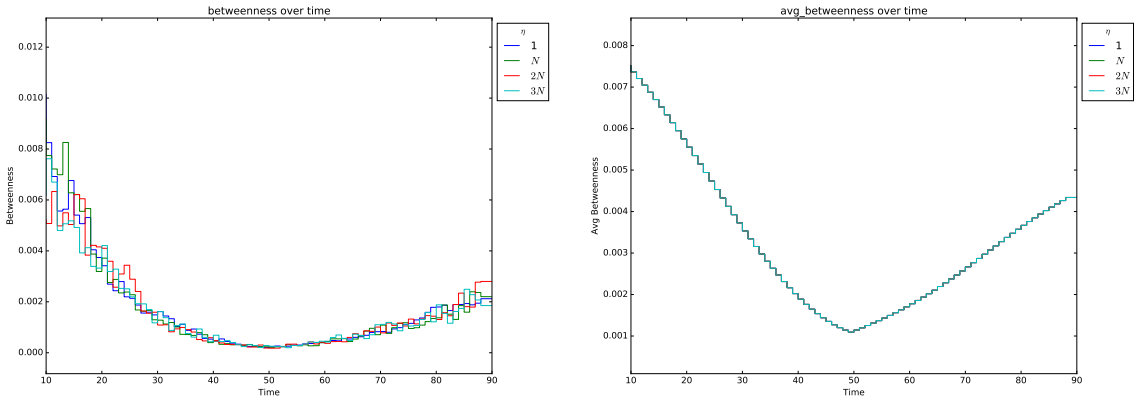
(f) Number of Edges

Figure 3.15: Centrality values for a constrained network of 100 nodes where μ prescribes an unexpected occurrence and $\eta = 1, N, 2N, 3N$. As expected, we see a steeper slope for the first-half of the lifespans with little to no change from jitter.



(a) $C_H(unconstrained)$

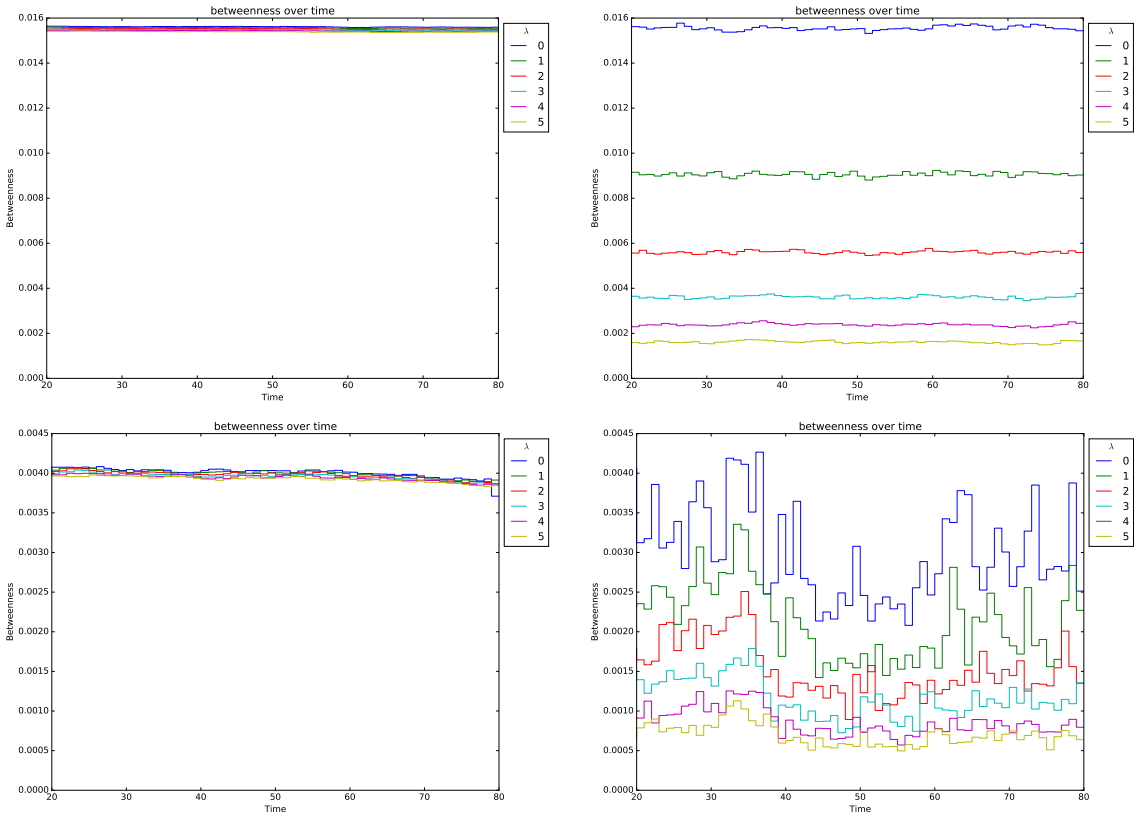
(b) $C^*_{AH}(unconstrained)$



(c) $C_B(unconstrained)$

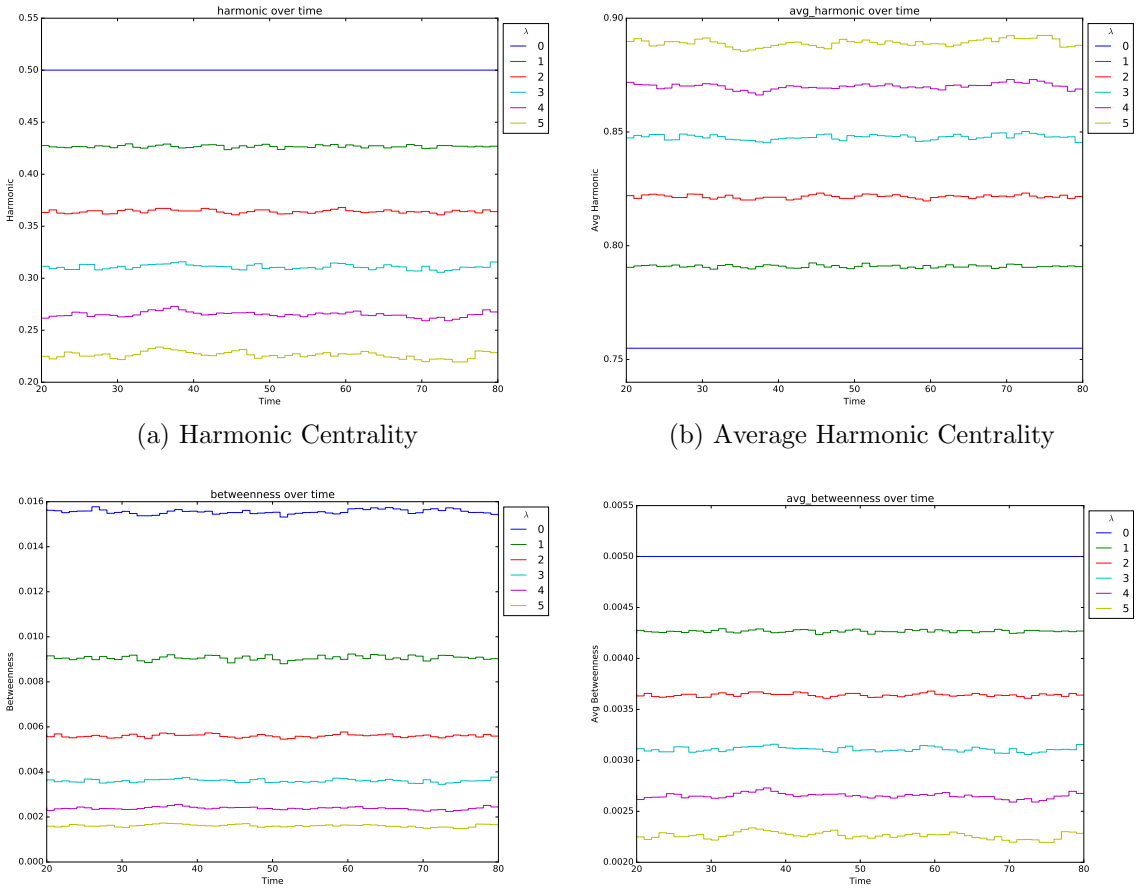
(d) $C^*_{AB}(unconstrained)$

Figure 3.16: Centrality values for an unconstrained network of 100 nodes where μ prescribes an unexpected occurrence and $\eta = 1, N, 2N, 3N$. Jitter produces a similar effect to the metrics as in Figure 3.13.



(a) $\eta = 1$ (b) $\eta = 2N$

Figure 3.17: Betweenness centrality over time of a constrained network (top) and unconstrained network (bottom), each with 100 nodes. Comparing the effect of changing λt size on the network with almost no jitter (a) and constant $2N$ jitter (b), using the Union sampling method.



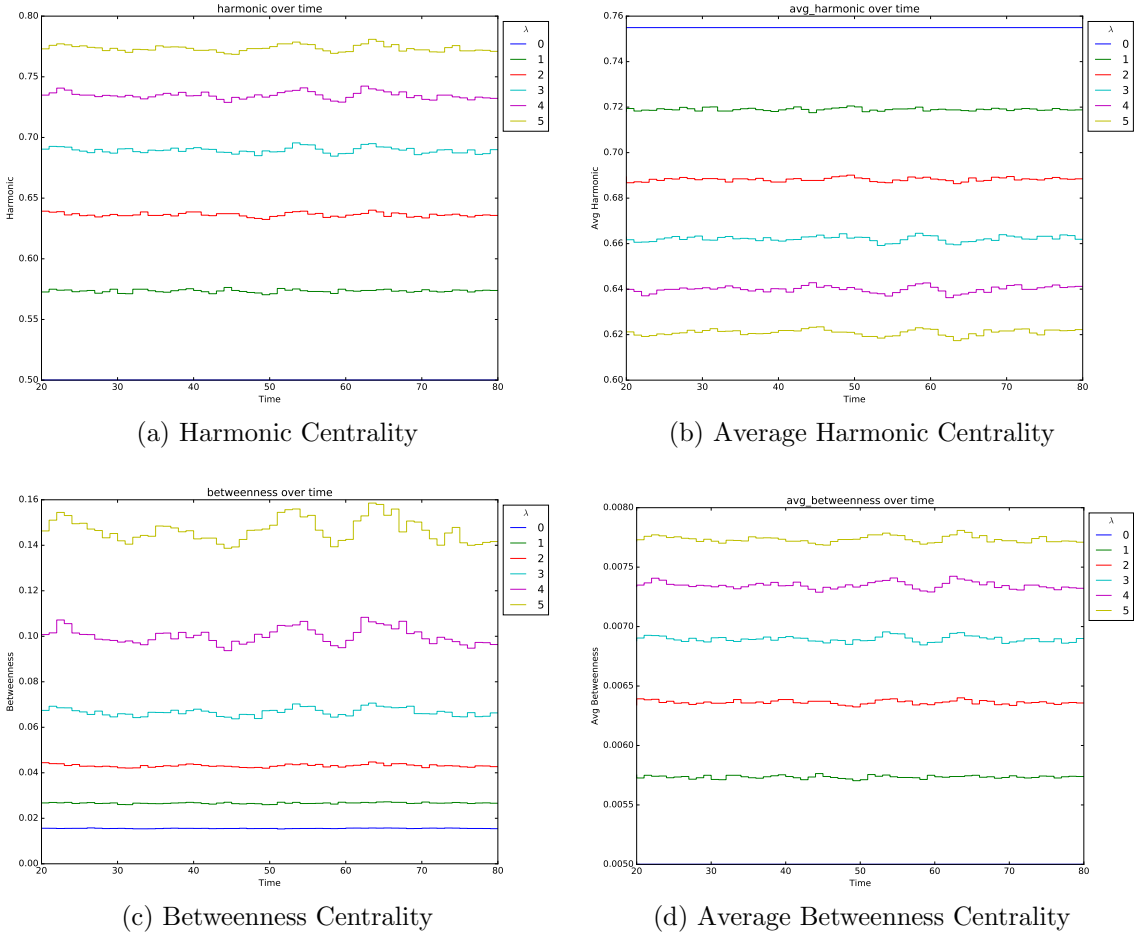
(a) Harmonic Centrality

(b) Average Harmonic Centrality

(c) Betweenness Centrality

(d) Average Betweenness Centrality

Figure 3.18: Centrality metrics with a constant $\eta = 2N$ jitter under the Union sampling and guaranteed minimal star. As the sampling interval size increases, more edges are included in the measured graph and the metric values match our theoretic proportionality.



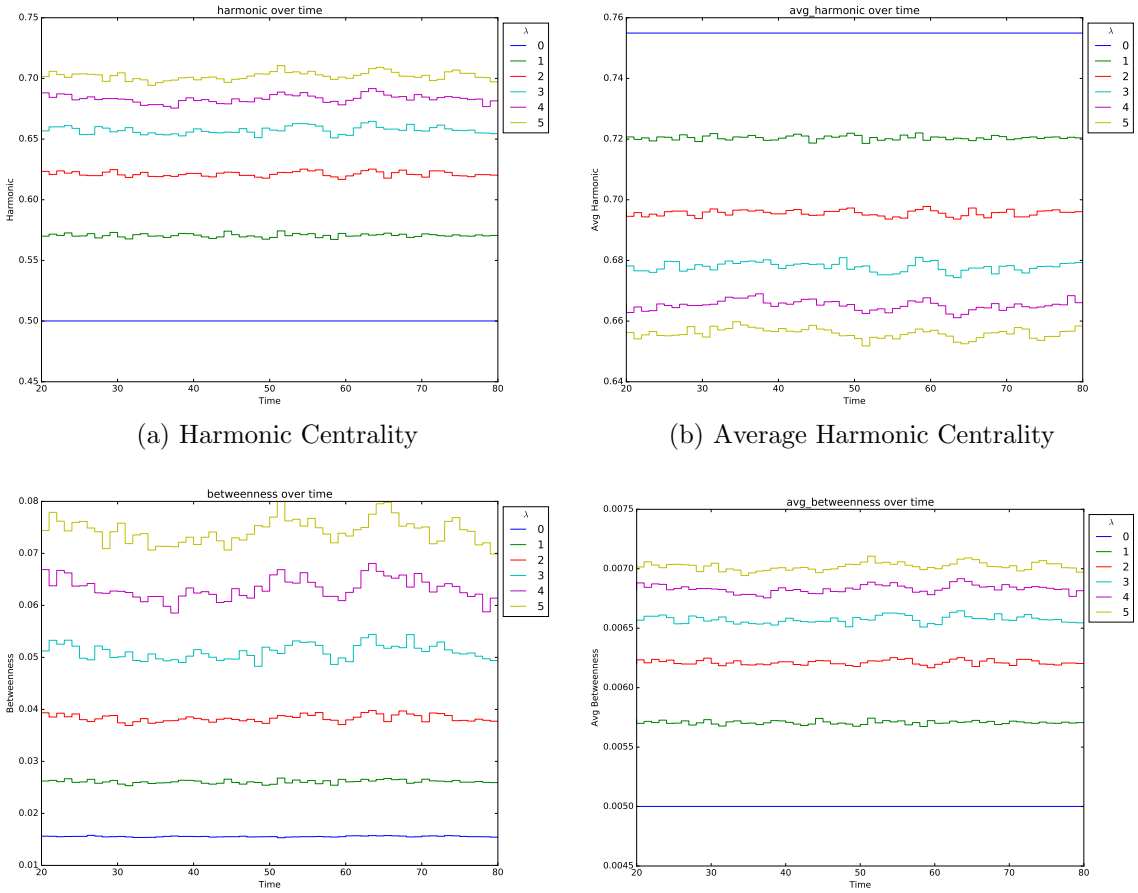
(a) Harmonic Centrality

(b) Average Harmonic Centrality

(c) Betweenness Centrality

(d) Average Betweenness Centrality

Figure 3.19: Centrality metrics with a constant $\eta = 2N$ jitter under the Intersection sampling and guaranteed minimal star. As the sampling interval size increases, less edges are included in the measured graph and the metric values evidence our theoretic claims.



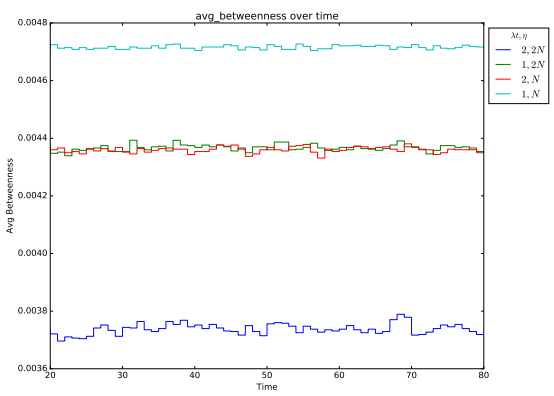
(a) Harmonic Centrality

(b) Average Harmonic Centrality

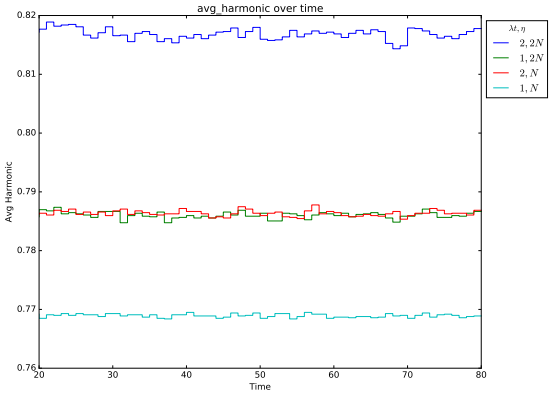
(c) Betweenness Centrality

(d) Average Betweenness Centrality

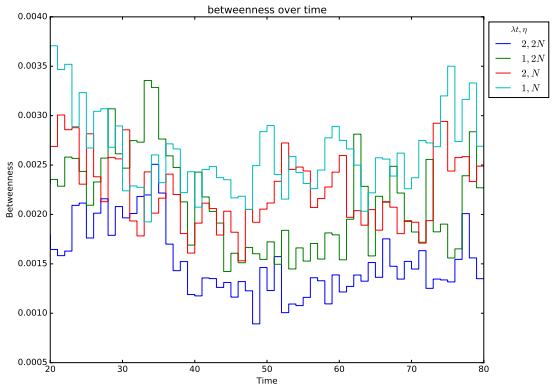
Figure 3.20: Centrality metrics with a constant $\eta = 2N$ jitter under the begin-end sampling and guaranteed minimal star. As the sampling interval size increases, less edges are included in the measured graph.



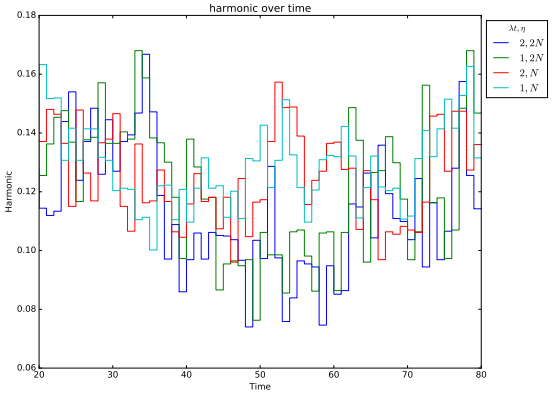
(a) Average Betweenness Centrality



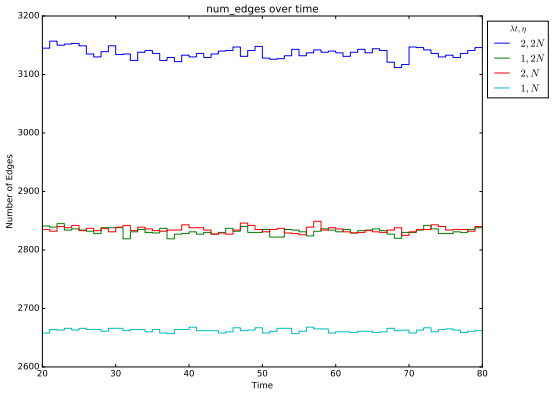
(b) Average Harmonic Centrality



(c) Betweenness Centrality

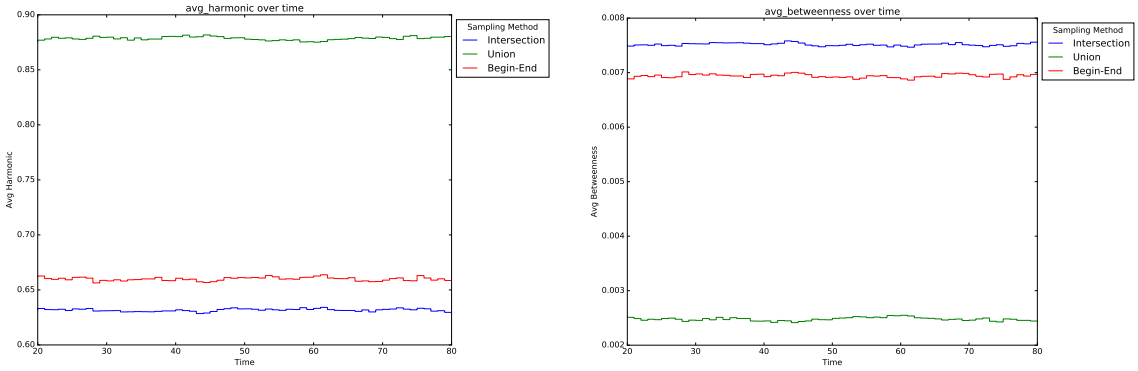


(d) Harmonic Centrality



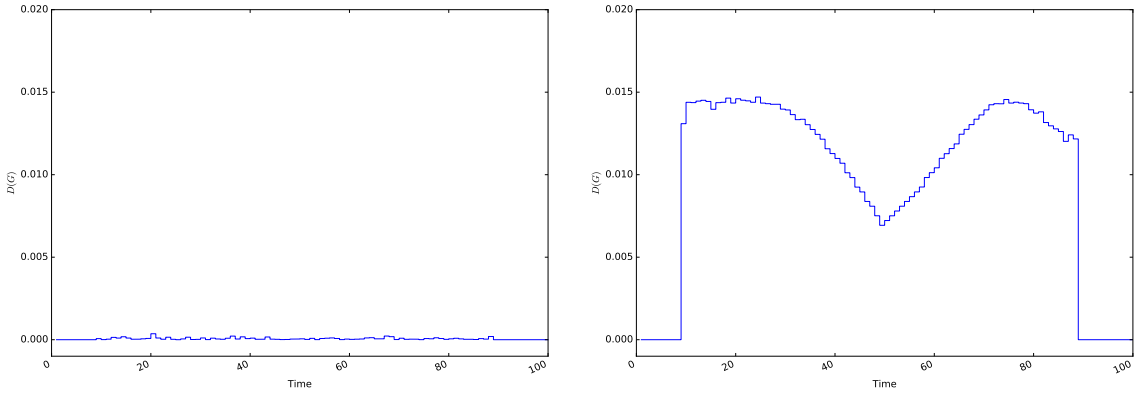
(e) Edge Count

Figure 3.21: Comparison of networks, one with $\eta = 2N$ and another with $\eta = N$ using two different sampling windows and the Union sampling. Each network is unconstrained, has 100 nodes, and $\mu = 1$. Aliasing may mask the additional network activity, corroborating Ribeiro's work with random walkers [1].



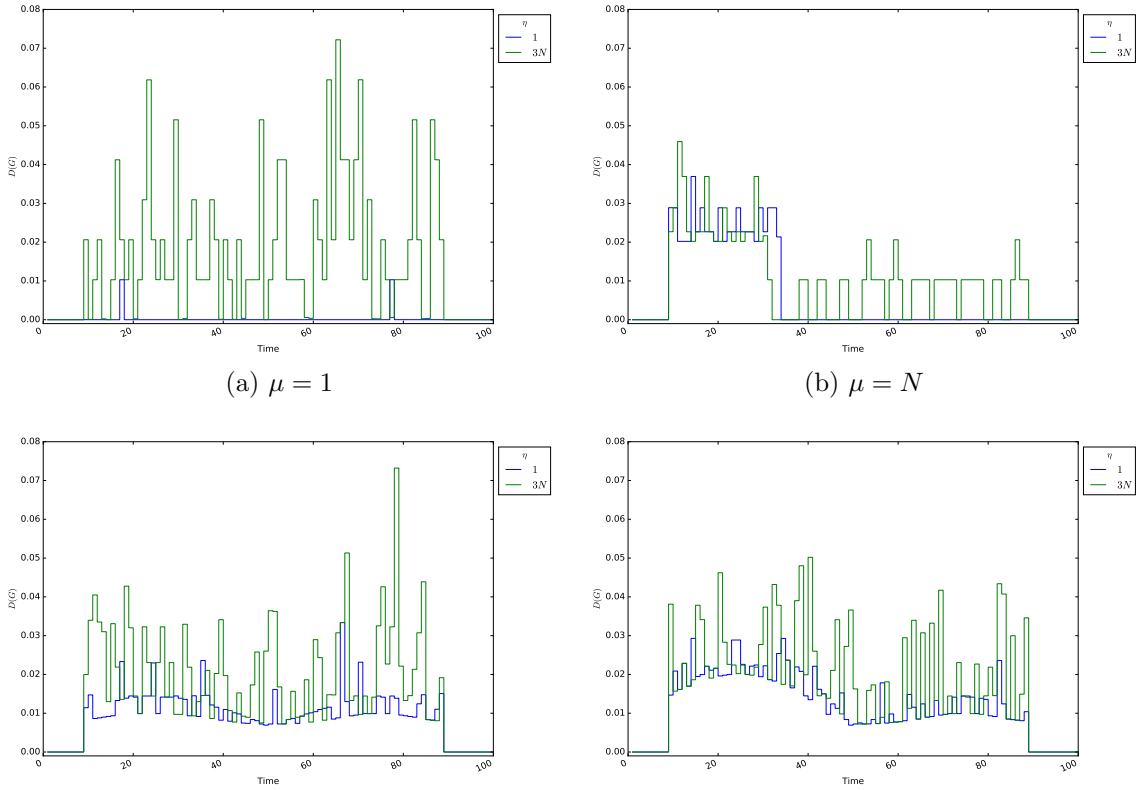
(a) Average Harmonic Centrality (b) Average Betweenness Centrality

Figure 3.22: Choice of sampling method produces different distributions of centrality metrics. This constrained network has $\gamma_0 = 0.5$, 100 nodes, and $\eta = 3N$ jitter sampled over a $\lambda t = 3$ sized window.



(a) $\mu = 1$ (b) $\mu =$ expected occurrence

Figure 3.23: The dynamic $D(\mathcal{G})$ profile of a network with no change in density (a) and a network with an expected occurrence centering on time $t = 50$ (b). Both networks contain a minimal star and consistent $3N$ jitter.



(a) $\mu = 1$ (b) $\mu = N$

(c) $\mu =$ expected occurrence (d) $\mu =$ unexpected occurrence

Figure 3.24: The dynamic $D(\mathcal{G})$ profile of networks with no change in density (a), networks with consistently increasing density until it becomes complete at $t = 35$ (b), and networks with an expected occurrence (c) and unexpected occurrence (d) centering on time $t = 50$, all under two different amounts of jitter. None of the networks guarantee a minimal star.

Chapter 4

Real-World Experiments

We apply the previous techniques from synthetic datasets onto three real-world examples, since the value of this research is in its effectiveness at capturing crucial aspects in their application-specific domains. Let us therefore examine these metrics and extensions in relation to the following motivating examples: the Nauvoo Marriage Project, the Social Networks and Archival Context project, and an ArXiv.org co-authorship network.

Each of these real-world experiments sheds light on core questions of their motivating application. They test the ability of our identity function, sampling, and metric approaches to capture the underlying processes taking place in these networks and to expose network dynamics. Our approaches highlight and confirm historic interpretations of the social development and organization present in each application and lead the domain-specific researchers both to new insights and new lines of inquiry.

4.1 Nauvoo Marriage Project

The Nauvoo Marriage Project, led by Dr. Flake and in conjunction with the *Institute for Advanced Technology in the Humanities*, seeks to investigate and understand the concept of "marriage" in early Mormonism and its impact on the societal and church structure of the day. In the mid-1800s, Nauvoo, Illinois, early Mormons began to re-define marriage in a way that is still not fully understood by scholars. Individuals participated in both polygynous and polyandrous marriages to create desired

family units, kinship structures, and religious lineages. Our dataset evidences multiple types of marital attachment: traditional civil marriages recognized by the government and religious *sealings*–including *eternal sealings* which were believed to persist past death, *temporal sealings* which were believed to last only during life (i.e., "until death do us part"), and posthumous eternal sealings performed by proxy. Each of these types of marriage has its own significance, but understanding their use and inter-connectedness among family units and across generations is key to uncovering the social constructs the early Mormons were creating.

Adding to the complexity of the dataset, the construction of the Mormon temple in 1845 played an integral role, as marriages and sealings were repeated in the temple. Parent-child relationships also gained religious significance during this time. Children born before the temple was built were "adopted" to their birth parents in the temple to connect them under this new kinship and religious lineage network. Continuing this trend forward, some mothers are sealed to their daughter's husbands, aunts are maritally attached to their nephews, and brothers or sisters are joined with their siblings' family units.

Our dataset consists of over 70,000 individuals with numerous polygynous and polyandrous marriages. Those marriages include 5,318 civil marriages, 2,923 eternal sealings, and 187 temporal sealings. One man, the founder Joseph Smith, has 93 marital events in our database. Another 64 men have ten or more marital attachments each and 3,123 have at least two. Two women in our dataset have 8 marital events, while 2,744 have at least two. These events include both polygamous marriages and serial marriages, such as a death followed by a remarriage.

To encode these marriages, family units, and individuals into an evolving network, we create the inverse, or line graph [37], of the standard family tree structure by encapsulating the marriage events into nodes and the people as edges. Since individuals may participate in multiple marriages at any point in time, having nodes that represent the marriages was a more intuitive connection point for those individuals. An alternative approach would be to utilize a bipartite graph, with nodes representing both people and marriages. However, our approach is equivalent to this case, since we are most interested, from an application standpoint, in characterizing the changes on the marriage nodes. Our approach also becomes increasingly important in depicting the marital and lineage structures in a meaningful and evocative way, as discussed in Chapter 5.

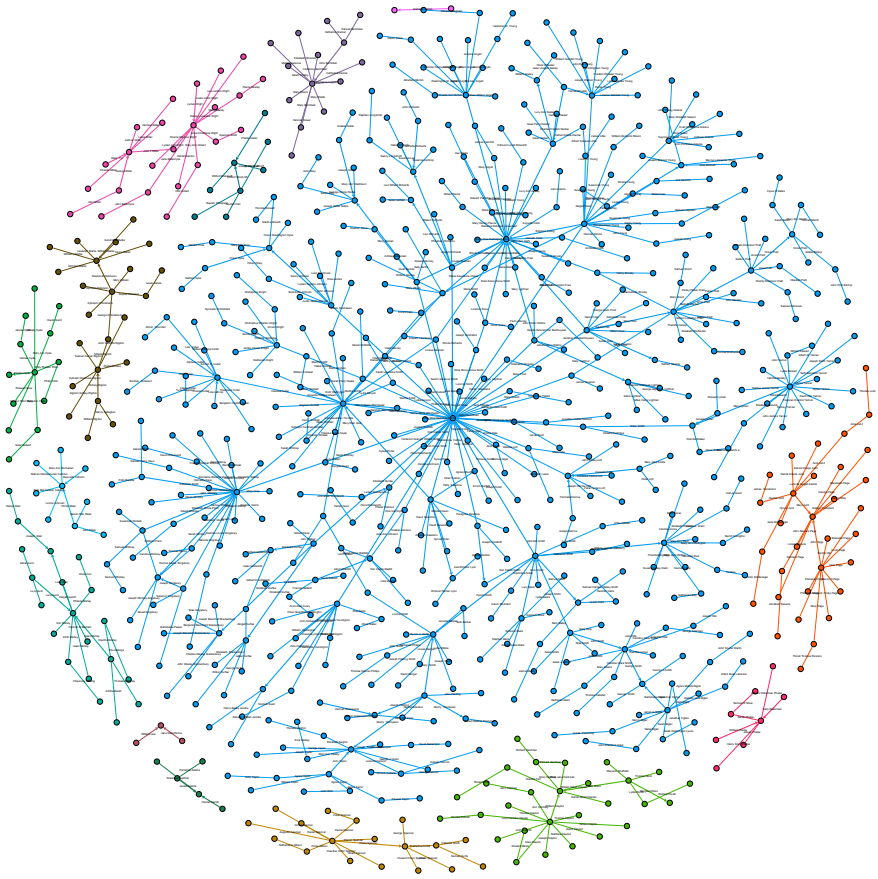
Without loss of generality, let us describe here the construction of a patriarchal network. The marriage nodes are constructed as the marriages centric to each male in the network. In this case,

the males are the *primary participants*s in their marriages, to whom all other participants will be collected. Their marital nodes are created when the primary participant is born and exist throughout time, however additional edges are not constructed until other individuals begin to participate in the marriage. Individuals are directed edges connecting their birth marriage, as an out-edge, to their marriage(s) as adults as an in-edge. For simplicity, one edge is created for each marriage a person participates in. Male edges, like the marital nodes, are created at the time of their birth; they only exist through their death date. Female edges appear at the time of their marriage and exist through either their adult-marriage divorce date or their death date, whichever comes first. This construction creates a caveat in the network that must be addressed: parental marriage nodes must be created for any individual to be included in the network even if their parents are not in the sample set. Likewise, for males, an adult marriage with no spouses or children must be created to include those person edges. A cumulative, non-temporal version of this network as of 1845 can be seen in Figure 4.1. Since marriages exist even after their primary participant dies, posthumous marriages of a living spouse to the person will still be captured in this design. Likewise, since the time window under examination is short, we treat eternal sealings and other marriages identically: the marriage persists after the participants are no longer alive. The only situation not covered in this model is a completely posthumous union, in which both participants have already passed away.

For this analysis, we consider three different identity classes and define three TIVGs based on marital focus: patriarchal marriages consisting of men and their connected wives; matriarchal marriages of women and their connected husbands; and a pairwise binary set of marriages in which a marriage consists of a pair of spouses: one man and one woman. The full dataset, therefore, contains the following features:

- $\mathcal{E}$, the set of marriages, divorces, births, deaths, and adoptions, with bursts of activity between 1844 and 1846,
- $|E| \approx 72,519$ individuals,
- $|V| \approx 20,534$ pairwise binary marriages,
- Three marital identity classes,
  - pairwise binary marriages, $|V_{pairwise}| \approx 20,534$,
  - matriarchal marriages, $|V_{matriarchal}| \approx 13,000$, and

Figure 4.1: Accumulated depiction of the patriarchal network of the Anointed Quorum individuals plus one degree of separation in 1845, consisting of all marriage events before December 10, 1845. Color coding denotes connected component, with the largest containing the polygamous marriages of Brigham Young and Newel Whitney.



– patriarchal marriages, $|V_{patriarchal}| \approx 17,000$.

4.1.1 Experimental Design

To analyze this network, we focused on a core set of individuals and expanded the proportion of the network under examination by increasing the degrees of separation from the focus set. The core group, important to Dr. Flake, is the Anointed Quorum: a group of individuals central to the formation of the church. Each increased "degree of separation" included an additional layer in the lineage structure: those individuals' parents and children.

We then performed the following analytical experiments over the Nauvoo marriage graph, examining each TIVG with λt granularities of ± 1 day, ± 1 month, ± 6 months, and $\pm 1, 3$, and 5 events as well

as instantaneous snapshots when $\lambda t = 0$. Each experiment tested metric variance across sampling windows and the effectiveness of dynamics analysis at uncovering key active points in time.

Experiment 10 (Nauvoo Marital Centrality)**.** We directly apply the C_H and C_B metrics individually to each of the identity classes: patriarchal, matriarchal, and pairwise binary. This experiment sought to address the following application questions: did marriages become more centrally connected over time, and did the dynamics of marital centrality change around key events in the Mormon church's history? Specifically, importance is placed on Joseph Smith's marriages; however, did another person, marriage, or event change that dynamic during the Nauvoo period? The dataset for this experiment will be the core group of approximately 800 individuals within 1 degree of separation around the Anointed Quorum, which is the most complete data from the Nauvoo Marriage Project.

Experiment 11 (Nauvoo Identity Comparison)**.** Continuing with Experiment 10, we compared the results of the metrics between the different identity classes to begin to understand if the people in that time were favoring patriarchal lineages over the other node identity classes.

Experiment 12 (Nauvoo Patriarchal Centrality with Adoptions)**.** An additional concern for Dr. Flake is the effect of adoption on the connectedness of the marital network. We therefore modify the patriarchal network to reroute individuals at the time of their adoption to "new" parents.

Experiment 13 (Nauvoo Scalability)**.** The first two Nauvoo experiments test the ability of the metrics to answer application questions related to the Nauvoo dataset, requiring better vetted data. To test the scale of our approach, we repeat Experiments 10 and 11 over a larger subset of the data including 10 degrees of separation from our selected individuals.

4.1.2 Results

For most of the above experiments, the analysis tool is capable of answering the posed questions. Experiment 10 attempted to evidence the changing centrality and cohesiveness of the network under each identity lens. Figure 4.2 displays the graph-wide harmonic and betweenness centralities over time for each TIVG. Since these measures only depict the disparity in the network between the most highly central marriage node and all other nodes, the average node-centric measure is important to convey the overall centrality of nodes. Average centrality measures for each identity lens are depicted in Figure 4.3. At a high level, we see a sudden decrease in centrality at the time of Joseph Smith's death on June 27, 1844. Similarly, we can see the completion time of the Nauvoo temple in early

1846 and the rush of marriages that occurred in the few months before the Mormons began leaving Nauvoo for the west.

An examination of the nodes with maximal centrality confirm that Joseph Smith's patriarchal marriage was of highest importance throughout the entirety of this time. His death in 1844 led to a decline in the cohesiveness of the networks as seen across all identity lenses, however additional posthumous marriages over the next few years continued to increase and strengthen his marriage's importance in the patriarchal network. Under the patriarchal lens, Joseph Smith's edge in the network provided a direct connection between his father's family, which included the polygamous unions of most of Smith's brothers as well as his uncle John Smith's polygamous family unit, and Smith's own dense polygamous union. When Smith died, even though his marriage node persists, the direct connection between these families is broken leaving only a shared wife between Smith and his brother Don Carlos as connector, which reduces the centrality of each of the nodes. The loss of Joseph Smith was more significant for the pairwise binary and matriarchal networks, since the loss of his edges removes the connection between his mother's marriage to each of his adult marriages.

In the time between Smith's death and the temple, we see a restructuring of the church, after which Brigham Young and the Quorum of the Twelve take charge. This restructuring is evidenced in the patriarchal centrality plots through a rise in centrality in two periods: late 1844 and early 1845. Another factor in the increased centrality in that time was men beginning to marry into polygamous unions and take Smith's wives into "caretaker marriages:" marriages designed to economically and socially sustain the woman for the rest of her life[1].

Figure 4.4 depicts each identity lens on the same plot for comparison in Experiment 11. It is important for us to note that the average centralities and disparity-highlighting graph-wide centrality measures both show the dominance of the patriarchal network over the other two lenses. Through harmonic centrality in the patriarchal network, we see an increase in important nodes, such as Joseph Smith's posthumous marriage and Brigham Young and Newel K. Whitney's marriages, compared with the other patriarchal marriages at the time. This results in a higher graph-wide harmonic centrality measure. The matriarchal and pairwise networks' loss of Joseph Smith's connections reduces the disparity between his mother Lucy Mack's highly connected marriage node, through her son, and other marriages. Following that loss, in August 1844 Brigham Young's mother Abigail Howe becomes

[1]For clarification, and due to the debate of sexuality in these marriages, it should not be inferred that these were sex-less unions. The most obvious example is the Mormon application of the Levitical law regarding brothers of the deceased marrying the widows of their brother to ensure his progeny.

the most central node through her son's marriages, but she is unable to surpass Mack's level of importance by the time of the temple. The patriarchal network, on the other hand, fairly quickly exhibits increased centrality after the loss of the Joseph Smith edge, highlighting an apparent desire to form a strong patriarchal system.

Adoptions, Dr. Flake speculated, would enhance the patriarchal system by connecting individuals and their spouses more directly into the lineages. We find that indeed is the case. In the underlying network, we affect an adoption by redirecting the source of a person's edge from the marriage into which they were born to the marriage of their adoption. This effectively changes the parents at that time, depicting the person as having been "born" from their adopted parents instead. Figure 4.5 plots the centralities of the patriarchal network under this notion of adoption compared with the original biological-only network.

On first impression, it appears that the adoptions beginning in 1846 reduced the overall centrality of the network. The adoptions increase the average centralities (C^*_{AH} and C^*_{AB}) of the network briefly in January 1846, but the marriage events in the network without adoptions produce a more highly central graph shortly afterwards. However, based on the underlying data, the plots here are misleading and evidence even more strongly the inter-workings of the individuals at the time. An examination of the network confirms that this drop is caused by adoptions serving as a reinforcement of the lineages already in place with the marriages: adoptions connect males into lineages already connected by their wives. For example, Joseph Kingsbury, Loenzo Pond, Mahala Dorcas, and Dorcas Kingsbury are all adopted to Newel Whitney. However, there's already a connection between Newel Whitney's marriage and Joseph Kingsbury's marriage: Sarah Whitney. So from the network standpoint, the adoptions reinforce Sarah Whitney's already existing connection and drops the connection between Joseph Kingsbury's marriage and that of his biological parents. C_H, depicted in Figure 4.5a, confirms the reinforcement by depicting an increasing disparity between the highly connected nodes and less connected nodes, such as those marriages which "lost" children to adoption by the core marriages. This outcome is significant, since it evidences the reinforcing and realigning of the lineages to this core group of early church fathers.

After understanding how the measures directly depict changes in the underlying networks, let us apply the two methods utilized in the synthetic analysis to gain a better picture of the dynamics in this dataset. Our initial attempt utilizing a comparison of the metric distributions across different time windows to showcase the dynamics does not appear as effective as in the synthetically generated

datasets. This is to be expected, since that method provides stronger evidence in periods of sustained change. Figure 4.6 displays the change in metric distribution across 3 and 5 events on the patriarchal network. Overall, the vertical shift we saw in the synthetic data is limited to the sampling windows encompassing the timepoints when the metrics exhibited significant change, such as Joseph Smith's death date. This vertical shift appears as a horizontal shift in the metric values, extending the maximal and minimal values around the change throughout the size of the λt window size. When the timepoint of change in the metric value passes out of view of the window, the change is taken into effect in the larger-window sampled graph. Using this method, we would therefore get a granular picture of the dynamics in the network.

Therefore, we compute $D(\mathcal{G})$ over each network to produce the profile plots in Figure 4.7. Most of the dynamics are low-level noise affecting only a fraction of the network. However, choosing a threshold value of 0.015, we obtain the set of highly-dynamic points in the networks seen in Tables 4.1, 4.2, and 4.3. These dynamic points provided Dr. Flake with an additional set of dates to examine, even though most are well-known events and shared across identity lenses. July 28, 1844, was the reaction to Joseph Smith's death. January 15, 1846, Dr. Flake speculates, is the re-sealing of Joseph Smith's widows to caretaker marriages. Adoptions started on January 25th of that year and a large number of temple sealings took place late that month and into early February.

The interesting highly-dynamic date is May 1, 1843; it appeared to be the outlier in our dynamics analysis, yet it provided one of the most important leads for our researcher. Originally, Dr. Flake speculated that this was around the time Emma Smith gave assent for her husband Joseph Smith to take additional wives, which was inconclusive in our data. Upon examination of the underlying network, we found that one important marriage event happened in May 1843: Joseph Smith and Helen Kimball were sealed for eternity. The May 1 date was implied in our analysis, due to the approximate nature of our data—lacking a specific day—however, this marriage connected the three most important lineages in the church at the time: Joseph Smith, Heber Kimball, and Brigham Young. Further investigation evidences a story of connection: Heber Kimball and Brigham Young were already connected by Young's sister, Fanny Young, who had married Kimball's father-in-law in 1832. In 1843, they had only two daughters of age to marry: Kimball's daughter Helen who was 14 years old, and Elizabeth Young, Young's 18-year-old daughter. Elizabeth, however, had already married Edmund Ellsworth the year before, who had recently been given a priestly office, and by May 1843 they were 7 months pregnant expecting their first daughter. Therefore, it is possible to hypothesize that Helen was chosen instead of Elizabeth. Additional scholarly research has begun on

Table 4.1: $D(\mathcal{G})$ values above the 0.015 threshold and its components for the patriarchal identity lens.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
1843-05-01	0.0182471	0.0147157	0.0107891	0.0000030
1844-06-28	0.0284042	-0.0204175	-0.0197465	-0.0000039
1846-01-15	0.0316787	0.0172641	0.0265611	0.0000247
1846-01-26	0.0278019	0.0055261	0.0272472	0.0000209
1846-01-27	0.0294976	0.0136052	0.0261726	0.0000133
1846-02-08	0.0181405	0.0043985	0.0175992	0.0000038

Table 4.2: $D(\mathcal{G})$ values above the 0.015 threshold and its components for the matriarchal identity lens.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
1844-06-28	0.1498526	-0.1414256	-0.0495441	-0.0000323
1846-01-15	0.0152299	0.0131969	0.0076020	0.0000071
1846-01-24	0.0183734	0.0096048	0.0156630	0.0000039
1846-01-28	0.0180701	0.0097912	0.0151875	0.0000039

Table 4.3: $D(\mathcal{G})$ values above the 0.015 threshold and its components for the pairwise binary identity lens.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
1844-06-28	0.1624156	-0.1045419	-0.1242974	-0.0000458
1846-02-03	0.0380062	0.0071122	0.0373348	0.0000137

this story because our dynamic analysis of the patriarchal marital network pointed the researcher to the events of the day and our conceptualization of the network allowed her to follow the lineages and discover this connection between the Smith, Young, and Kimball lineages.

As a future extension to this work, based on the successes of our dynamic points evidencing important events in the network domain and since our threshold values were based on heuristics, Dr. Flake would like to relax that threshold and investigate some of the lower dynamic points. By following this trail, examining each point in descending dynamic $D(\mathcal{G})$ value, we may reach a better threshold on which to base future analyses of networks similar to the Nauvoo dataset.

4.1.3 Discussion

Through our examination of the analysis results from Experiments 10-12, we were able to address the proposed questions for each experiment. The computation of the dynamics profile, evidencing the highly dynamic timepoints, was useful in pointing Dr. Flake towards interesting points in time for further inspection and inquiry into the historic texts. Aside from the usefulness of our analytic models, this marital-node mindset provided a foundation for useful new visualization techniques of lineage and marriage networks, in which the people are depicted as edges and the marriages as

nodes. They provide an evocative alternative to family tree diagrams, which we will discuss further in Chapter 5.

Overall, Dr. Flake found the techniques useful and insightful. They pointed her at previously undiscovered stories such as the connection between Young, Kimball, and Smith. They also provided a new way to conceptualize and depict the societal structure at the time. Previous work has been person-centric, focusing on the individuals of the time. By giving the marriages themselves the place of prominence, that of the primary element in the network, the nodes, this conceptualization—the shift to the line graph of the traditional conceptualization—bestows the place of significance to the sealings themselves. Dr. Flake's research attempts to understand polygamy, not the polygamists. Current research has put so much focus on the polygamists that they have obscured the meaning of polygamy; the significance of the sealing has been previously overlooked. Our techniques have centered the discussion on polygamy and the societal structure itself.

Regarding Experiment 13, our approach was able to scale to the larger network. Overall, the analysis over the larger network evidences a similar behavior to the smaller core group of individuals, but the inclusion of poorly connected nodes reduces the centrality values significantly. With each degree of separation included, we add additional individuals that have only been loosely vetted by historians, watering down the effects witnessed in the better-vetted core set. Much of the fringe data have yet to be verified and do not include specific birth, death, and marriage dates for individuals. This leads to assumptions in the analysis tool, i.e., the individuals and marriages must exist over the entire interval, which leads to inconclusive results. We leave the full discussion of the larger dataset, including plots and results, to Appendix F.1.

4.2 Social Networks and Archival Context Project

The Social Networks and Archival Context (SNAC) project was started in 2010 with the purpose of collating metadata on persons, corporate bodies, and families, and connecting those identities through their archival and bibliographic records found in repositories across the world. That process resulted in a documentary-social network, as identities are connected through a shared participation in documents and artifacts. Early in the project, SNAC partnered with archives in the United States, France, and the United Kingdom, as well as WorldCat, to process finding aids—archival

metadata records—and MARC[2] cataloging records. The project team extracted millions of names from these records, then matched and merged them into 3.7 million EAC-CPF[3] XML records describing identities: persons, corporate bodies, and families. The EAC-CPF records retain links back to the original archival materials as well as social links among the described identities. In totality, the network contained approximately 7.9 million relationships between identities and 8.3 million documents.

Due to SNAC's construction and the imperfect process of collecting identities from the original records, we extracted a clean subset on which to perform temporal analysis. We included all identities that had complete temporal data, such as persons with both a birth and death date. Even though SNAC's primary purpose is archival data, we found that it also contains a large subset of individuals who are still alive. Therefore, we interjected an assumption into the network to produce a better depiction over the entirety of the SNAC lifespan: since any individual born after 1920 could feasibly still be alive in 2010, the start date of the SNAC project, in the absence of a death date we assume that they died in 2015. This assumption significantly increased the number of individuals included in the network in the later portion of its lifespan. Secondly, we had to make an additional assumption about relationships between identities. Since our social connections lacked time sequences, we connected identities in the SNAC TVG for the entire intersection of their individual lifespans. For example, Thomas Jefferson and George Washington are considered associated from April 1743, Jefferson's birth date, to December 1799, Washington's death date. It is highly unlikely that Washington knew Jefferson at his birth; likely, they would have associated later in life. Therefore, this construction produces temporally coarse-grained, but evolving, relationships.

After our filtering and assumptions, we created a network with the following properties:

- $\mathcal{E}$, the set of $89,324$ birth and death events,
- $|V| = 558,527$ identities (persons, corporate bodies, and families),
- $|E| = 1,151,932$ relations between entities, and
- $T = [1600..2010] = 410$ years, the lifespan of the network.

[2]MAchine-Readable Cataloging
[3]Encoded Archival Context–Corporate bodies, Persons, and Families

In totality, there were 95,306 unique events in the entirety of the extracted network. Those events fell between 1 CE and 2014 CE. Eliminating those events before and after our analysis lifespan, T, we are left with 89,324 relevant unique events.

4.2.1 Experimental Design

SNAC is the most typical social network we examine using our approaches, even though the social nature of the connections is very broad. To be precise, in the overall dataset, some relations define an "associated with" connection that may relate a 20th century collector with an 18th century historical figure. However, since we limit relations temporally to overlapping life dates of their participants, the network exhibits only those social connections of contemporaries. Due to the nature of the network, we propose an experiment asking typical social network questions.

Experiment 14 (SNAC Centrality)**.** The SNAC dataset primarily tests the robustness and scalability of the sampling method and metric approach. It contains the largest number of events, defined by birth and death dates of individuals, and therefore is the network with the longest lifespan among our motivating applications. We will analyze the evolving network under C_H and C_B to answer the following application questions: How does the network change over time? Are the individuals represented in the dataset well connected over time, or are there periods of low centrality? In this case, low centrality may be an indicator of less-represented or less-described time periods in the archival collections.

4.2.2 Results

The SNAC dataset proved to be a challenge for the analytics tools due to its size. While the data structure was capable of storing the large amount of data due to the sparse nature of the graph, the analysis took a considerable amount of time. We therefore limit ourselves to Δt-time strides of 10 years or more to analyze the network throughout its over 400-year lifespan.

Figure 4.11 displays the centrality measures over the lifespan of the network at 10-year Δt samplings for sampling windows of 0, 10, and 20 years around each timepoint. These samplings are performed with the Union sampling method; if a node or edge exists at any point during the window, it is included in the sampled graph for measurement. Focusing on average harmonic centrality in Figure 4.11b, we see a sharp rise in centrality beginning in the early 1700s and a sharp decline in the

Table 4.4: $D(\mathcal{G})$ values above the 0.015 threshold and its components across time in the SNAC network.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
1710-01-01	0.0283797	0.0279264	0.0050519	0.0000019
1720-01-01	0.0266933	0.0264140	0.0038508	0.0000014
1730-01-01	0.0202645	0.0199296	0.0036687	-0.0000013
1740-01-01	0.0171492	0.0169741	0.0024443	0.0000006
1750-01-01	0.0739329	0.0715276	0.0187051	0.0000034
1770-01-01	0.0152875	-0.0137824	-0.0066146	-0.0000077
1790-01-01	0.0168337	-0.0152515	-0.0071251	-0.0000043
1960-01-01	0.0155100	-0.0154963	-0.0006522	-0.0000012
1970-01-01	0.0151710	-0.0151551	-0.0006934	-0.0000009
1980-01-01	0.0226061	-0.0226003	-0.0005159	-0.0000009
2000-01-01	0.0190049	-0.0190049	-0.0000445	-0.0000006
2010-01-01	0.0235548	-0.0235531	-0.0002842	-0.0000007

late 1900s. We postulate here that the bulk of SNAC's collected data is from archives in the United States, and therefore we see a larger average centrality measure starting around the formation of the nation and lasting until the mid-1900s at the peak of network size.

The density measure, Figure 4.11e, shows that our network is very sparse, with less than 1% of the edges present. In fact, our experimental design shows us that only 0.0003693% of the possible edges in the TVG exist at all during its lifespan. Due to this sparseness, our method for detecting dynamics by comparing distributions across sampling window sizes is not applicable. In Figure 4.11e, we can see that measuring with a larger λt sampling window produces a less dense graph even as Figure 4.12 shows an increase in network size, as larger sections of the disconnected network are included.

Therefore, we must turn to our derivative measure of dynamics to highlight interesting points and produce a profile of the network dynamics. To preform these metrics, we examine the SNAC graph using the 10-year Δt stride with a sampling interval $\lambda t = 0$. Figure 4.13 shows the dynamics profiles, including $D(\mathcal{G})$, for SNAC, using the same threshold from our Nauvoo experiments results in the list of dynamic points in Table 4.4. The largest change in dynamics occurred between 1740 and 1750. We originally assumed this was due to the prominent figures in history, Thomas Jefferson and George Washington, becoming connected in 1743. However, upon further examination, we find that two prominent English authors, James Boswell and Samuel Johnson, are the cause of the spike in centrality. As data becomes more sparse in the late 1900s and early 2000s, we see another surge in dynamics, however all the component measures exhibit a drop in centrality, denoting a negative change in the network.

4.2.3 Discussion

From these results, we can answer our questions from Experiment 14. The network becomes more connected and central in the mid 1700s and maintains a consistent state through the 1800s. The 1900s see a decline in the network connectedness as the density decreases. We also see identities that are fairly well connected in the mid-1700s evidenced by the larger disparity in the betweenness centrality measure in Figure 4.11c.

The discovery of James Boswell's centrality in the mid-1700s led us to reexamine the premise of our original experiment. Specifically, Boswell is connected directly to 17.6% of the network in 1744, while historically important figures like George Washington are only directly connected with 3.5%. After discussions with colleagues at Harvard and Yale, whose contributions of descriptions to SNAC are among the most dense, we conclude that our measurements may tell us more about the collectors, donors, and institutions than the actual importance of the historical individuals in the SNAC network. Harvard has an extensive Theatre collection that includes both Boswell, Johnson, and other central individuals at the time, such as David Gerrick. It is highly likely, we speculate, that the degree to which these identities are described and detailed contributed to their central nature.

Our results may therefore be the cornerstone of a larger study examining archival collection practices and the focus of those collections. Archives face an overwhelming amount of records to accession their collections, but there is an economy of how much time, money, and effort each is willing to spend to describe the material in detail. There is also a high value placed on describing records of historical interest rather than a broader historical record. By analyzing the connections in their collections, researchers may uncover the under- and over-described materials and proscribe directions to target their future accessions.

4.3 ArXiv.org Co-Authorship Network

For our last real-world application, we turn to ArXiv.org. ArXiv is an online archive and distribution platform for a wide variety of scientific research papers. One of its larger uses is a method of preprinting manuscripts before they are submitted for journal publication. The service was started as `xxx.lanl.gov` in 1991 and is currently maintained at `arxiv.org` by the Cornell University Library.

ArXiv allows bulk access to metadata and LaTeX paper submissions, allowing researchers to aggregate and analyze all papers in their collection.

We chose to combine the Computational Complexity, Discrete Mathematics, Data Structures and Algorithms, and Logic in Computer Science sections of ArXiv's Computer Science category to analyze as one network. These sections consisted of $21,148$ papers published between 1990 and 2015. Since ArXiv is based on user submissions, where names of authors and institutions are entered manually without requirements on a prescribed naming convention, ArXiv is the least vetted dataset we consider for analysis. We normalize the dataset by combining, through string matching, unique author names as well as institutional affiliation names to arrive at a final ArXiv dataset with the following properties:

- $\mathcal{E}$, the set of publications with publication dates,
- $|V| = 16,919$ individual authors,
- $|E| = 34,677$ co-authorships,
- Identity classes including:
  - individual authors, $|V_{author}| = 16,919$,
  - institutions and corporations, $|V_{institution}| = 1,505$, and
- $T = [1990..2015] = 25$ years, the lifespan of the network.

Since a publication set only details publication dates without capturing the collaboration involved in submissions, we make additional assumptions on the network. First, we define a collaboration period of 2 months before the publication date in ArXiv. Secondly, if collaboration periods overlap between two coauthors, we consider them to have collaborated throughout the entire time. Lastly, since authors and institutions may stop publishing at any point, due to a variety of reasons, we limit the lifespan of both author and institution nodes to the entirety of their collaboration event periods. That is, a node begins at its first collaboration period and exists through the end of its last collaboration period, ending on its final ArXiv submission date.

4.3.1 Overview

Experiment 15 (ArXiv Authorship Centrality)**.** In this experiment, we seek to uncover how well-connected authors and institutions are over time and if the patterns in co-authorship at the institution level mirror those of the individual authors. We apply the metrics to each of the identity classes, V_{author} and $V_{institution}$, for the ArXiv dataset.

4.3.2 Results

ArXiv presents a network that is much more dynamic and changing than either of our other two real-world applications. Figure 4.14a-b depict the harmonic centrality measures over time for both the author and institution identity lenses. In each lens, we see a highly dynamic centrality measure, with what appears to be a publication-year periodicity among the institutional harmonic centrality measure. The measures are more volatile in the early stages of the network due to its smaller size, as evidenced in Figure 4.14c-d. A few changes in a small network produce a larger change in the centrality and dynamics measures than similar changes in a larger network.

Comparing the identity lenses to answer our experimental questions, we look at the overall average harmonic centrality measures of Figure 4.14b. Before 1996, the network connectivity of the authors was higher than that of the institutions; author co-authorship played a more important role than institutions at that time. The early few years of the timeline connect only two authors, Donald Knuth and Arvind Raghunathan. Authors did not submit work coauthored with institutional affiliation until late 1995, when authors from LRI and CNRS coauthored a paper. From 1996, the institutional network became the more connected and central network. We see a spike in the institutional centrality in late 1999 as CWI becomes most central coauthoring with 7 other institutions in a well-connected network of 15 institutions, 22 institutional co-authorships, and only 3 connected components.

To examine dynamics, like our SNAC graph, we find that the approach of comparing distributions across varying sampling window sizes is also not as useful in the ArXiv network. Figure 4.15 shows the harmonic centrality and edge count over time for the author identity lens using 4 different sampling windows at each event in event-driven time: 0 days, 2 day, 30 days, and 2 months. As expected with the Union sampling method, as the λt window size increases, more edges and nodes are included in the sampled graphs. However, like SNAC, the increased sampling size does not produce

Table 4.5: $D(\mathcal{G})$ values above a 0.05 threshold and its components for the author identity lens.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
1994-03-20	0.2632317	-0.2327381	-0.0278183	-0.1197917
1994-09-21	0.0704901	0.0101181	0.0000000	-0.0697602
1998-03-24	0.0896949	0.0893543	-0.0004729	0.0077942
1999-07-24	0.0625785	0.0618878	0.0092673	0.0002792
2015-06-02	0.0951500	0.0680375	0.0000000	0.0665164
2015-06-03	0.1913665	-0.1551587	-0.0000000	-0.1120130

Table 4.6: $D(\mathcal{G})$ values above a 0.05 threshold and its components for the institutional identity lens.

t	$D(\mathcal{G})$	$\Delta C_H(\mathcal{G})$	$\Delta C_B(\mathcal{G})$	$\Delta C_D(\mathcal{G})$
1998-08-02	0.0522878	0.0201554	0.0475678	0.0080668
1998-10-26	0.0523975	0.0387437	0.0349743	-0.0046063
2002-01-10	0.1448145	-0.1435426	-0.0165945	-0.0095599
2008-09-12	0.0688699	0.0682915	0.0088891	0.0005581
2009-02-12	0.0698431	-0.0690900	-0.0102145	-0.0005408
2014-08-03	0.0547843	-0.0546899	0.0000442	-0.0032155
2015-04-15	0.0699230	-0.0696970	-0.0054451	0.0013824
2015-05-23	0.0859796	0.0601551	0.0000000	0.0614316
2015-05-27	0.0650398	-0.0513393	0.0000000	0.0399306
2015-05-28	0.3194444	0.0000000	-0.0000000	-0.3194444

the same effect we evidenced in our synthetic dataset, due to the inclusion of additional disconnected components.

Let us therefore examine $D(\mathcal{G})$ for our identity lenses. Since the ArXiv networks are significantly more dynamic than either SNAC or the Nauvoo Marriage Project datasets, we must set our threshold higher to eliminate the larger variations in the centrality measures. Figure 4.16 shows the $D(\mathcal{G})$ for both institutions and authors and Tables 4.5 and 4.6 list the dynamic points above our higher threshold of 0.05. Here we see a highly active early period for authors submitting to ArXiv without institutional affiliation. That gives way to more dynamic periods where institutions are co-authoring at a larger percentage than individual authors throughout the 2000s. We see an activity spike under both identity lenses at the end of our dataset's lifespan in mid-2015.

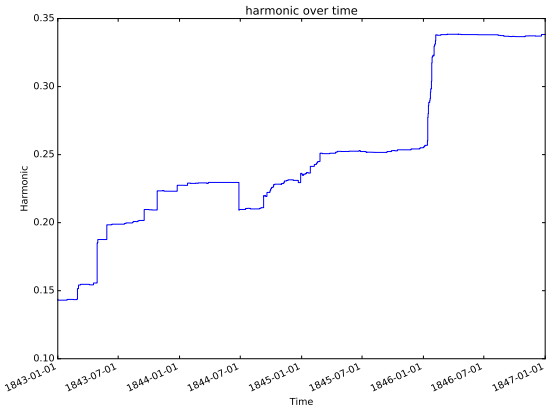
4.4 Real-World Summary

Each of our real-world datasets presented interesting challenges both for our researchers as well as the metrics themselves. Throughout our investigation, we noticed differences between the synthetic and real-world networks. Specifically, all of our real-world datasets are significantly less dense than the

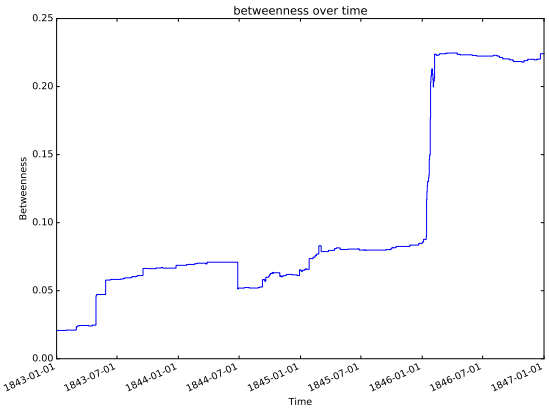
synthetic data. Our properties in Chapter 2, proven for connected networks as well as networks with a small diameter, do not empirically hold for the real-world examples due to their sparseness.

For future work, we note the importance of producing additional synthetic networks that not only mimic the network changes, such as our expected and unexpected event experiments, but also that incorporate the sparseness and disconnectedness typical of real-world networks. Most of our real world datasets have $|V| \approx |E|$, and for our main motivating application, the Nauvoo Marriage Project, $|V| > |E|$.

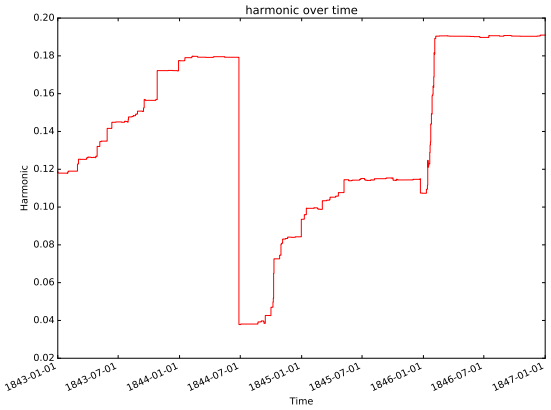
Figure 4.2: Harmonic and betweenness centralities across each of the identity classes for our core set of individuals. We see a marked increase in the centrality of the patriarchal network, while the matriarchal and pairwise binary networks depict a significant decrease in centrality around Joseph Smith's death in 1844. All networks show a rise after the temple was constructed in early 1846.



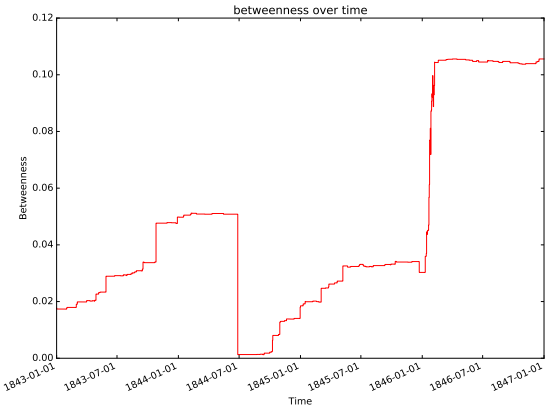
(a) $C_H(patriarchal)$



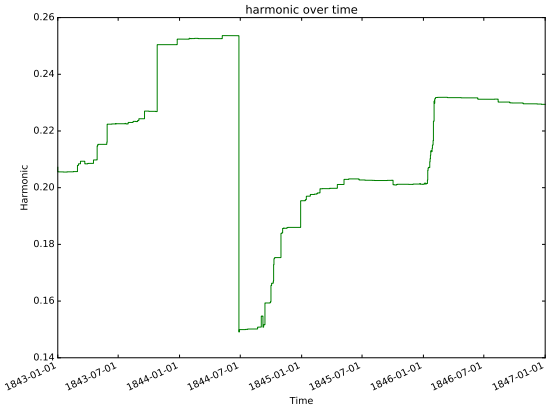
(b) $C_B(patriarchal)$



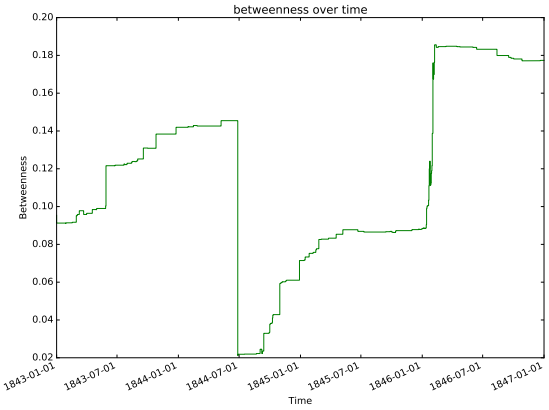
(c) $C_H(matriarchal)$



(d) $C_B(matriarchal)$

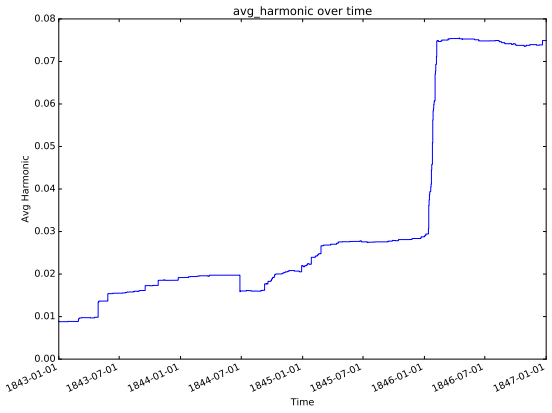


(e) $C_H(pairwise)$

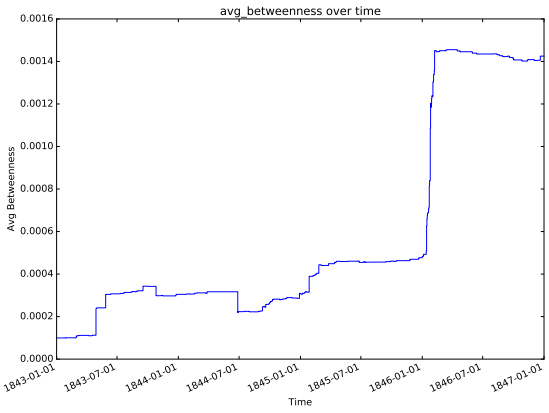


(f) $C_B(pairwise)$

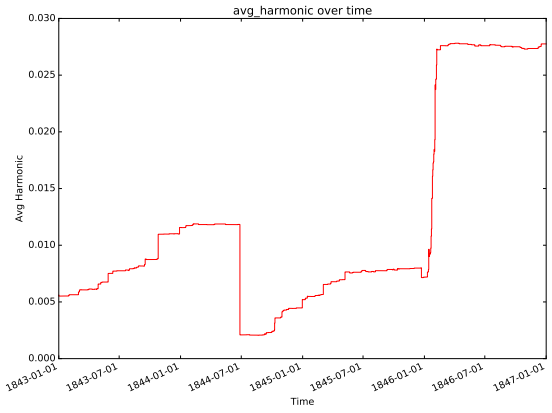
Figure 4.3: Average harmonic and betweenness centralities across each of the identity classes highlight a drop in centrality among the marriages at the time of Joseph Smith's death (June 27, 1844) followed by a significant increase as marriage sealings were performed upon creation of the temple in early 1846.



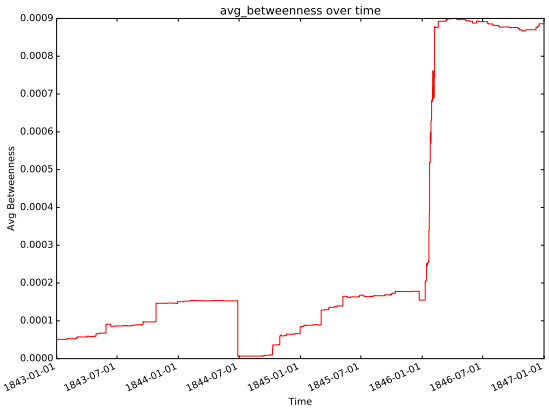
(a) $C^*_{AH}(patriarchal)$



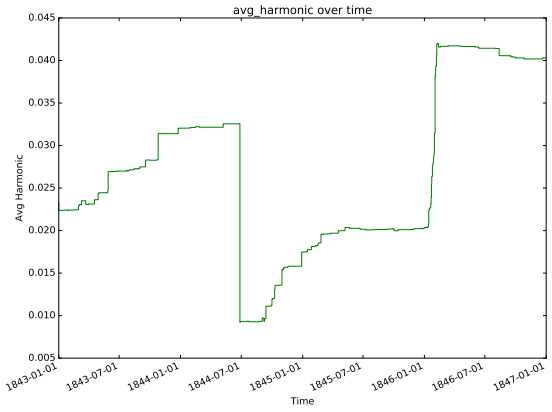
(b) $C^*_{AB}(patriarchal)$



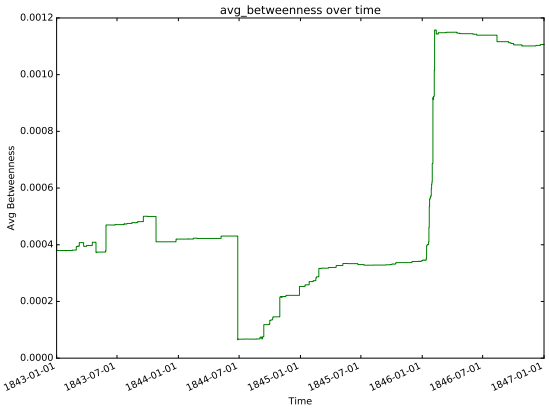
(c) $C^*_{AH}(matriarchal)$



(d) $C^*_{AB}(matriarchal)$

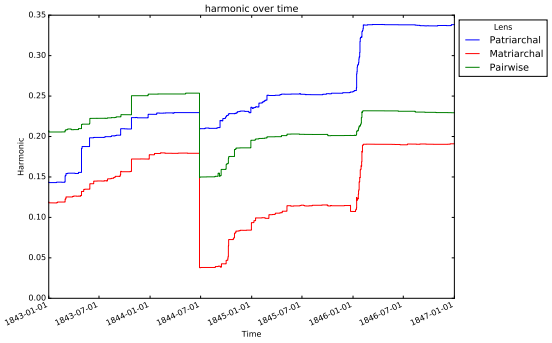


(e) $C^*_{AH}(pairwise)$

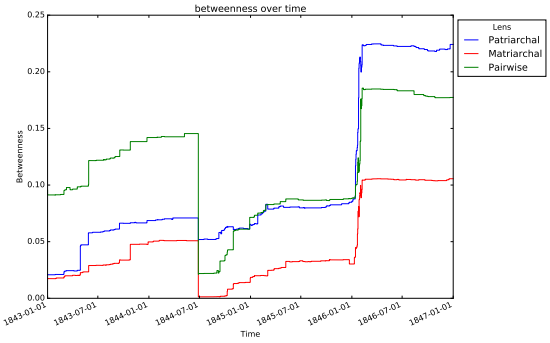


(f) $C^*_{AB}(pairwise)$

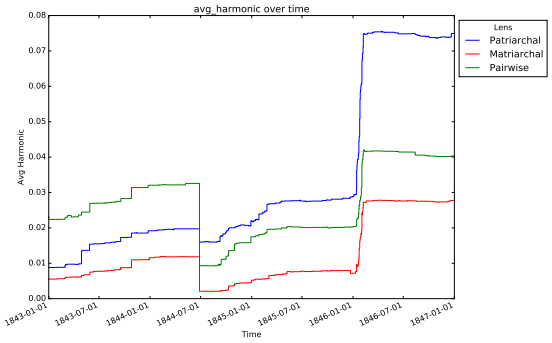
Figure 4.4: Plotting the metrics across identity lenses, we see a trend towards a more cohesive and central network with respect to the role of patriarchy.



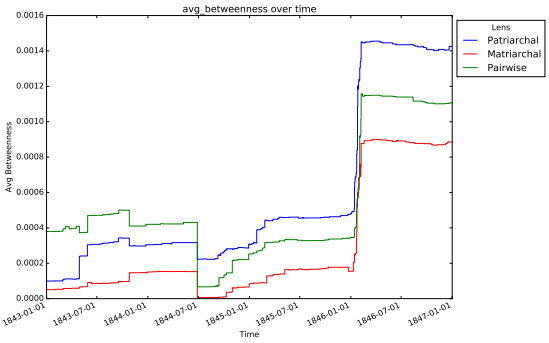
(a) C_H



(b) C_B

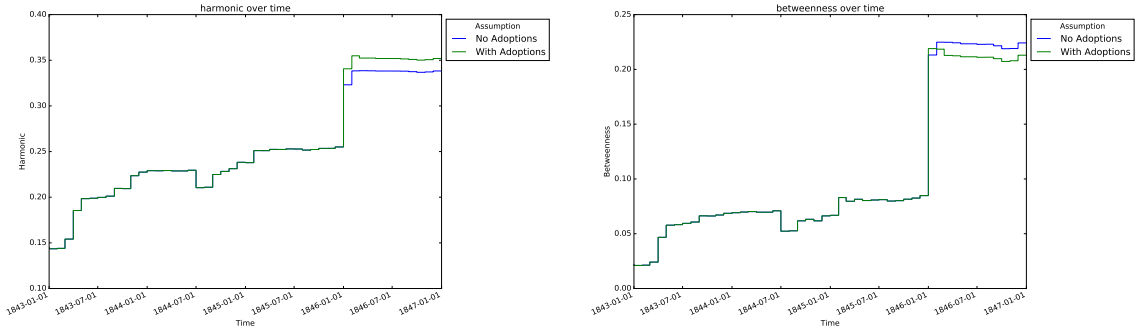


(a) C^*_{AH}



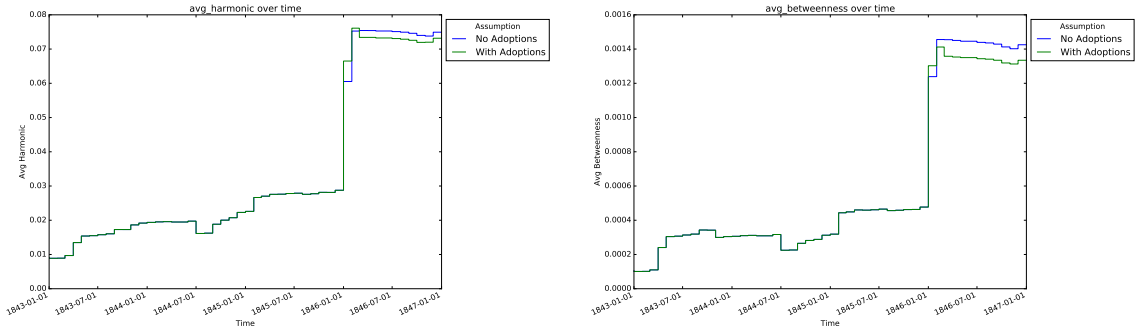
(b) C^*_{AB}

Figure 4.5: Adoptions, which started in 1846, reduce the overall centrality measures relative to the network without adoptions due to a reinforcement of already extant edges in the network.



(a) $C_H(patriarchal)$

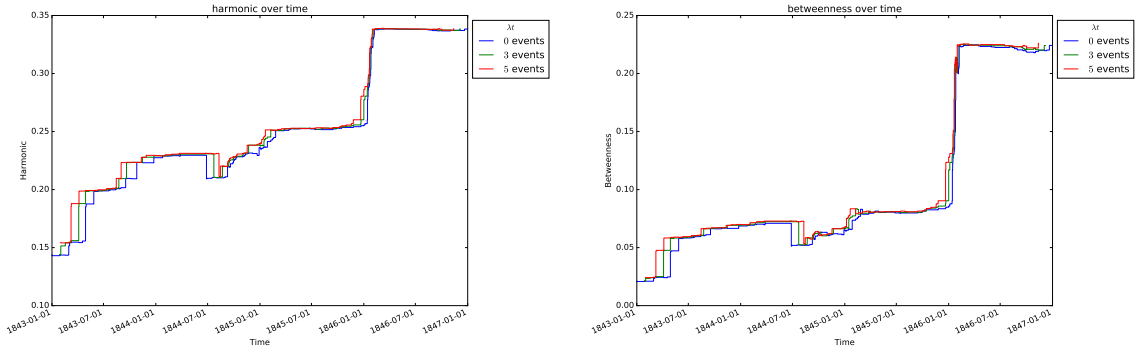
(b) $C_B(patriarchal)$



(c) $C^*_{AH}(patriarchal)$

(d) $C^*_{AB}(patriarchal)$

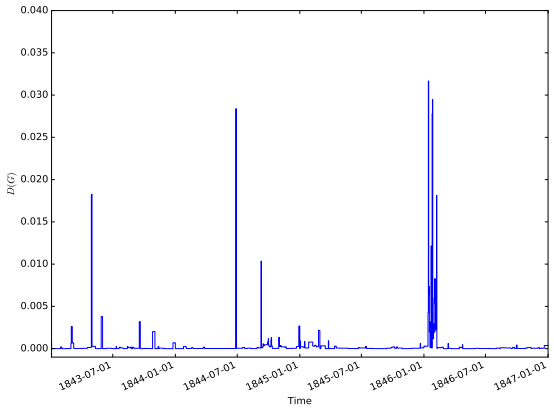
Figure 4.6: Varying λt in the real-world experiments does not produce as defined of an increase in the centrality measures as in our synthetic examples. Here we see evidences of change as the larger sampling windows pick up points of change sooner—λt events sooner—and holding on to them longer under the Union sampling.



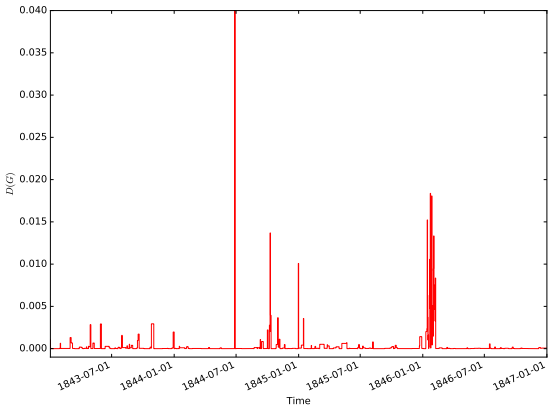
(a) $C_H(patriarchal)$

(b) $C_B(patriarchal)$

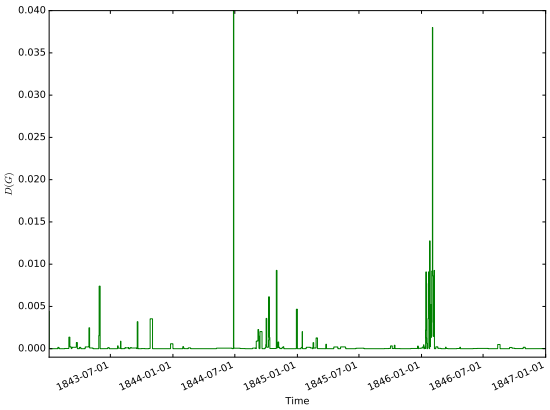
Figure 4.7: Comparing the dynamics measure $D(\mathcal{G})$ for all identity lenses over event-driven time. Here the matriarchal and pairwise binary values for Joseph Smith's death, evidenced on June 28, 1844, are truncated for readability; they are 0.15 and 0.16 respectively. Note: He died June 27, and therefore the drastic network change is shown on the next event date, June 28.



(a) Patriarchal

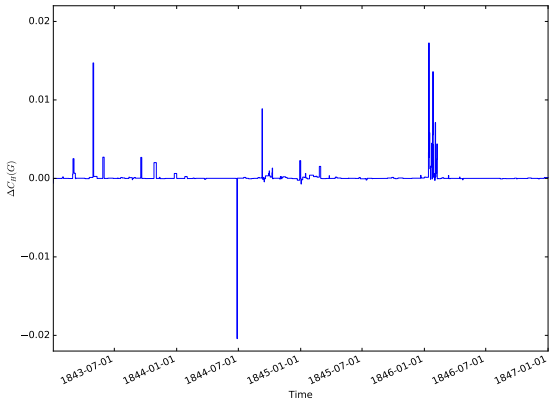


(b) Matriarchal

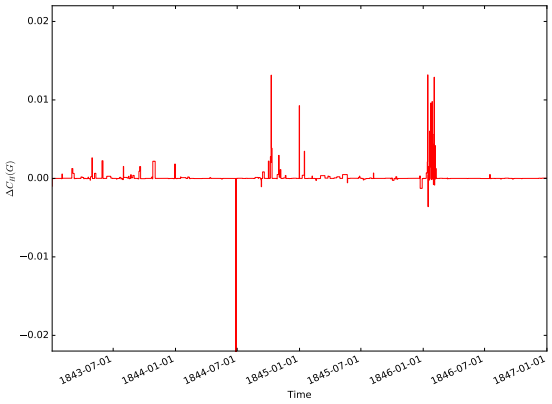


(c) Pairwise Binary

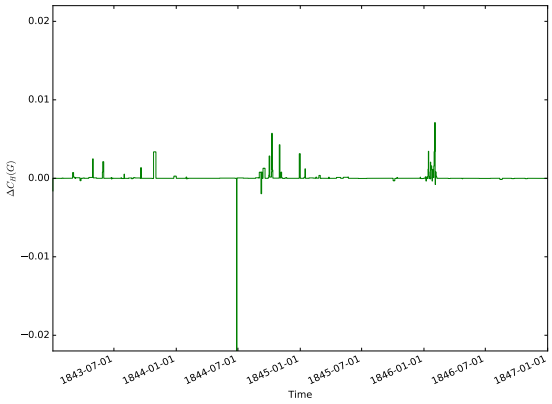
Figure 4.8: Comparing the dynamics measure $\Delta C_H(\mathcal{G})$ for all identity lenses over event-driven time evidences the large decrease at Smith's death in 1844 and an increase during the temple time.



(a) Patriarchal

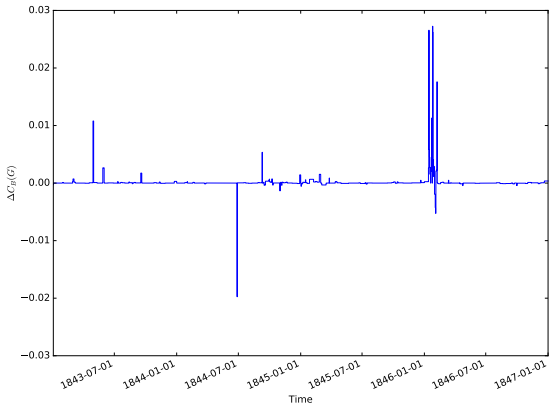


(b) Matriarchal

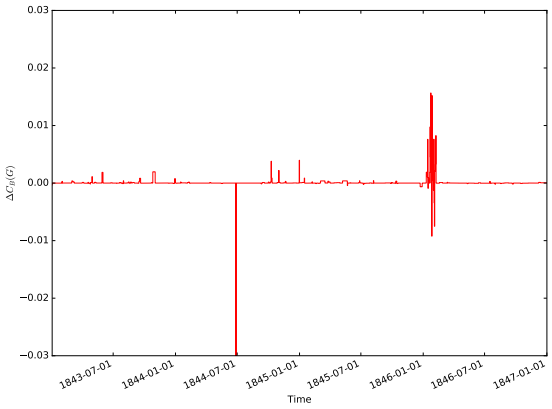


(c) Pairwise Binary

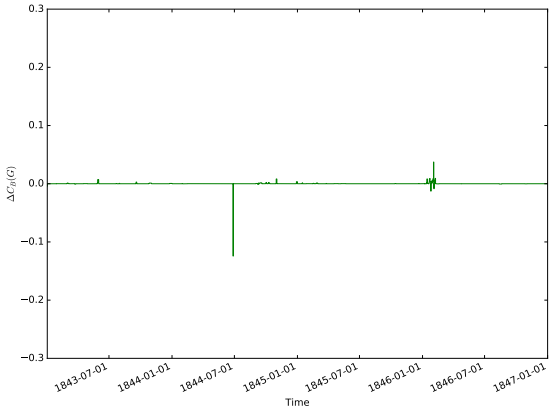
Figure 4.9: Comparing the dynamics measure $\Delta C_B(\mathcal{G})$ for all identity lenses over event-driven time. Joseph Smith's death produces a larger decrease in betweenness centrality for the matriarchal network (b) compared with the others.



(a) Patriarchal

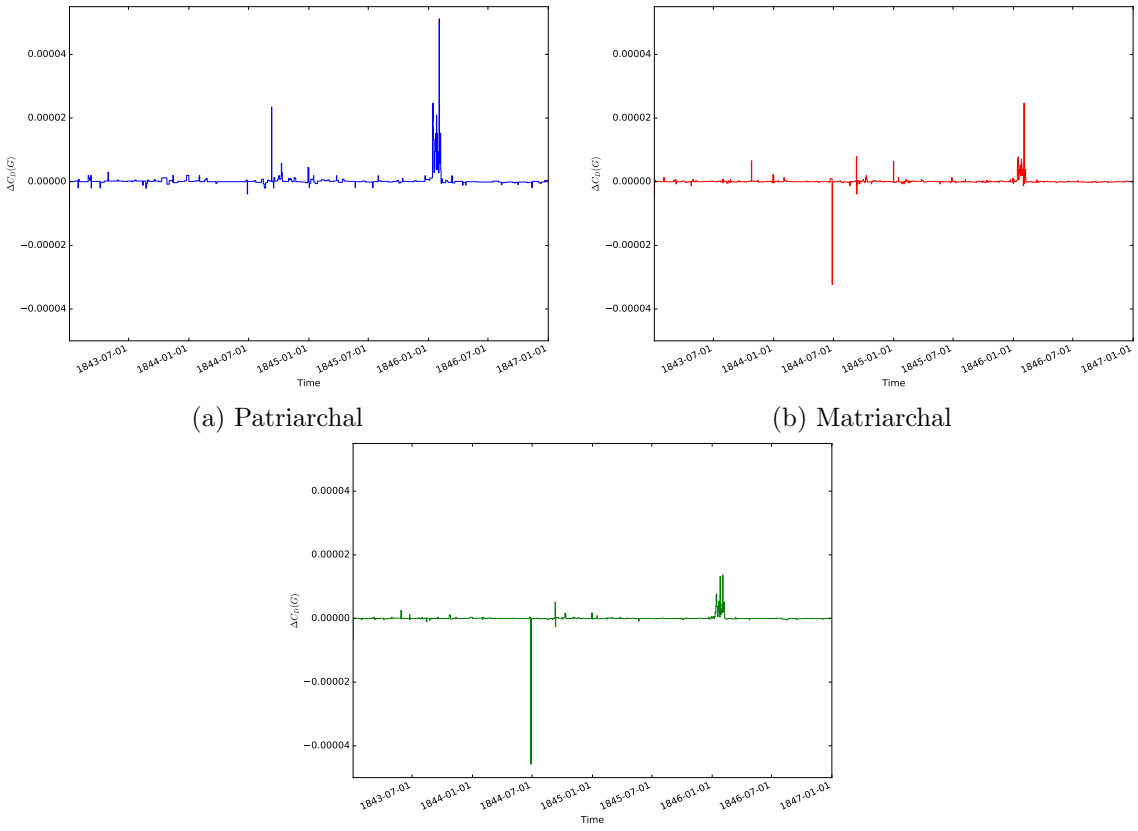


(b) Matriarchal



(c) Pairwise Binary

Figure 4.10: Comparing the dynamics measure $\Delta C_D(\mathcal{G})$ for all identity lenses over event-driven time. Smith's death has the largest effect on density for the pairwise binary network, but little effect on the patriarchal network's density.

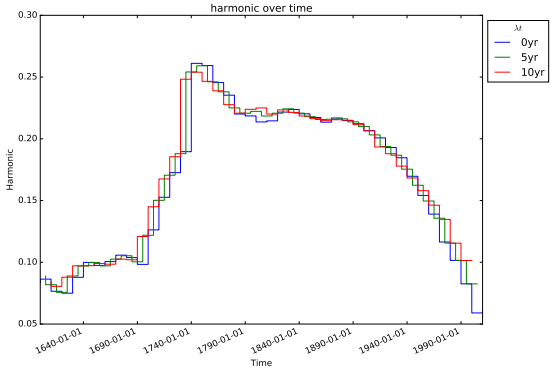


(a) Patriarchal

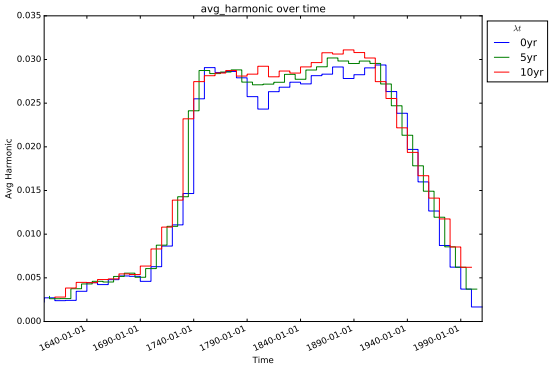
(b) Matriarchal

(c) Pairwise Binary

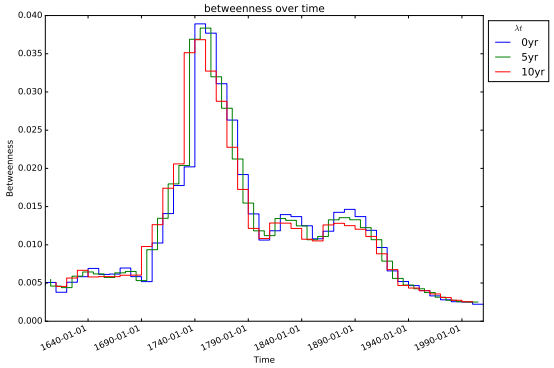
Figure 4.11: Centrality measures over the SNAC dataset under the Union sampling method. Comparing values under different λt-sized windows does not prove as useful in these plots, since node volatility affects the distraction of the distributions as λt increases.



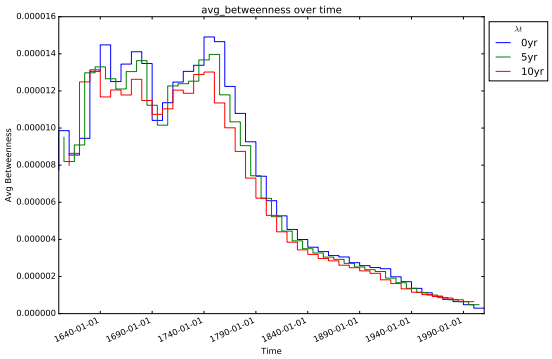
(a) C_H



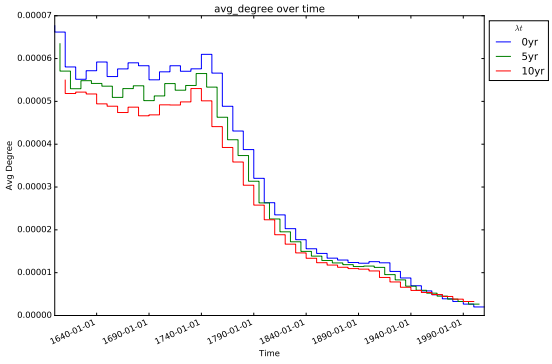
(b) C^*_{AH}



(c) C_B

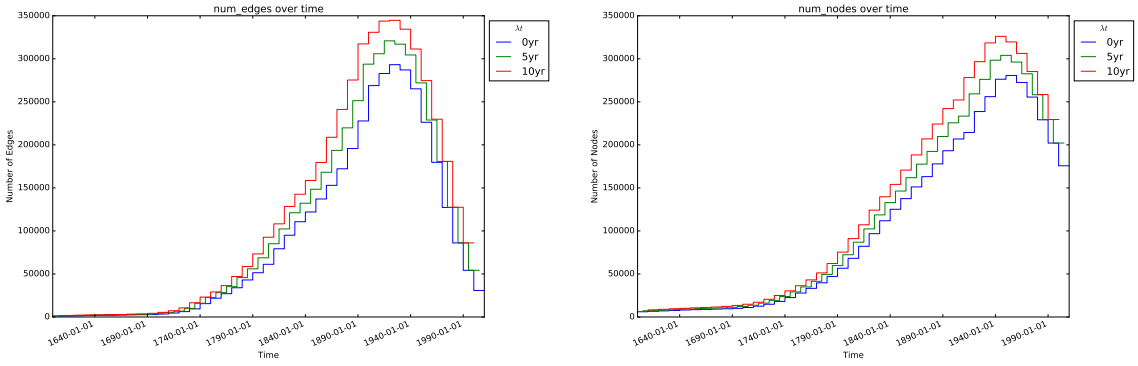


(d) C^*_{AB}



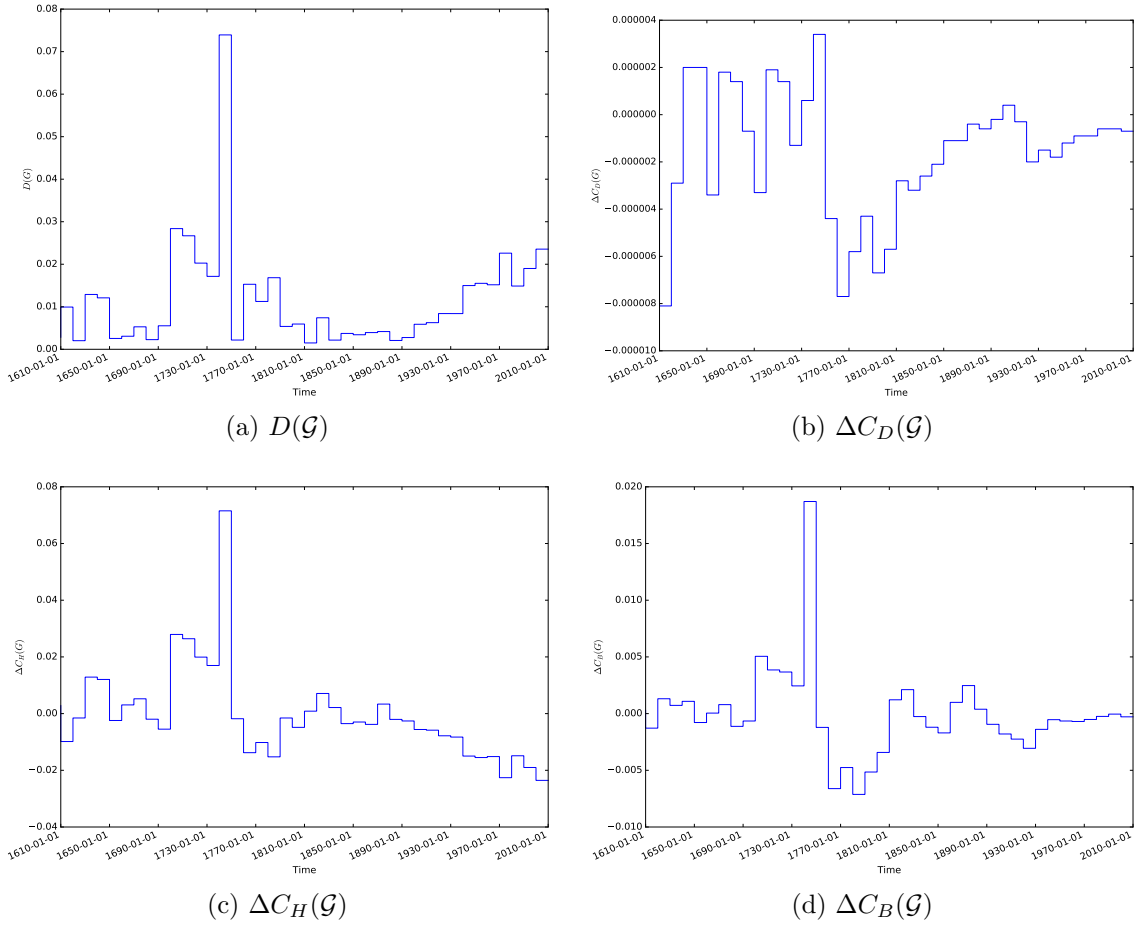
(e) C_D

Figure 4.12: More identities (nodes) and edges exist in SNAC during the mid-1900s. Sampling with larger λt-sized windows using the Union sampling produces larger-sampled graphs.



(a) Edges in SNAC (b) Nodes in SNAC

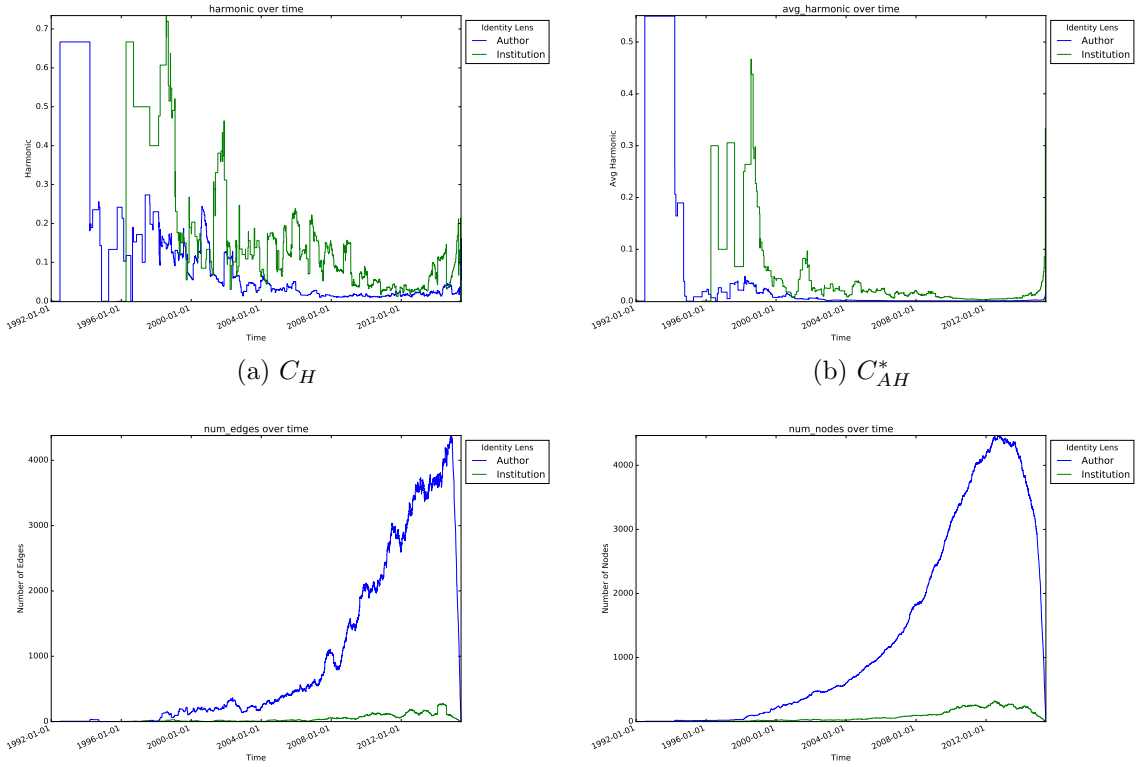
Figure 4.13: Dynamics measures over SNAC. A spike activity is observed in the mid-1700s corresponding to well-described individuals in the Harvard and Yale holdings imported into SNAC.



(a) $D(\mathcal{G})$ (b) $\Delta C_D(\mathcal{G})$

(c) $\Delta C_H(\mathcal{G})$ (d) $\Delta C_B(\mathcal{G})$

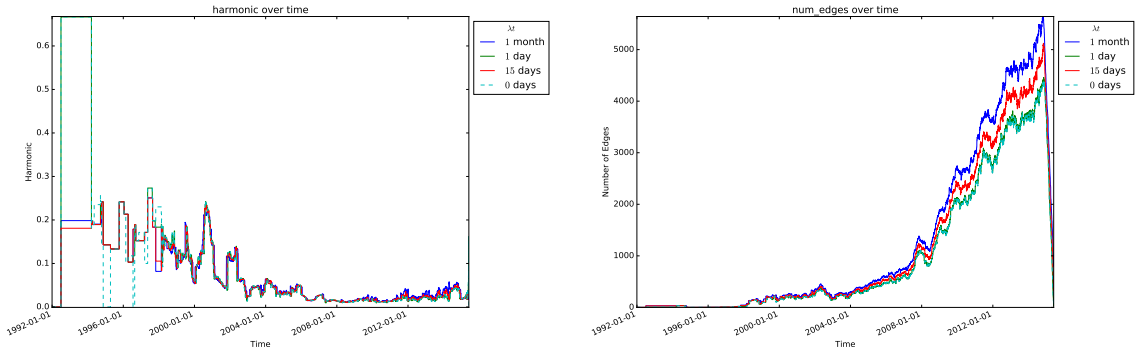
Figure 4.14: Harmonic centrality (a) and average node-centric harmonic centrality measures (b) over both author and institutional identity lenses under a 0-width λt interval sampled using event-driven time. A nearly exponential increase in co-authorship is evidenced in the edge (c) and node (d) counts.



(a) C_H (b) C^*_{AH}

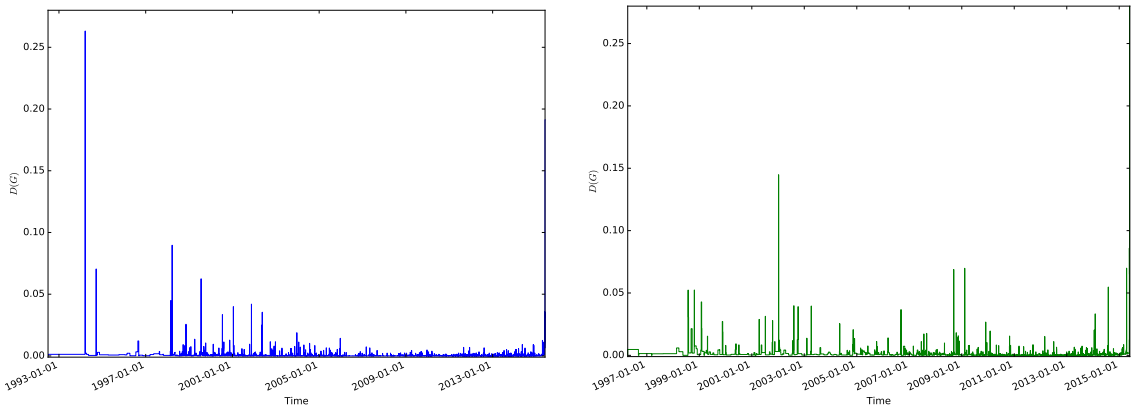
(c) Edge Count (d) Node Count

Figure 4.15: Similar to the SNAC example, while increasing the λt window size using the Union sampling includes more nodes and edges (b) into the samples, the metrics (a) do not exhibit pronounced differences. These plots depict the author identity lens under 4 sampling window sizes.



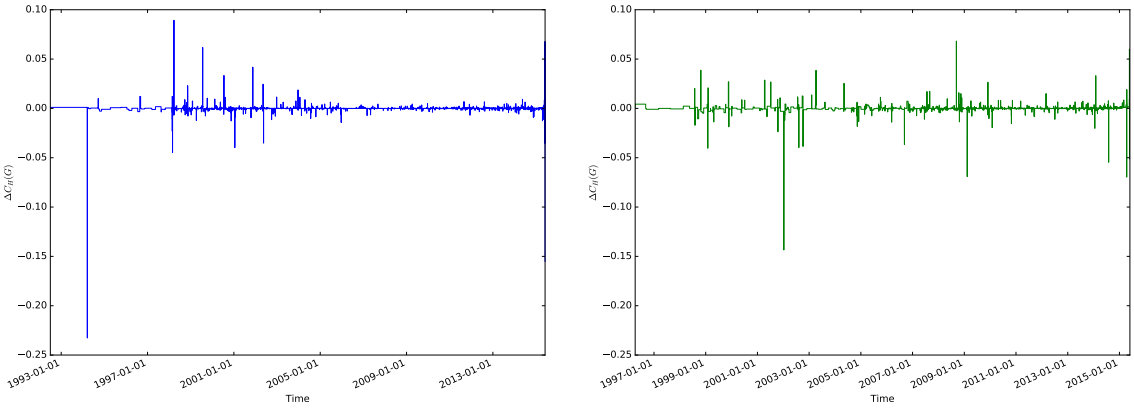
(a) C_H (b) Number of Edges

Figure 4.16: Comparing the dynamics measure $D(\mathcal{G})$ for both the author (a) and institutional (b) identity lenses. The author network shows high dynamics early, while the institutional network exhibits sustained activity.



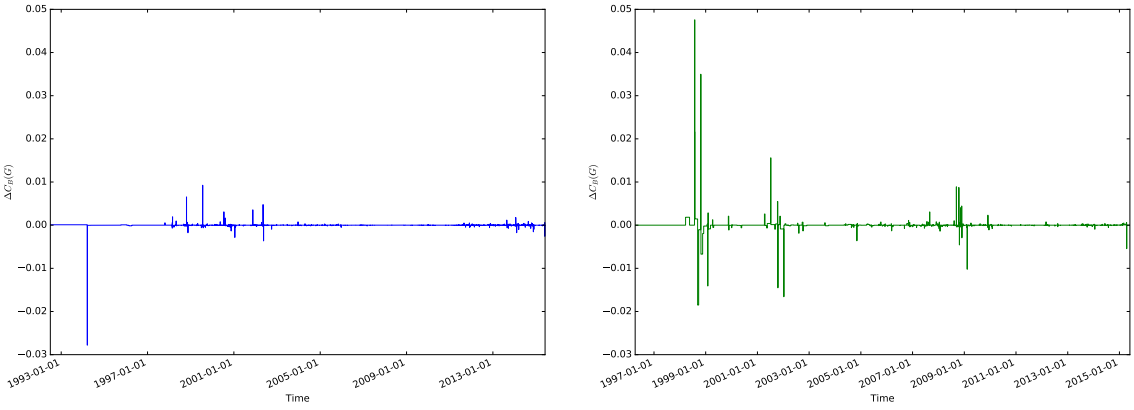
(a) Author (b) Institution

Figure 4.17: Comparing the change in harmonic centrality ΔC_H for both the author (a) and institutional (b) identity lenses.



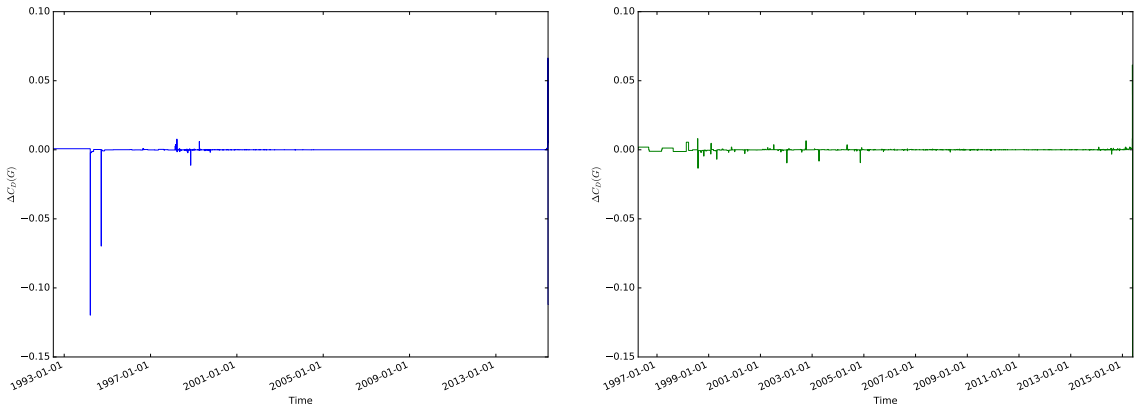
(a) Author (b) Institution

Figure 4.18: Comparing the change in betweenness centrality ΔC_B for both the author (a) and institutional (b) identity lenses.



(a) Author (b) Institution

Figure 4.19: Comparing the change in density ΔC_D for both the author (a) and institutional (b) identity lenses. Aside from large drops in density when the author network was relatively small, there is very little change in density under either identity lens.



(a) Author (b) Institution

Chapter 5

Visualization Extensions: Nauvoo

In the process of defining the domain of the Nauvoo Marriage Project for analysis, we open a natural extension of the research: visualization. Specifically, an attempt to provide a mechanism to display, interact with, and begin to understand the highly evolving marriages extant in the dataset. We explored an array of visualization techniques for depicting family and genealogical structures and social networks [2, 8, 38–49]. It was imperative to Dr. Flake that we express the family unit and its participants as a cohesive group, maintain a temporal ordering of events, and produce a depiction in which the persons and various relationship types are quickly discernible. We found that current visualization techniques are insufficient to fully express the level of complexity found in our familial structures.

Adequately visualizing the familial, kinship, and lineage structures—and the connections between them—are important to begin to uncover and understand the concepts of marriage and kinship being defined and redefined in early Mormon culture. Although polygamy appears in many cultures, to the best of our knowledge there has never been a study on the visualization of the polygamous extended-nuclear "family unit."

5.1 Related Visualizations

Visualizing traditional lineages and familial structures is a well-studied problem [42, 44–49]. Early attempts such as traditional family trees, while useful to record historical genealogical data, fail to evidence details of more complex marital structures, including marriages with more than two spouses. Figure 5.1a depicts a traditional representation of Parley Pratt's family unit consisting of 13 parents and 16 children. As more spouses and children are added, for example Brigham Young's family unit shown in Figure 5.1b, the diagram expands to become unreadable. This visualization also breaks a few guiding principles for family unit depictions: the depiction should maintain a temporal ordering (*temporality*), participants should be displayed together as a unit (*locality*), and the types of spousal and parental relationships should be quickly discernible (*distinguishability*). Even though order can be maintained in the spousal relationships, maintaining temporal order in the children would further complicate the visualization. Likewise, the children in this diagram visually separate the spouses because of their parental connections, giving a false impression of time-span between marriages. We therefore highlight a few recent related works which seek to address similar more complex genealogies and to capture the temporal nature of family units.

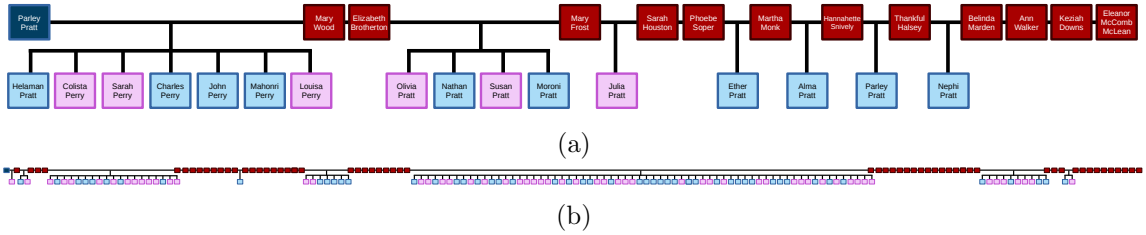


Figure 5.1: Traditional family tree depictions for (a) Parley Pratt and (b) Brigham Young, similar to the design of Family Echo's layout, `http://familyecho.com`. Spouses are shown in order, left-to-right, but the spacing with children produces a false sense of time across the visualization. Similarly, overall child ordering is not depicted. Including ancestors for the parents will combine these two trees, decreasing their legibility, since Pratt and Young married two sets of sisters.

GeneaQuilts [44], was created by Bezerianos, et al, to display large-scale genealogies of thousands of individuals. Individuals from each generation are listed, connected by grids that match parents from the previous generation to children in the subsequent generation. The visualization flows from left to right through the generations. GeneaQuilts achieves *locality* only in small cases, but it fails to provide *temporality*. The authors employ an intra-generational layout algorithm that favors grouping siblings of the same biological parents, which provides *locality*. As the number of parents in the same generation increase, parents and children are spread farther apart in the display, reducing *locality*. An attempt to introduce *temporality* by replacing the layout algorithm with one ordering participants

temporally would require sorting by birth date and marriage date simultaneously, since parents and their siblings are shown in the same genealogical level.

In contrast, Kim, et al [45], created the TimeNet visualization for genealogical data to better address and visualize each family unit's temporal relationships. Their visualization focuses on a time-line of individuals' lives as a line from their birth to their death. When individuals marry, their time-lines converge to a horizontal axis; when they divorce, their time-lines bifurcate. Additional spouses are depicted as additional converging time-lines to the "cluster" of individuals in the shared relationship. Children are depicted with their own time-lines connected to their parents by a vertical dashed line. Since their work focuses heavily on visualizing the temporal nature of the lineages, it prioritizes this over capturing internal relationships between participants of one family unit, including those in which multiple spouses and children are involved. As individuals divorce and marry, the diagram becomes more visually complicated and nuclear family units are spread farther apart. The authors make the choice to exhibit *temporality* at the cost of *locality* and *distinguishability*.

Ball and Cook [46] define a similar time-line-based visualization scheme to address the connection of individuals in a common family unit. Their work depicts time vertically, with individuals as time-lines from their birth to death; however they visually group children of the same binary (i.e. husband and wife) marriage. A box is drawn around these children, with the top of the box denoting the marriage date of their parents and the bottom denoting the death date of the last remaining child. Parents of the marriage are depicted as small boxes above the bounding box, and a connecting line drawn from the parent to their actual time-line depicted within the marriage into which they were born. While Ball and Cook's visualization exhibits the same overall *temporality* as Kim, et al, it better visually collects binary family units, increasing its *locality*. However, their visualization's attempt to provide *locality* fails when considering marriages consisting of more than two spouses.

While both Kim and Ball's work attempt to better visualize the temporal aspects of individual family units (*temporality*), all three fail to fully capture the internal dynamics and relationships within the family units (*locality*). Scaling or extending these methods for complex families such as polygamous units would degrade *distinguishability* among the visualizations. Our approach attempts to capture and visualize the temporal aspects of family units, similar to Kim and Ball, while at the same time maintaining family unit cohesiveness and providing a depiction of the larger genealogical flow and its evolution.

5.2 Visualizing Family Units: Chords

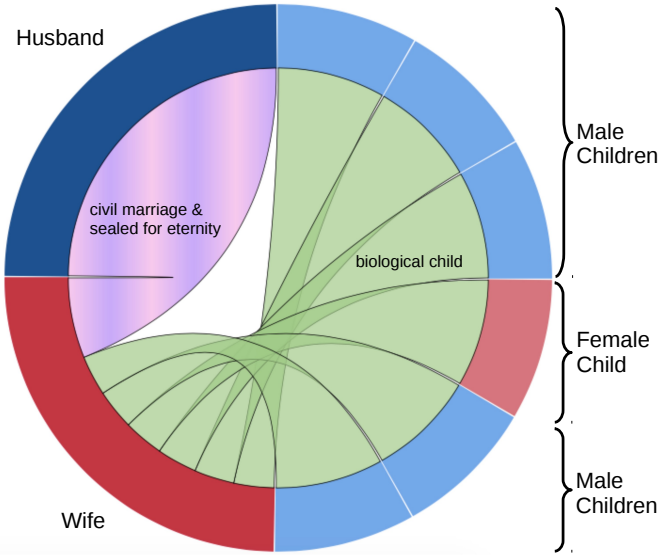


Figure 5.2: Family unit visualization displaying two parents and six children, annotated for clarity. The chord connecting the husband (top-left) and wife (bottom-left) denote their multiple relationships (pink and purple). The chords connecting the wife to the children denote a biological (green) relationship between them. To simplify the diagram, we do not also connect the children to the husband.

The participants and relationships within a family unit are often depicted as a standard directed graph—a node for each participant and an edge for each relationship between two participants [47]. The creation of such depictions immediately faces the question of "layout," to best depict locality and cohesiveness among members. A chord diagram [5, 38, 50–52] partially answers the layout question by representing each participant as a section of the circumference of a circle. This is only a partial answer to the layout question because the order of the participants around the circle and the size of the circumference section for each still need to be specified. By displaying the family unit as a circle, we embed the concept of a "nuclear family unit" into the visualization; a concept that is not found in the traditional family trees, such as those depicted in Figure 5.1. The relationships are then edges connecting two points on the circle, passing through the interior of the circle, namely "chords" of the circle, hence the name, "chord diagram."

We adapt the chord diagrams to cover familial structures' temporal nature by imposing a generational structure on the participant layout. Participants in the family unit are depicted as sections of the circumference with adult participants along the left semi-circle and with children and adoptees along the right semi-circle. This layout then produces a left-to-right flow of the lineage and generation. The choice of left-to-right is inspired by time-lines; genealogy visualizations of many orientations are common in tools today.

The relationships between participants, including spouse/spouse and parent/child relationships, are depicted as the chords. To include an additional overall temporal aspect to the diagram, participants are arranged in chronological order from top to bottom, such that newer members of the family unit, in either generation, are drawn closer to the bottom.

Figure 5.2 visualizes a straight-forward family unit consisting of two parents and six children. Here, red and blue depict the gender of the participants and the chord colors depict the different types of relationships present. Green represents a biological child relationship, while pink and purple denote two different types of marital relationship. As more participants enter the family unit, the expressiveness of the diagram becomes increasingly important. Figure 5.3 depicts Parley Pratt's family unit with 13 parents, 3 types of marital connections, and 16 children.

Because the complexity of the diagram increases with the number of participants, we make conceptualization decisions to maintain readability. The family unit is depicted from the perspective of one member, the *primary participant*, who is always drawn as the top-left parent of the diagram. In Figures 5.2 and 5.3, both of these individuals are patriarchs, i.e. the diagrams are shown from the husband's point-of-view. For our research, we assume that children are connected to two parents, the primary participant and another spouse. Therefore, parental relationships, either biological birth or adoption, are only connected to the spouses of the primary participant. This is a simplification on our part, due to our motivating application. The diagram could be extended to display step-children and children born to the spouses before they were married using a secondary chord color scheme or by connecting the children to each of their birth parents.

5.3 Visualizing Lineages: Lineage Flow

We approached the problem of connecting family unit diagrams by considering Sankey [53] diagrams. These diagrams can be used to convey both flow and geo-spatial information. Our primary interest is in the former, therefore we originally considered more simplified flow diagrams [54, p. 153–158], which consist of directed graphs where all edges "flow" in a particular direction. Lineages have this inherent "flowing" nature, directed from ancestors to descendants. Specifically, when considering lineages, we note that the flow is verbalized as individuals flowing from the marriage of their parents to their marriages as adults, then their children flowing to the next generation, and so on. Therefore,

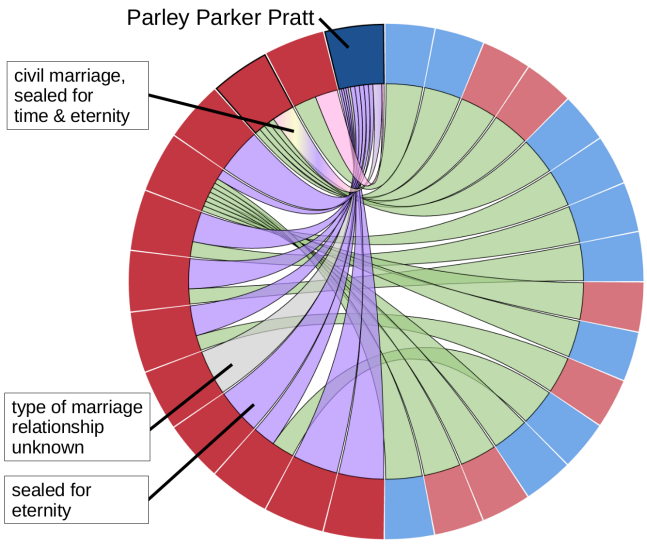


Figure 5.3: Parley Pratt's family unit depicted using our modified chord diagram, showing his 12 wives and 16 biological children. He was civilly married (pink) to his first wife and participated in three distinct spousal relationships with his second wife: civil marriage (pink), sealed for time and eternity (yellow and purple). With most of his other wives, he was sealed for eternity.

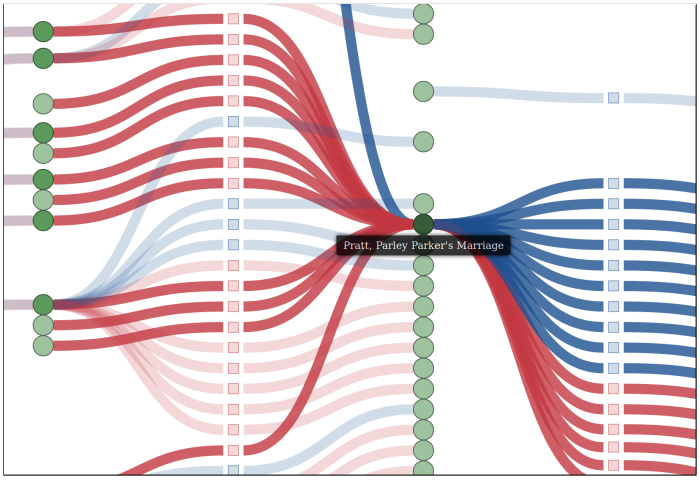


Figure 5.4: A subset of Parley Pratt's lineage as depicted in the lineage flow diagram. His family unit and its participants are highlighted. Parental figures flow into the left of the family unit node (dark green); children flow out to the right.

we conceptualize this network with family units as nodes and the individuals as the directed edges from their birth to adult families.

Under this conceptualization, we create *lineage flow diagrams* that join individual family units into a comprehensive ancestral network: an identifiable family at each node connected with edges representing the people that constitute the participants. This diagram depicts male participants as blue edges and females as red edges. The individuals then connect, in a directed left-to-right flow, the family unit of their birth to their own marriages as an adult, exampled in Figure 5.5. As

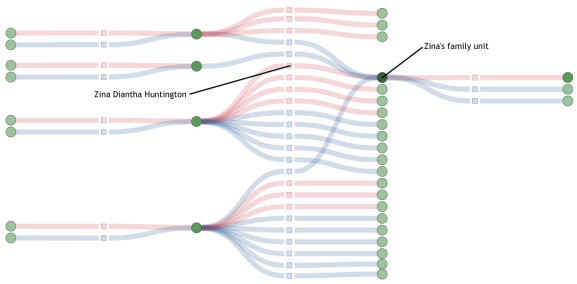


Figure 5.5: The lineage flow diagram from a matriarchal perspective, depicting family units and individuals one generational level from Zina Huntington. Men are depicted as blue edges, women as red edges. The green circles are family units and are arranged in four columns. The edges connecting circles in the leftmost column to circles in the second column are people in the generation of Zina's parents. The ones connecting the second and third columns are people in Zina's generation, and those connecting to the rightmost column are Zina's children. Here we can see three husbands connected with Zina's matriarchal family unit.

individuals connect to multiple adult spouses, their edges conceptually bifurcate to connect to all adult family units. Since hyper-edges are difficult to portray, we instead depict the lineage flow as a bipartite graph with small boxes at the bifurcation points. This tweak provides the historian with a definitive point to examine along the edge for identification. Figure 5.6 shows Catherine Kremer's connections to both John Bernhisel's father as well as Bernhisel himself.

We utilize the Sankey diagram's spatial properties to convey generational information, similar to the method employed by Cui, et al [55], in TextFlow's topic flow visualization. As they depicted time on the x-axis, we align our family units temporally into relative generations from left-to-right. In general, and for straightforward lineages, this relative generational layout will directly correlate to actual generations. In Figure 5.5, we see four generations of Zina Huntington's lineage. Her and her husbands' grandparents' family units align on the far left, the family units of both her and her husbands' parents align in the second vertical generation, her family unit and those of her siblings align in the third vertical generation, and finally those of her children on the far right. Spatial irregularities in a lineage depicted in this way alert the researcher to cases where further investigation is needed. Cross-generational connections when family units are depicted in their correct relative generation, as seen in Figure 5.6, indicate that an individual married someone of an earlier generation (e.g., a parental figure). Conversely, if an individual's family unit is depicted in a subsequent (i.e. later) generation from their temporal peers, it likely means they married a peer's child. In either case, these irregularities evidence important changes to the kinship line to be investigated.

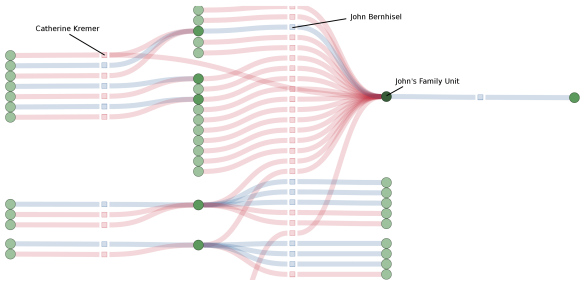


Figure 5.6: This patriarchal lineage flow diagram depicts the individuals one generational level removed from John Bernhisel. It depicts a cross-generational plural marriage of Catherine Kremer to the family units of both John Bernhisel and his father, who both were sealed to multiple spouses.

5.4 Extensions

We extend these visualization techniques to provide our historians with access to more fine-grained facets in the data. By employing a consistent and distinct use of color, refining data focus through mouseovers, and providing temporal filters using time sliders, the historian can quickly gain an overview depiction of the families and lineages while exploring in greater detail how relationships and family units evolve.

5.4.1 Use of Color

We use colors to increase the *distinguishability* in our visualizations. In order to depict the multiple relationship types needed for our complex familial structures, we chose a distinct color palette to disseminate the information on first glance. Table 5.1 provides our full use of color.

Table 5.1: Color palette for visualizations

Color	Definition
red/pink	Female individual
blue	Male individual
purple	“Eternal” sealing
yellow	“Temporal” sealing
light pink	Civil marriage
light gray	Unknown marriage type
gold	Adopted child-parent relationship
light green	Biological child-parent relationship (chord diagram)
dark green	Family unit (lineage flow diagram)

During our prototype and design phase, we tested out various neutral colors to denote gender, but found that the historian had difficulty reading the diagrams without heavily relying on the color legend. Therefore, we chose traditional gender colors, red/pink and blue, to provide a more instinctive

differentiation. In the chord diagrams, we darken the parental participant colors to better distinguish them visually from their children.

Relationship types, also differentiated by color, are independent of one another but may be adjacent or overlap on the visualization. We opted for distinct colors to provide *distinguishability* between them. In certain cases, such as the husband-wife relation back in Figure 5.2, we utilize a repeating gradient to denote multiple relationship types rather than increasing the number of connecting chords.

5.4.2 Mouseover Focus

To expose more detailed information about the individuals and family units, we use the mouse as an analogue for the historian's focus. By hovering over a specific person or relationship in our family unit visualization, as shown in Figure 5.7, other relationships are shaded to highlight only those connected to the one under observation. In this screenshot, the historian is highlighting Mary Ann Frost to feature her four children and multiple relationships with her husband. When more information is known about a relationship, such as marriage or divorce dates, we provide those details in a "more information" box on mouseover.

By providing this functionality to the historian, we maintain *locality* of the participants in the display while increasing the amount of detail available.

5.4.3 Time Sliders

Since families and lineages are evolving networks [56], as new individuals get married into the family or children are born and adopted, we provide time sliders to allow the historian to view the changes in these networks over time.

Internal familial relationships may also change as the family adapts to these new individuals. Therefore, we may use our chord diagrams to depict the state of the family unit at a particular point in time and consider the diagram to evolve with the family unit. We provide a time-line and "time slider," as seen in Figure 5.7, to allow historians to navigate through the internal dynamics of the family unit over its lifetime, beginning when the first parents get married and continuing until the last child passes away. Figure 5.8 shows the historian stepping through another family unit, that of

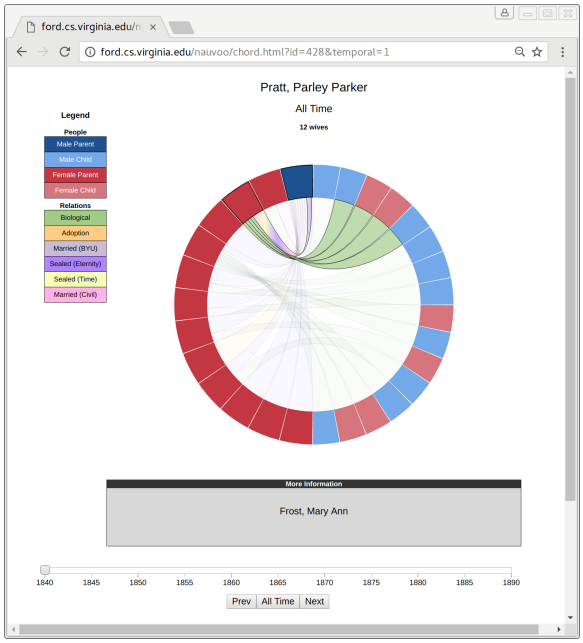


Figure 5.7: Chord diagram visualization interface. Hovering over the participants or relationship chords provide more information in the box below the diagram. Here, the researcher is choosing to inspect Mary Ann Frost and her connections to Parley Pratt and her four biological children.

Alpheus Cutler, over four different time points. Conceptually, each of these chord diagrams would constitute a slice of a spatio-temporal 3D chord cylinder depicting the overall evolution of the family. However, meaningfully representing and navigating such a diagram is difficult, leading us to focus on our time-line visualization.

By adding the time slider to our lineage flow diagrams, we evidence two dimensions of time. First is the who-beget-whom, left-to-right flow that is inherent in the diagram. Second is the "wall-clock" time evidenced by animation under the time slider interaction. This is a hybrid approach according to Beck, et al [57], and Hadlak, et al [58], combining an integrated network time-line (time as the base representation) with user-driven animation. It therefore allows the historian to get both a glance of the entire lineage and see the lineage progress as time passes, highlighting anomalies in the relative generational layout of the diagram. As the slider is moved, all individuals and marriages of the lineage flow diagram not present during the time chosen are shaded from view, highlighting the "current" picture of the lineage. Spouses that were born but not yet married are depicted as partial, disconnected edges until being connected to their adult family units on their marriage dates. Likewise, they are disconnected from those family units on divorce before being completely hidden from view on death dates. Even though we saw back in Figure 5.5 that Zina had three husbands in her lineage, stepping through time in Figure 5.9c-d we note that she only had two living husbands

simultaneously, marrying both Joseph Smith and Henry Jacobs in 1842 before having her first child. Examining the temporal aspects of Figure 5.6 shows that Kremer's spousal attachment to John Bernhisel's family unit does not happen within their lifetimes, indicating that it is a posthumous marital sealing for kinship purposes.

5.5 Evaluation

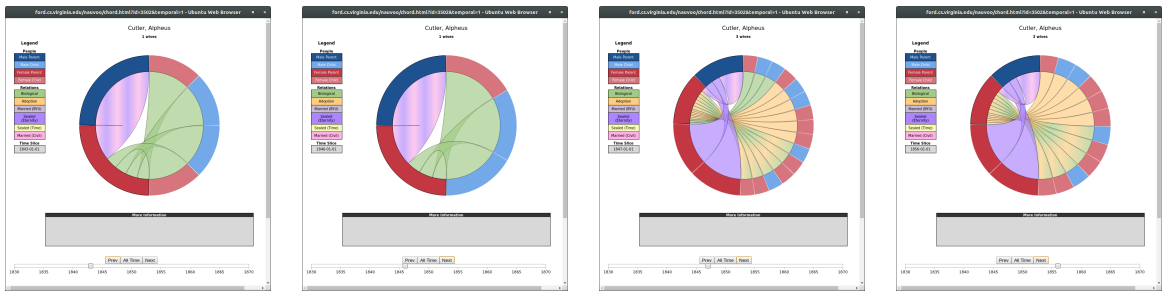
We implemented our visualizations for web consumption to allow our historians the greatest access to new views of the dataset. We then captured their reactions to evaluate the usefulness of the approaches.

5.5.1 Implementation Details

Our implementation began with code from the D3js [59] framework. We augmented the existing chord and Sankey diagram layout engines with additional layout constraints to depict our new family and lineage flow structures. We chose a simplified interface, as seen throughout Figures 5.7, 5.8, and 5.9, to maintain the historian's focus on the visualization itself.

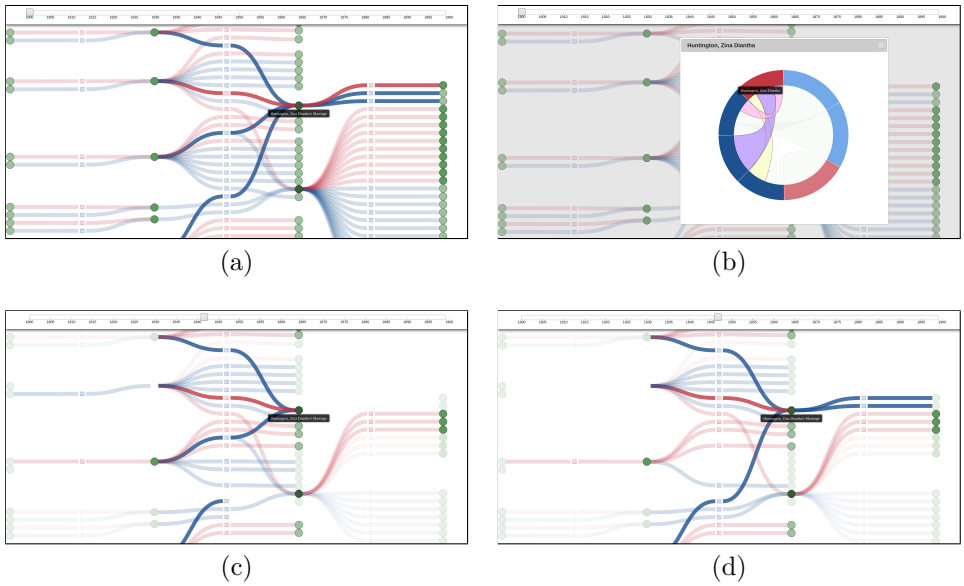
To create the lineage flow diagram interface, we adapted the D3js Sankey algorithm to depict all nodes and edges as the same size. We also modified the layout algorithm to maintain our relative generational separation, so that all family units (nodes) depicted together vertically belong to the same relative generation. An in-place family unit visualization is supplied to connect the lineage view with the intra-familial view. Upon clicking a family unit node, the historian is presented with a modal box containing a simplified chord diagram interface depicting that family unit's relationships—producing a "zooming" feature to inspect each node of the diagram.

We provide a web interface to the dataset and visualizations focused on 82 individuals central to Mormonism during the Nauvoo Temple era, available at `http://nauvoo.iath.virginia.edu/viz`. Family unit chord diagrams and lineage flows, from the point of view of each of these individuals, showcase the complex structures and dynamics of the best quality data available from the dataset. These diagrams are capable of depicting the marital relationships between Joseph Smith and his 57



(a) (b) (c) (d)

Figure 5.8: Temporal interaction with the time slider at the bottom of the diagram allows the user to see how the intra-familial interactions change over time. Over the course of Alpheus Cutler's family unit, we see an initial binary marriage with four children in 1843 (a), followed by one biological child passing away in 1844, leaving three children in 1846 (b). In 1847, there are two new wives and twelve children newly adopted to the first wife (c), with the remaining biological children dying by 1856 (d).



(a) (b)

(c) (d)

Figure 5.9: The lineage flow diagram interface (a) begins by showing the entire available lineage throughout time. Zina Huntington's matriarchal family unit is shown highlighted, including more context than Figure 5.5. Clicking any family unit provides an in-place interactive chord diagram for that unit (b). Interacting with the time-line above the diagram shows only those individuals and family units existing during the time chosen (c, d). In 1842 (c), Zina's plural marriage to Henry Jacobs and Joseph Smith (both living) is highlighted. Brigham Young is also highlighted as a future participant, but he has not yet joined Zina's family unit. By 1847 (d), Smith has died and Zina is married to Jacobs and Young.

wives[1] and Brigham Young and his 67 wives[2], as well as Zina Huntington and her 3 husbands[3]. Our interface allows the historian to choose "degrees of separation" from the focused individual, and is capable of visualizing multiple relative generations with hundreds of participants[4].

[1] Joseph Smith's family unit and lineage visualizations are available at `http://nauvoo.iath.virginia.edu/viz?q=smith`.

[2] Brigham Young's visualizations are available at `http://nauvoo.iath.virginia.edu/viz?q=young`.

[3] Zina Huntington's visualizations are available at `http://nauvoo.iath.virginia.edu/viz?q=zina`.

[4] For the Nauvoo Marriage Project visualizations, we recommend limiting lineage flow displays to 2 "degrees of separation" which produce 5-6 relative generations.

5.5.2 Reception

The usefulness of our approaches, especially the temporal versions of the visualizations, was immediately apparent. Within the first 20 minutes of examining the temporal data views, the historians were able to identify dozens of problems in the dataset that were not apparent over years of data curation. We found children sealed to the wrong parents, missing dates from our core example set, and erroneous marriages. Within that brief time, Dr. Flake remarked that this was "a marked shift in what we can do with the data:" it gives rise to questions, allowing her to do further research.

Aside from surfacing data issues, by providing this "wall-clock" view of the evolving lineage structure, we allowed the historian to see patterns in relation to their larger historical context, such as the merging of older and younger generations as part of the preparation for the westward movement of the Mormons to a more hostile environment.

Our matriarchal lineage visualizations have enabled historians to identify and interrogate the gendered status relationships in Mormonism's genealogical and familial structures. Particularly novel in this approach is the usefulness of these representations to enable analysis of women's liberty to marry and divorce largely at will; analysis made possible by the visualizations from a matriarchal point of view, such as Zina Huntington's shifting family unit in Figure 5.9. Though this marital liberty has long been known by scholars, discussion of it has been limited to anecdotal accounts. There has been no attempt to measure it on a social scale or test the anecdotal hypotheses regarding the dynamics of polyandry among Mormon polygamists or even whether these relationships should be deemed polyandrous.

Likewise, these visualizations have highlighted some of the patterns extant within the patriarchal structure. The variety of kinship ties and the sheer number of attachments in Mormon polygamous practices have frustrated efforts to arrive at defensible generalizations about the practice's nature and effects through traditional historical means of analysis. While countless claims have been made that the Mormons' iconoclastic marital system was fundamental to sophisticated political, economic, and ecclesiastical structures, the inter-connectedness (both its purposes and effects) has escaped comprehension due to its size and number of moving parts. Being able to discover through these depictions, and evidence with the temporal visualizations, the variety of posthumous or "paper" marriages to founder Joseph Smith is likely a key to understanding the dynastic forms of Mormon religious and political office and authority.

5.6 Future Work on Visualizations

Our future work focuses on refining our lineage flow diagrams and using the underlying kinship networks as the basis for further analytic evaluation. These lineage flow diagrams are still in their infancy: they can showcase familial interactions, cross-generational anomalies, and multiple concurrent marriages. We are considering further refinements and extensions to express more of the richness within the Nauvoo dataset. First, we will consider techniques to evidence in the lineage flow visualizations the categories of marital form (i.e. civil, "eternal," "temporal," or posthumous marriages) that are currently expressed in the chord diagrams. Likewise, we will apply techniques from the lineage flow found useful by our historians to the chord diagrams. Specifically, our historians noted that it was more apparent when children and spouses were spatially accounted for in the diagram layout, but simply hidden from view until they are born or married. Applying this technique, we will fix an individual's location in the chord layout, leaving the geometric structure of the circumference stable, while the dynamics are indicated by chords and colors coming and going through time.

A major assumption of the D3js Sankey diagram engine is that the "flow" to be displayed is acyclic. Since there are parts of the Nauvoo dataset that are not acyclic due to adoption or other inter-generational relations, we will be considering techniques to modify and extend the Sankey display to account for the exceptional connections, i.e., producing flow diagrams that maintain the relative generational flow while capturing the cycles.

In addition to increasing the amount of data provided in the diagrams, we will also investigate the user interface improvements to enable the historian easier access to that data. For example, to provide clean diagrams, identities of participants in the visualizations are currently accessible only through mouse-over interactions. By providing a search interface or legend of participants that highlight relevant portions of the diagram, we may further increase the readability and *distinguishability* of the visualizations.

Lastly, as noted above, visualizations for a specific part of the Nauvoo dataset differ depending on the choice of perspective on the marital unit, such as patriarchal, matriarchal, or binary. The differences are not standard graph relationships, e.g., dual graphs. Thus, we will investigate visualizations (and analysis) to highlight differences evidenced by choice of perspective both in their entirety and as the lineage networks evolve over time.

5.6.1 Applications to Other Domains

We note that our visualization and network conceptualization techniques, displaying people as the edges, may be applied to other datasets containing graph-centric data, such as social-document networks, communication networks, and citation networks. In social-document networks, groups of individuals participate in shared documents. For example, the Social Networks and Archival Contexts project [60] connects identities with archival documents using "referenced in" or "creator of" links. As an application of our techniques, we may consider the documents as nodes and the authors as edges connecting from the documents they are referenced in to the documents of which they are the creators.

Similarly, we may consider citation networks, those derived from electronic paper archives such as arXiv.org. In this case, we may consider the co-authored papers as the nodes with authors as edges connecting the papers they author to those in which they are cited. Visually, we would be able to see the flow of information across articles as well as disciplines. In our Nauvoo dataset, we considered different definitions of family units: binary marriages, matriarchal, and patriarchal units. Applied to citation networks, different definitions of authorship, such as individual author, departmental affiliation, or university or institutional affiliation may be considered. Visualizing and analyzing these different authorship lineages may provide larger insights into the publication trends of departments as compared to their individual faculty members. In each of these cases, we could analyze these re-conceived networks as mentioned in our future work: evaluating centralities and connectedness as the networks evolve and papers and authors, or documents and identities, are cited or included in the graph.

5.7 Conclusion

In this chapter, we have adapted existing data visualization techniques, flow and chord diagrams, to produce novel conceptualizations and visualizations of genealogical and familial structures. Compared with existing techniques and best practices, our visualizations are capable of providing a rich view of kinship structure and lineage flow as well as the internal relationships and dynamics of the individual familial structures. They maintain *locality* among members of the familial units, provide an overall temporal flow (*temporality*), and allow for quick *distinguishability* between relationship types.

We implemented and applied our techniques to a complex set of families from early Mormonism in Nauvoo, IL. In our preliminary research group, we have found that the lineage flow and chord diagrams provide the researchers with evocative and provocative visual cues for relationships in the Nauvoo dataset. That includes highlighting data irregularities and patterns in the kinship structure over time, while also hinting at the larger role and freedom enjoyed by women in their marriages. While we anticipate additional refinements in diagram structure, our lineage flow visualizations are providing starting points for interesting analytic results on the underlying re-conceived network of marriage nodes and connecting individuals.

Chapter 6

Conclusion

Through this work, we have created and discussed new methods for analyzing evolving networks and providing useful dynamics profiles and visualizations to researchers. We began with a formalization of evolving networks as Time-Varying Graphs, from Casteigts, et al [14]. We expanded upon this formalism to produce additional views, or "identity lenses," of the networks, which we call Time/Identity-Varying Graphs. These related networks are then compared for competing dynamics to understand if the underlying network favored one particular definition of node over its lifespan. We then defined six new sampling methods to produce a comprehensive state of the network, which may be sampled at any point over its lifespan. Those sampling methods provide different depictions of the underlying dynamics inherent in the network at any given point; specifically, if nodes and edges were stable across a researcher-defined sampling window. Utilizing these sampling methods to produce time-driven static graphs, we discussed graph-wide metrics for determining the centrality and topology of the graph and we proved properties of those metrics under certain assumptions.

By combining the sampling methods, λt sampling window sizes, and centrality metrics, we were able to evidence a difference in metric distributions based on sampling choice that allowed us to determine a granular dynamic picture of the network under certain conditions. Secondly, we utilized the rate of change in our metrics throughout the evolving networks to create a dynamics profile based on $D(\mathcal{G})$, the L^2 norm of three important metrics of change: graph-wide betweenness centrality, graph-wide harmonic centrality, and graph density. Our overall approach is depicted in Figure 6.1.

We verified this approach by conducting a large number of experiments over synthetically generated

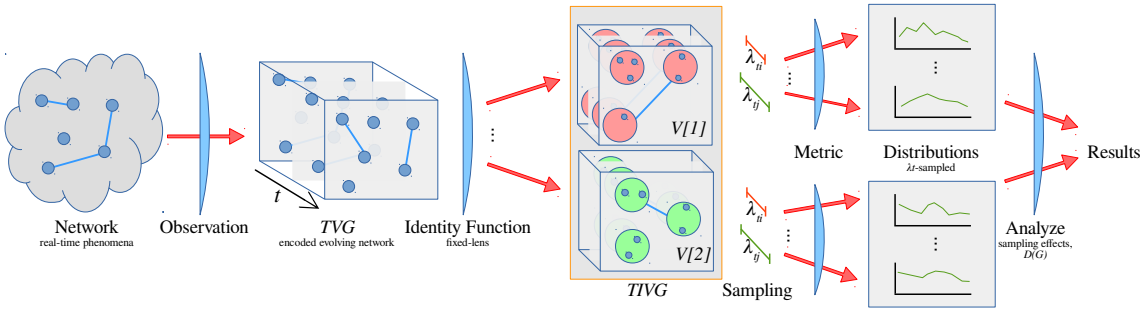


Figure 6.1: A high-level overview of our approach, from capturing a network as a TVG, conceptualizing that TVG under domain-specific identity lenses, sampling and analyzing each evolving network to produce and compare distributions to uncover the dynamics in the network.

evolving networks. Generating both TVGs with a connectedness constraint that required a minimal connecting star and similar TVGs without connectedness constraints allowed us to test the assumptions about our metrics without and with network properties that were shown to produce specific results. We found that in both cases, our analysis techniques were successful at evidencing the topological changes as well as periods of high dynamics in the networks.

We then applied these approaches to three motivating examples: the Nauvoo Marriage Project, an ArXiv co-citation graph, and the SNAC social-document network. Each of these examples provided distinct research questions that involved understanding their evolving network in its context. By working with those researchers, we were able to successfully utilize our methods to address their posed questions and uncover new questions about their underlying domains. For Dr. Flake this was especially useful as it guided her to address women's roles in the society and directed her to new historical narratives such as Helen Kimball's connection of the Smith, Young, and Kimball families. For the SNAC project, our methods highlighted dynamic points in the historical connections that may evidence bias in archival collections and may drive future research to address that bias.

Throughout our synthetic and real-world examples, we have determined a usefulness profile for our approach. When nodes are relatively stable throughout the lifespan of any TVG $\mathcal{G}$, then utilizing both our sampling size comparison and $D(\mathcal{G})$ approaches may highlight important and dynamic timepoints over the lifespan of $\mathcal{G}$. Without stable nodes, the volatility and dynamic nature of the nodes may skew the results of the sampling size comparison approach, however the $D(\mathcal{G})$ computation will result in a profile that highlights changes in both edges and nodes.

Lastly, our research has incorporated strands of complementary research along the way, including visualization in digital humanities. Our design and implementation of the chord diagram visualization

came out of and prompted new conceptualizations of "marriage" in the Nauvoo Marriage Project, while the interactive forms of the chord diagrams and of Sankey-like flow diagrams provided new avenues for exploration of the temporally-accessible lineages and the inter-marital social dynamics in the Nauvoo community. These visualization techniques have shown to be provocative and evocative to Dr. Flake and other family history researchers. Dr. Flake has said that our approaches produced "a marked shift in what we can do with the data."

We have therefore successfully validated our extensions to evolving networks and related measures through the analysis of synthetic datasets. Through our motivating examples, we have exhibited the effectiveness of our approaches in the context of the applications, pointed the domain-specific researchers to new questions, and extended state of the art visualizations to depict our more dynamic networks.

6.1 Future Work

This work is a starting point in capturing and presenting the dynamics of Time-Varying Graphs and opens the door to multiple avenues of future research. Let us pinpoint three major areas of expansion and provide further discussion on them: theoretic analysis and empirical studies; addressing computational and implementation challenges; and application to additional domains of study.

6.1.1 Theoretic and Empirical Studies

There is more work to be done in the theoretic realm, both exploring the properties of the metrics over evolving networks and using additional synthetic experiments to validate those results. We have been able to prove properties over harmonic and betweenness centrality under a constrained minimal connectedness property, as well as reason about those properties in unconstrained networks and evidence them in our synthetic experiments. A subsequent stage of theoretical analysis would be to produce formal proofs for the properties of other centrality measures over the constrained networks and proofs for the properties of all centrality measures over the unconstrained networks.

Secondly, we maintained our synthetic design for comparison between the constrained and unconstrained graphs, which created dense graphs. A natural extension, therefore, is to relax the density constraint and attempt to uncover properties of networks that are significantly sparser, on the order

of $|V| \approx |E|$. The Nauvoo Marriage Project uncovered one specific density test to perform, since by design whenever a new node is added to the network, it must be connected by one edge. That is, when a person's marriage node is added, it must be the case that the person-edge has been "born." By beginning with a TVG containing only a minimal spanning star, if at each synthetic timepoint one node and one edge are connected to the hub, will the network maintain its centrality as the edge count increases? Likewise, can we increase the network connecting at least one edge for each additional node and maintain a level of centrality that evidences this change?

Likewise, since we were able to relax the "minimal star" requirement for our proofs on the properties of harmonic centrality, we would like to empirically test other minimal connectivity topologies that adhere to the assumptions in the proof, namely a $diameter(G) \leq 2$. Additional minimally connected constraints should include two-, three-, up to M-star graphs with M hubs each connected to $N - M$ other nodes.

Lastly, we would like to explore further the possibility of automating the threshold selection process when analyzing the dynamics with $D(\mathcal{G})$. In our initial studies, we arbitrarily chose the threshold value that removed a large number of local maxima, based on observation. This was shown to be sufficient, but, for example, analyzing the maximal $D(\mathcal{G})$ values in decreasing order until an insignificant point of change is found may produce an algorithm for choosing threshold values in future applications.

6.1.2 Computational and Implementation Challenges

In order to facilitate these new empirical studies, we must address the challenges in the implementation and produce an analytic tool that scales better and supports additional TVG properties.

First, our *TVGAnalyze* implementation was limited to networks of less than 2^{32} edges, nodes, and timepoints. We could not efficiently store more than 2^{32} nodes, edges, or timepoints due to the limitations of the Java programming language. Java limits array size, and therefore all of the Collections objects sizes, to a 32-bit integer, even in 64-bit architectures. To create and store a larger network would require a purely linked list implementation or other specialized structure not indexed or addressed using an in-memory array. For this purpose, we contemplate utilizing Neo4J[1], an open-source graph database that provides a Java API. It has a specialized storage scheme without

[1] `http://neo4j.com`

relying on the Collections interface or in-memory arrays; instead, making efficient use of linked lists. We presume that following their design philosophy, we may be able to store larger TVGs, however we have not considered a computational complexity analysis when time is factored into the properties of each node and edge.

Likewise, while our underlying data structure was able to support hyper-edges, the underlying analysis tool GraphStream [31] does not. We may remove both of these limitations of the implementation by redesigning our data structure and porting it to another language, as well as re-implementing the subroutines for Dijkstra's and Brandes' algorithms.

Since our algorithms may be highly parallelizable, our re-implementation should make use of this distributed nature. Our current implementation uses shared memory and threading across one system, but future designs can divide up the analysis tasks across nodes based on sampling time. The bottleneck, we believe, will be the sampling process itself and the need for all compute nodes to access the entire TVG to produce the time-specific samples. We envision one solution increasing parallelism in analyzing TVGs is through the MapReduce paradigm [61]. In that paradigm, sampling and analysis may be performed using multiple map-reduce stages, in which there are multiple mapping stages; each compute node has a portion of the TVG to sample, those samples get collected and assembled by the mappers into time-dependent sampled graphs, which then get distributed to mappers for analysis and a final reducing to collate the results across time.

6.1.3 Cross-domain Applications

The widest future area of this research is application. We have seen successful results for each of our motivating applications, especially for Dr. Flake's Nauvoo Marriage Project. Applying these techniques to other projects containing evolving networks, from healthcare applications to humanities projects, will both refine the tools and metrics we've designed while also providing a benefit to the researchers in uncovering the dynamics in their networks. Our research has led us to new short-term projects, including a dataset provided by the US National Archives containing minute-by-minute social connections during the Nixon presidency as captured in the Nixon Tapes archive.

By making our tools publicly available and open source, we anticipate that others may utilize our methods across disciplines.

Bibliography

[1] Bruno Ribeiro, Nicola Perra, and Andrea Baronchelli. Quantifying the effect of temporal resolution on time-varying networks. *Scientific reports*, 3, 2013.

[2] Albert-László Barabási. *Linked: The New Science of Networks*. Basic Books, 2002.

[3] John R Hott, Worthy N Martin, and Kathleen Flake. Evolving social structures: Networks with people as the edges. *Digital Humanities Forum, University of Kansas*, 2014.

[4] John R Hott, Worthy N Martin, and Kathleen Flake. Evolving family structures: Representation and visualization. *Family History Technology Workshop, Brigham Young University*, 2015.

[5] V. A. Vassiliev. Cohomology of knot spaces. In V. I. Arnold, editor, *Advances in Soviet Mathematics*, volume 1, pages 23–69. American Mathematical Society, 1990.

[6] Wolgang Mullër and Heidrun Schumann. Visualization methods for time-dependent data-an overview. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 1, pages 737–745. IEEE, 2003.

[7] Matthieu Barjon, Arnaud Casteigts, Serge Chaumette, Colette Johnen, and Yessin M. Neggaz. Testing temporal connectivity in sparse dynamic graphs. April 2014.

[8] Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)*, 47(1):10, 2014.

[9] Afonso Ferreira and Laurent Viennot. A note on models, algorithms, and data structures for dynamic communication networks. 2002.

[10] Afonso Ferreira. On models and algorithms for dynamic communication networks: The case for evolving graphs. In *In Proc. ALGOTEL*, 2002.

[11] Peter Grindrod, Mark C Parsons, Desmond J Higham, and Ernesto Estrada. Communicability across evolving networks. *Physical Review E*, 83(4):046120, 2011.

[12] Betsy George and Sangho Kim. Spatio-temporal networks. *SpringerBriefs in Computer Science*, 2013.

[13] Afonso Ferreira. Building a reference combinatorial model for MANETs. *Network, IEEE*, 18(5):24–29, 2004.

[14] Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012.

[15] Arnaud Casteigts, Paola Flocchini, Emmanuel Godard, Nicola Santoro, and Masafumi Yamashita. Expressivity of time-varying graphs. In *Fundamentals of Computation Theory*, pages 95–106. Springer, 2013.

[16] Walter Quattrociocchi and Frederic Amblard. Emergence through selection: The evolution of a scientific challenge. February 2011.

[17] Betsy George and Shashi Shekhar. Time-aggregated graphs for modeling spatio-temporal networks. In *Journal on Data Semantics XI*, pages 191–212. Springer, 2008.

[18] Arnaud Casteigts, Paola Flocchini, Bernard Mans, and Nicola Santoro. Shortest, fastest, and foremost broadcast in dynamic networks. *arXiv preprint arXiv:1210.3277*, 2012.

[19] B Bui Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02):267–285, 2003.

[20] Carter T Butts. A relational event framework for social action. *Sociological Methodology*, 38(1):155–200, 2008.

[21] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[22] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[23] Linton C Freeman, Douglas Roeder, and Robert R Mulholland. Centrality in social networks: Ii. experimental results. *Social networks*, 2(2):119–141, 1979.

[24] Alex Bavelas. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 22(6):725–730, 1950.

[25] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, Dec 1966.

[26] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*, volume 8. Cambridge University Press, 1994.

[27] Massimo Marchiori and Vito Latora. Harmony in the small-world. *Physica A: Statistical Mechanics and its Applications*, 285(3):539 – 546, 2000.

[28] Anthony Dekker. Conceptual distance in social network analysis. *Journal of Social Structure (JOSS)*, 6, 2005.

[29] Yannick Rochat. Closeness centrality extended to unconnected graphs: The harmonic centrality index. In *ASNA*, number EPFL-CONF-200525, 2009.

[30] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262, 2014.

[31] Yoann Pigné, Antoine Dutot, Frédéric Guinand, and Damien Olivier. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. *Emergent Properties in Natural and Artificial Complex Systems. European Conference on Complex Systems*, March 2008.

[32] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[33] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[34] Scott Hendrickson. Data scientist's approach to social data, April 2014.

[35] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[36] Alan V. Oppenheim and Ronald W. Schafer. *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1975.

[37] Lowell W. Beineke. Characterizations of derived graphs. *Journal of Combinatorial Theory*, 9(2):129 – 135, 1970.

[38] Andy Kirk. *Data Visualization: A Successful Design Process*. Community experience distilled. Packt Publishing, Limited, 2012.

[39] Linton C Freeman. Visualizing social networks. *Journal of social structure*, 1(1):4, 2000.

[40] Vladimir Batagelj and Andrej Mrvar. *Pajekanalysis and visualization of large networks*. Springer, 2004.

[41] Nathalie y Henr, Anastasia Bezerianos, and Jean-Daniel Fekete. Improving the readability of clustered social networks using node duplication. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1317–1324, Nov 2008.

[42] Janet Wesson, MC du Plessis, and Craig Oosthuizen. A zoomtree interface for searching genealogical information. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '04, pages 131–136, New York, NY, USA, 2004. ACM.

[43] Tuan Nhon Dang, Nick Pendar, and Angus G. Forbes. Timearcs: Visualizing fluctuations in dynamic networks. *Computer Graphics Forum*, 35(3), 2016.

[44] Anastasia Bezerianos, Pierre Dragicevic, Jean-Daniel Fekete, Juhee Bae, and Ben Watson. Geneaquilts: A system for exploring large genealogies. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1073–1081, Nov 2010.

[45] Nam Wook Kim, Stuart K. Card, and Jeffrey Heer. Tracing genealogical data with timenets. In *Proceedings of the International Conference on Advanced Visual Interfaces*, AVI '10, pages 241–248, New York, NY, USA, 2010. ACM.

[46] Robert Ball and David Cook. A family-centric genealogy visualization paradigm. In *Proceedings of 14th Annual Family History Technology Workshop*, 2014.

[47] Michael J. McGuffin and Ravin Balakrishnan. Interactive visualization of genealogical graphs. In *IEEE Symposium on Information Visualization*, pages 16–23, Oct 2005.

[48] Geoffrey M Draper and Richard F Riesenfeld. Interactive fan charts: A space-saving technique for genealogical graph exploration. In *Proceedings of the 8th Annual Workshop on Technology for Family History and Genealogical Research (FHTW 2008)*. Citeseer, 2008.

[49] Claurissa Tuttle, Luis Gustavo Nonato, and Claudio Silva. Pedvis: A structured, space-efficient technique for pedigree visualization. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1063–1072, Nov 2010.

[50] Maxim Kontsevich. Vassiliev's knot invariants. *Adv. in Sov. Math*, 16(2):137–150, 1993.

[51] Christian Kassel, Vladimir Turaev, et al. Chord diagram invariants of tangles and graphs. Technical report, Inst. de Recherche Math. Avancée, 1995.

[52] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):741–748, 2006.

[53] Patrick Riehmann, Manfred Hanfler, and Bernd Froehlich. Interactive sankey diagrams. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 233–240. IEEE, 2005.

[54] Robert L. Harris. *Information Graphics: A Comprehensive Illustrated Reference : Visual Tools for Analyzing, Managing, and Communicating*. Management Graphics, 1996.

[55] Weiwei Cui, Shixia Liu, Li Tan, Conglei Shi, Yangqiu Song, Zekai J Gao, Huamin Qu, and Xin Tong. Textflow: Towards better understanding of evolving topics in text. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2412–2421, 2011.

[56] John R Hott, Worthy N Martin, and Kathleen Flake. Identity lenses in analyzing evolving social structures. *Digital Humanities*, pages 565–567, 2016.

[57] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum*, 36(1):133–159, 2017.

[58] Steffen Hadlak, Heidrun Schumann, and Hans-Jrg Schulz. A Survey of Multi-faceted Graph Visualization. In R. Borgo, F. Ganovelli, and I. Viola, editors, *Eurographics Conference on Visualization (EuroVis) - STARs*. The Eurographics Association, 2015.

[59] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec 2011.

[60] Ray R. Larson and Krishna Janakiraman. Connecting archival collections: The social networks and archival context project. In *Proceedings of the 15th International Conference on Theory and Practice of Digital Libraries: Research and Advanced Technology for Digital Libraries*, TPDL'11, pages 3–14, Berlin, Heidelberg, 2011. Springer-Verlag.

[61] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

Appendix

A Publication List

- Hott, J. R., Martin, W. N., and Flake, K. 2018. Visualization of Complex Familial and Social Structures. Electronic Imaging, Burlingame, CA.

- Hott, J. R., Martin, W. N., and Flake, K. 2016. Visualizing Dynamics of Complex Familial Structures (Poster). IEEE Information Visualization, Baltimore, MD.

- Hott, J. R., Martin, W. N., and Flake, K. 2016. Identity Lenses in Analyzing Evolving Social Structures. Digital Humanities Conference. Kraków, Poland.

- Hott, J. R., Martin, W. N., and Flake, K. 2015. Visualizing and Analyzing Identity Classes in Evolving Social Structures. Chicago Colloquium on Digital Humanities and Computer Science, University of Chicago. Chicago, IL.

- Hott, J. R., Martin, W. N., and Flake, K. 2015. Evolving Family Structures: Representation and Visualization. Family History Technology Workshop, Brigham Young University. Provo, UT.

- Hott, J. R., Martin, W.N., et al. 2014. Evolving Social Structures: Networks with People as the Edges. Digital Humanities Forum, University of Kansas. Lawrence, KS. *Best paper award.*

- Hott, J. R., Brunelle, N., Myers, J., Rassen, J. and shelat, a. 2012. KD-Tree Algorithm for Propensity Score Matching With Three or More Treatment Groups. Technical Report Series. Division of Pharmacoepidemiology And Pharmacoeconomics, Department of Medicine, Brigham and Women's Hospital and Harvard Medical School. Boston, MA.

- Noonan, R. E. and Hott, J. R. 2007. A course in software development. In Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (Covington, Kentucky, USA, March 07 - 11, 2007). SIGCSE '07. ACM Press, New York, NY, 135-139.

B Table of Notation

Table B.1: Notation used throughout this dissertation

$\mathbb{T}$	objective, continuous, time
$\mathcal{T}$	Δt time, measured in equal-spaced intervals
T	event-driven time, set of time-points when events occur
$\mathcal{E}$	set of all events
ε_i	one event
$\tau(\varepsilon_i)$	function mapping events to their time-point in $\mathbb{T}$
δt	fixed-size evenly spaced time interval
λt	analysis time interval for sampling around a given t
$\lambda_{\mathbb{T}} t$	analysis time interval specified in objective time
$\lambda_T t$	analysis time interval specified in event-driven time
$\mathcal{G}$	Time-Varying Graph (TVG)
V	set of vertices
$v \in V$	one vertex (node)
E	set of labeled edges
$e \in E$	one edge
$\rho(e,t)$	edge presence function
$\zeta(e,t)$	edge latency function
$\psi(v,t)$	node presence function
$\varphi(v,t)$	node latency function
G_{t_i}	static snapshot of TVG $\mathcal{G}$ at time t_i
$G(t,\lambda t)$	sampling (static graph) of $\mathcal{G}$ in a λt interval around time t
C_H	graph harmonic centrality
C_H^*	harmonic centrality for a given vertex
C_{AH}^*	average harmonic centrality over all vertices
C_{mH}^*	minimum harmonic centrality over all vertices
C_{MH}^*	maximum harmonic centrality over all vertices
$C_{\sigma H}^*$	standard deviation of harmonic centrality over all vertices
C_B	graph betweenness centrality
C_B^*	betweenness centrality for a given vertex
C_{AB}^*	average betweenness centrality over all vertices
C_{mB}^*	minimum betweenness centrality over all vertices
C_{MB}^*	maximum betweenness centrality over all vertices
$C_{\sigma B}^*$	standard deviation of betweenness centrality over all vertices
C_D	average degree centrality over all vertices, or density
C_D^*	degree centrality for a given vertex
$d(u,v)$	distance between nodes u and v
$d_{min}(u,v)$	shortest path between u and v
$dia(G)$	diameter of the network
$\sigma_{xy}(v)$	number of shortest paths between x and y passing through v
σ_{xy}	number of shortest paths between x and y
$D(\mathcal{G})$	dynamics measure, L^2 norm of changing centralities C_H, C_B, C_D
$\mathcal{G}_i$	Time/Identity-Varying Graph (TIVG) of $\mathcal{G}$ under identity $i \in I$
I	set of possible identities
$f(i, \mathcal{P}(V))$	node-identity function, applying identity $i \in I$ to vertex set V
$\Omega(\mathcal{G}, i, f)$	TIVG transformation function, applying identity $i \in I$ to $\mathcal{G}$, to produce $\mathcal{G}_i$
$V[i]$	node-identity classes, given by identity i
$E[i]$	set of edges, after applying f_i
$\rho[i]$	edge presence function under f_i
$\psi[i]$	node presence function under f_i

Table B.2: Notation used throughout this dissertation, continued

$G = (V, E)$	static, flat graph
$G_s = (V, E_s)$	the star-graph of G
$G_c = (V, E_c)$	the complete graph (clique) of G
$G_C = (V, E)$	a constrained graph containing at least a minimal star
$N = \|V\|$	number of vertices in G
γ	observed percentage between star and complete G (density)
γ_0	initial γ parameter
$\mu(t)$	ratio of add-to-delete edges for time t
η	amount of jitter (activity) in the network; the number of edges added per event

C Additional Calculations

C.1 Harmonic Centrality

The normalization of harmonic centrality, $C_H(G)$, follows from the predefined maximum: a star graph. In the star graph, the hub node $h \in V$ will have harmonic centrality,

$$C_H^*(h) = 1,$$

as discussed in Rochat [29] and others. All other nodes $v \in V$, being 1 connection away from the hub and 2 connections from all other nodes, will have centrality

$$\begin{aligned} C_H^*(v) &= \frac{1}{N-1}\left(1 + (N-2)\frac{1}{2}\right) \\ &= \frac{N}{2N-2}. \end{aligned}$$

To discover the coefficient for normalization, we must solve the following equation for x, noting that we have $N-1$ nodes whose difference from the maximal centrality is equal.

$$\begin{aligned} C_H(G) &= x\sum_{v\in V}\left(\max_{u\in V}(C_H^*(u)) - C_H^*(v)\right) \\ 1 &= x(N-1)\left(1 - \frac{N}{2N-2}\right) \\ 1 &= \frac{N-2}{2}x \\ x &= \frac{2}{N-2}. \end{aligned}$$

Table D.3: Union Sampling: Increase in average number of edges as η and λt increase for the unconstrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0.00	0.00	0.00	0.00	0.00
1	2.00	97.09	188.38	355.95	505.52
2	3.99	190.33	362.40	660.65	907.65
3	5.98	279.89	523.23	921.54	1,227.47
4	7.97	365.99	671.94	1,144.95	1,481.85
5	9.96	448.68	809.33	1,336.64	1,684.22

We may therefore formally define normalized harmonic centrality as

$$C_H(G) \quad = \quad \frac{2}{N-2} \sum_{v \in V} \left(\max_{u \in V} (C_H^*(u)) - C_H^*(v) \right).$$

D Additional Results of Sampling Size Effects

D.1 Unconstrained Networks

Similar to the constrained network cases in Section 3.5.1, our unconstrained networks exhibited the same properties as λt and η increase. This is explained by the algorithm used to create our synthetic TVGs, which used edge addition and deletion to create jitter. For the constrained network generation, we fix the minimal star and allow other edges to jitter. In the unconstrained case, in which the algorithm was allowed to remove any edge from the existing network at each timepoint, there are $|V| - 1$ fewer edges existing network since the minimal star is not fixed beforehand. Therefore, we find that these values differ slightly from the earlier constrained values, however they appear equivalent. Tables D.3 and D.5 displays the average number of additional edges included in the sampled network as λt and η increase for the Union and Intersection sampling methods respectively. Tables D.4 and D.6 extrapolate those values into the percentages of our theoretic maximal change, $2\lambda t \cdot \eta$, existing in each sampling size and jitter activity.

Table D.4: Union Sampling: Average sustained percentage of the additional $2\lambda t\eta$ edges for each η and λt over the unconstrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0.00	0.00	0.00	0.00	0.00
1	99.92	97.09	94.19	88.99	84.25
2	99.84	95.16	90.60	82.58	75.64
3	99.74	93.30	87.21	76.80	68.19
4	99.65	91.50	83.99	71.56	61.74
5	99.59	89.74	80.93	66.83	56.14

Table D.5: Intersection Sampling: Decrease in average number of edges as η and λt increase for the unconstrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0.00	0.00	0.00	0.00	0.00
1	-2.00	-97.06	-188.41	-356.12	-505.59
2	-4.00	-190.37	-362.37	-661.05	-908.04
3	-6.00	-280.03	-523.52	-922.50	-1,228.38
4	-8.00	-366.20	-672.40	-1,146.22	-1,483.19
5	-9.99	-449.04	-810.00	-1,337.61	-1,685.84

Table D.6: Intersection Sampling: Average sustained percentage of the $2\lambda t\eta$ less edges for each η and λt over the unconstrained network with $\gamma_0 = 0.5$.

λt \ η	1	$0.5N$	N	$2N$	$3N$
0	0	0	0	0	0
1	99.92	97.06	94.21	89.03	84.26
2	99.92	95.18	90.59	82.63	75.67
3	99.93	93.34	87.25	76.88	68.24
4	99.95	91.55	84.05	71.64	61.8
5	99.94	89.81	81	66.88	56.19

E Implementation Details

E.1 Time-Varying Graph: TemporalGraph

```
package com.robbiehott.temporalgraph.graph;


import com.robbiehott.temporalgraph.sampling.Sampling;
import com.robbiehott.temporalgraph.sampling.SamplingInterface;
import com.robbiehott.temporalgraph.time.TimeInterval;
import org.graphstream.graph.EdgeRejectedException;
import org.graphstream.graph.ElementNotFoundException;


import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;
import java.util.TreeSet;
```

```
/**
 * Temporal Graph
 *
 * This class is an implementation of a Time-Varying Graph (TVG), consisting of edges and vertices that each have
 * a presence function.
 *
 * @param <V> Vertex value class, usually a String, but may be any object
 * @param <E> Edge value class, usually a String, but may be any object
 * @param <T> Class that defines "time," which must be comparable for the time and events to be ordered
 */
public class TemporalGraph<V, E, T extends Comparable> {

    /**
     * Set of temporal vertices
     */
    private Set<TemporalVertex> vertices;

    /**
     * Set of temporal edges
     */
    private Set<TemporalEdge> edges;

    /**
     * Ordered set of events that happen
     */
    private TreeSet<T> events;

    /**
     * Constructor
     *
     * Creates an empty graph.
     */
    public TemporalGraph() {
        vertices = new HashSet<TemporalVertex>();
        edges = new HashSet<TemporalEdge>();
        events = new TreeSet<>();
    }

    /**
     * Add a temporal vertex
     *
     * Adds a temporal vertex to the TVG, if the TVG doesn't already contain the vertex
     *
     * @param vertex Temporal vertex to add
     */
    public void addTemporalVertex(TemporalVertex vertex) {
        if (!this.vertices.contains(vertex)) {
            this.vertices.add(vertex);
```

```
            this.events.addAll(vertex.getTemporalEvents());
        }
    }

    /**
     * Add a temporal edge
     *
     * Adds a temporal edge to the TVG, if every endpoint in the edge exists as a vertex in the graph.  This does not
     * currently check that the vertex actually exists during the time the edge is present.
     *
     * @param edge Temporal edge to add
     */
    public void addTemporalEdge(TemporalEdge edge) {
        boolean verticesExist = true;

        Iterator<TemporalVertex> itr = edge.getEndpoints().iterator();

        while (itr.hasNext()) {
            TemporalVertex vertex = itr.next();
            if (!this.vertices.contains(vertex))
                verticesExist = false;
        }

        if (verticesExist) {
            this.edges.add(edge);
            this.events.addAll(edge.getTemporalEvents());
        }
    }

    /**
     * Add Edge by value
     *
     * Creates a new edge out of the edge value and adds the temporal edge to the TVG.  Note: these created edges will
     * not have endpoints.
     *
     * @param edge Edge value to add
     */
    public void addEdge(E edge) {
        TemporalEdge<E, T> tempEdge = new TemporalEdge<>(edge);
        this.addTemporalEdge(tempEdge);
    }

    /**
     * Add Vertex by value
     *
     * Creates a new vertex out of the vertex value and adds the temporal vertex to the TVG.
     *
     * @param vertex Vertex value to add
     */
```

```
public void addVertex(V vertex) {
    TemporalVertex<V, T> tempVertex = new TemporalVertex<>(vertex);
    this.addTemporalVertex(tempVertex);
}

/**
 * Get temporal edge by value
 *
 * Returns the temporal edge associated with the given edge value.
 *
 * @param value Edge value to look up
 *
 * @return Temporal edge associated with that value or null if no edge exists
 */
public TemporalEdge<E, T> getTemporalEdge(E value) {
    for (TemporalEdge edge : this.edges) {
        if (edge.getValue().equals(value))
            return edge;
    }
    return null;
}

/**
 * Get temporal vertex by value
 *
 * Returns the temporal vertex associtated with the given edge value.
 *
 * @param value Vertex value to look up
 *
 * @return Temporal vertex associated with that value or null if no vertex exists
 */
public TemporalVertex<V, T> getTemporalVertex(V value) {
    for (TemporalVertex vertex : this.vertices) {
        if (vertex.getValue().equals(value))
            return vertex;
    }
    return null;
}

/**
 * Get Temporal Vertices
 *
 * Returns the set of all temporal vertices in the graph
 *
 * @return Set of temporal vertices
 */
public Set<TemporalVertex> getTemporalVertices() {
    return vertices;
}
```

```
/**
 * Get Temporal Edges
 *
 * Returns the set of all temporal edges in the graph
 *
 * @return Set of temporal edges
 */
public Set<TemporalEdge> getTemporalEdges() {
    return edges;
}

/**
 * Get GraphStream Graph over interval
 *
 * Produces a flattened GraphStream version of the TVG.  This produces a static non-temporal graph (V,E) using the
 * provided sampling method over the time interval provided.
 *
 * @param interval Time interval over which to flatten the TVG
 * @param method The sampling method to use in flattening the TVG
 * @return GraphStream SingleGraph representation of the TVG over the provided interval
 */
public org.graphstream.graph.Graph getGraphStreamGraphOver(TimeInterval interval, Sampling method) {

    // Create new graph G
    org.graphstream.graph.Graph G = new org.graphstream.graph.implementations.SingleGraph(interval.toString());
    if (!method.requiresGraphExtension()) {
        for (TemporalVertex vertex : vertices) {
            if (vertex.isPresentOver(interval, method)) {
                // add the vertex to the graph G
                G.addNode(vertex.getValue().toString());
            }
        }
        for (TemporalEdge edge : edges) {
            // Apply the sampling method over the interval for the edge,
            // adding it if the edge exists during the interval.
            if (edge.isPresentOver(interval, method)) {
                // add edge.value to the graph G
                Set<TemporalVertex> verts = edge.getEndpoints();
                TemporalVertex one = null;
                TemporalVertex two = null;
                int i = 0;
                for (TemporalVertex v : verts) {
                    if (i++ == 0)
                        one = v;
                    else
                        two = v;
                }
                if (one != null && two != null && G.getEdge(edge.getValue().toString()) == null) {
```

```
                try {
                    G.addEdge(edge.getValue().toString(), one.getValue().toString(), two.getValue().toString());
                } catch (EdgeRejectedException e) {
                    // If the nodes for this edge don't exist, then just ignore the edge.  DON'T STOP.
                    System.err.println("On interval " + interval.toString());
                    System.err.println("    Skipped: " + e.getMessage());
                    System.err.println("             with EdgeRejectedException");
                } catch (ElementNotFoundException e) {
                    // If the nodes for this edge don't exist, then just ignore the edge.  DON'T STOP.
                    System.err.println("On interval " + interval.toString());
                    System.err.println("    Skipped: " + e.getMessage());
                    System.err.println("             with ElementNotFoundException");
                }
            }
        }
    }
} else {
    // We must do the graph extension from the method; the sampling method itself provides a method
    // for finding what vertices and eges exist over a given time interval.  This allows graph-wide
    // samplings rather than element-centric samplings: transitive closure vs node/edge existence.
    for (Object vObject : method.getVerticesOver(interval, this)) {
        TemporalVertex<V, T> v = (TemporalVertex<V, T>) vObject;
        G.addNode(v.getValue().toString());
    }

    for (Object eObject : method.getEdgesOver(interval, this)) {
        TemporalEdge<String, T> edge = (TemporalEdge<String, T>) eObject;
        Set<TemporalVertex> verts = edge.getEndpoints();
        TemporalVertex one = null;
        TemporalVertex two = null;
        int i = 0;
        for (TemporalVertex v : verts) {
            if (i++ == 0)
                one = v;
            else
                two = v;
        }
        if (one != null && two != null && G.getEdge(edge.getValue().toString()) == null) {
            try {
                G.addEdge(edge.getValue().toString(), one.getValue().toString(), two.getValue().toString());
            } catch (EdgeRejectedException e) {
                // If the nodes for this edge don't exist, then just ignore the edge.  DON'T STOP.
                System.err.println("On interval " + interval.toString());
                System.err.println("    Skipped: " + e.getMessage());
                System.err.println("             with EdgeRejectedException");
            } catch (ElementNotFoundException e) {
                // If the nodes for this edge don't exist, then just ignore the edge.  DON'T STOP.
                System.err.println("On interval " + interval.toString());
                System.err.println("    Skipped: " + e.getMessage());
```

```
                    System.err.println("            with ElementNotFoundException");
                }
            }
        }
    }

    return G;
}

/**
 * Get First Event Time
 *
 * Returns the timepoint of the first event in the evolution of the network.
 *
 * @return First event timepoint
 */
public T getFirstEventTime() {
    return this.events.first();
}

/**
 * Get Last Event Time
 *
 * Returns the timepoint of the last event in the evolution of the network.
 *
 * @return Last event timepoint
 */
public T getLastEventTime() {
    return this.events.last();
}

/**
 * Get Next Event Time
 *
 * Returns the timepoint of the next event in the evolution of the network following
 * the time given.
 *
 * @param current A timepoint to query
 * @return First event timepoint after current
 */
public T getNextEventTime(T current) {
    return this.events.higher(current);
}

/**
 * Event Iterator
 *
 * Produces an iterator over the event timepoints in the evolution of the network.
 *
```

```
     * @return Iterator over event timepoints
     */
    public Iterator<T> getEventIterator() {
        return this.events.iterator();
    }

    /**
     * Event over Range Iterator
     *
     * Produces an iterator over the event timepoints in the evolution of the network
     * between the given start and end times.
     *
     * @param start Time to start the iterator
     * @param end Time to end the iterator
     * @return Iterator over event timepoints
     */
    public Iterator<T> getEventIterator(T start, T end) {
        return this.events.subSet(start, true, end, true).iterator();
    }

    /**
     * Get Event Timepoints Around Time
     *
     * Gets a time interval of event times that occur within a specific
     * lambda time window around the given time point.
     *
     * @param timepoint The timepoint to examine
     * @param lambda The number of events around the timepoint to include in the time interval
     * @return A timeinterval containing plus/minus lambda events around timepoint
     */
    public TimeInterval<T> getEventsAround(T timepoint, int lambda) {
        T begin = null;
        T end = null;

        if (lambda == 0) {
            return new TimeInterval<>(timepoint, timepoint);
        }

        begin = this.events.ceiling(timepoint);
        end = begin;

        if (begin == null) {
            return new TimeInterval<>(timepoint, timepoint);
        }

        T cur = begin;
        for (int i = 0; i < lambda; i++) {
            cur = this.events.lower(cur);
            if (cur != null)
```

```
                begin = cur;
            else
                break;
        }

        cur = end;
        // unlike begin, we count the current end as one step forward because of ceiling above
        for (int i = 1; i < lambda; i++) {
            cur = this.events.higher(cur);
            if (cur != null)
                end = cur;
            else
                break;
        }

        return new TimeInterval<>(begin, end);
    }

    /**
     * To String
     *
     * Produces a string representation of this TVG.
     *
     * @return String representation of this TVG
     */
    public String toString() {
        String output = "{ \"Temporal Graph\" : { \"Vertices\" : [";
        for (TemporalVertex vertex : this.vertices)
            output += vertex.toString() + ",";
        if (output.endsWith(","))
            output = output.substring(0, output.length()-1);
        output += "], \"Edges\" : [";
        for (TemporalEdge edge : this.edges) {
            output += edge.toString() + ",";
        }
        if (output.endsWith(","))
            output = output.substring(0, output.length()-1);
        output += "] } }";
        return output;
    }

}
```

E.2 Time-Varying Graph: TemporalVertex

```
package com.robbiehott.temporalgraph.graph;

import com.robbiehott.temporalgraph.sampling.Sampling;
import com.robbiehott.temporalgraph.sampling.SamplingInterface;
import com.robbiehott.temporalgraph.time.TimeInterval;
import com.robbiehott.temporalgraph.time.Timeline;

import java.util.Set;

/**
 * Temporal Vertex Class
 *
 * Storage class for temporal vertices, which must maintain a timeline of their lifespan and the vertex's value.
 *
 * @param <V> Vertex value class, usually a String, but may be any object
 * @param <T> Class that defines "time," which must be comparable for the time and events to be ordered
 */
public class TemporalVertex<V, T extends Comparable> {

    /**
     * Lifespan of the vertex
     */
    private Timeline<T> presence;

    /**
     * Value of the vertex
     */
    private V value;

    /**
     * Constructor
     *
     * Create a new temporal vertex with a given value.  By default, it is not present at any time.
     * @param value
     */
    public TemporalVertex(V value) {
        this.presence = new Timeline<T>();
        this.value = value;
    }

    /**
     * Set vertex presence
     *
     * Set the vertex as present during this interval
     *
     * @param interval Interval of presence
```

```
 */
public void setPresent(TimeInterval interval) {
    this.presence.setValueAt(interval, true);
}

/**
 * Get earliest presence
 *
 * Returns the first time this vertex is present in the network.
 *
 * @return Timepoint where vertex is first present
 */
public T earliestPresence() {
    return this.presence.getBegin();
}

/**
 * Get latest presence
 *
 * Returns the last time this vertex is present in the network.
 *
 * @return Timepoint where vertex is last present
 */
public T latestPresence() {
    return this.presence.getEnd();
}

/**
 * Set vertex as not present
 *
 * Sets the vertex as not present during this interval
 *
 * @param interval Interval of non-presence
 */
public void setNotPresent(TimeInterval interval) {
    this.presence.setValueAt(interval, false);
}

/**
 * Checks presence at a timepoint
 *
 * Returns whether or not the vertex is present at a given point in time
 *
 * @param timepoint Timepoint to check presence
 * @return True if present, false otherwise
 */
public boolean isPresentAt(T timepoint) {
    return this.presence.getValueAt(timepoint);
}
```

```
/**
 * Is present over an interval
 *
 * Uses the given sampling method to determine if the vertex is present over the given interval.  The definition of
 * "present" is defined by the sampling method.  For example, "is present for the entirety of the interval" or "is
 * present at any point during the interval."
 *
 * @param interval Interval to test presence over
 * @param sampling com.robbiehott.temporalgraph.sampling.Sampling method to define presence over the interval
 * @return True if present, false otherwise
 */
public boolean isPresentOver(TimeInterval interval, Sampling sampling) {
    // System.out.print("Node " + value + ": ");
    return this.presence.getValueOver(interval, sampling);
}

/**
 * To String
 *
 * Returns a string representation of this temporal vertex
 *
 * @return String representation of the temporal vertex
 */
@Override
public String toString() {
    return "{ \"value\": \"" + this.value.toString() + "\", \"presence\" : " + this.presence.toString() + "}";
}

/**
 * Get Vertex value
 *
 * Returns the value of this vertex.
 *
 * @return Value of the vertex
 */
public V getValue() {
    return value;
}

/**
 * Get Vertex temporal events
 *
 * Returns a set of time points during which this vertex changes
 *
 * @return Set of time points
 */
public Set<T> getTemporalEvents() {
    return this.presence.getEvents();
```

```
    }

    /**
     * Equals
     *
     * Check the vertex equality to the given parameter.  Currently checks that the vertex values are identical, rather
     * than timelines/lifespans.  This allows the same vertex to be updated multiple times with varying lifespan events.
     *
     * @param o Vertex to compare
     * @return True if equal, false otherwise
     */
    public boolean equals(Object o) {
        if (o instanceof TemporalVertex ) {
            return ((TemporalVertex) o).getValue().equals(value);
        }
        return false;
    }


}
```

E.3 Time-Varying Graph: TemporalEdge

```
package com.robbiehott.temporalgraph.graph;

import com.robbiehott.temporalgraph.sampling.Sampling;
import com.robbiehott.temporalgraph.sampling.SamplingInterface;
import com.robbiehott.temporalgraph.time.TimeInterval;
import com.robbiehott.temporalgraph.time.Timeline;

import java.util.HashSet;
import java.util.Set;

/**
 * Temporal Edge
 *
 * This class defines the storage for a temporal edge, which contains its endpoints, timeline, and value.
 *
 * @param <E> Class that holds the edge's value (Presumably String or Integer)
 * @param <T> Class that defines "time," which must be comparable for the time and events to be ordered
 */
public class TemporalEdge<E, T extends Comparable> {

    /**
     * The timeline defining the lifespan of this edge
     */
    private Timeline presence;

    /**
     * The set of endpoints for this edge (it may be a hyperedge)
     */
    private Set<TemporalVertex> endpoints;

    /**
     * The value of this edge
     */
    private E value;

    /**
     * Constructor
     *
     * Construct a new edge with the given value
     *
     * @param value Value to assign this edge
     */
    public TemporalEdge(E value) {
        this.presence = new Timeline();
        this.endpoints = new HashSet<>();
        this.value = value;
```

```
}

/**
 * Set Edge present over interval
 *
 * Sets the edge as present over the given interval (closed left, open right)
 *
 * @param interval Interval to set present over (closed left, open right)
 */
public void setPresent(TimeInterval interval) {
    this.presence.setValueAt(interval, true);
}



/**
 * Set Edge not present over interval
 *
 * Sets the edge as not present over the given interval (closed left, open right)
 *
 * @param interval Interval to set not present over (closed left, open right)
 */
public void setNotPresent(TimeInterval interval) {
    this.presence.setValueAt(interval, false);
}

/**
 * Is present at a timepoint
 *
 * Checks to see if the edge is present at a given timepoint
 *
 * @param timepoint Timepoint to test presence at
 * @return True if present, false otherwise
 */
public boolean isPresentAt(T timepoint) {
    return this.presence.getValueAt(timepoint);
}

/**
 * Is present over an interval
 *
 * Uses the given sampling method to determine if the edge is present over the given interval.  The definition of
 * "present" is defined by the sampling method.  For example, "is present for the entirety of the interval" or "is
 * present at any point during the interval."
 *
 * @param interval Interval to test presence over
 * @param sampling com.robbiehott.temporalgraph.sampling.Sampling method to define presence over the interval
 * @return True if present, false otherwise
 */
public boolean isPresentOver(TimeInterval interval, Sampling sampling) {
```

```
    // System.out.print("Edge " + value + ": ");
    return this.presence.getValueOver(interval, sampling);
}

/**
 * Get the endpoints
 *
 * Get all endpoints of this edge.  There may be more than two, since this is an undirected and possible hyper graph.
 *
 * @return Set of endpoints (Temporal Vertices)
 */
public Set<TemporalVertex> getEndpoints() {
    return endpoints;
}

/**
 * Set all endpoints
 *
 * Sets all endpoints of this edge to the parameter.  This will remove any endpoints already set for this edge and
 * replace them.
 * @param endpoints Set of endpoints for this edge
 */
public void setEndpoints(Set<TemporalVertex> endpoints) {
    this.endpoints = endpoints;
}

/**
 * Add an endpoint
 *
 * Adds and endpoint to the set of endpoints for this edge.
 *
 * @param vertex Vertex to add as an endpoint
 */
public void addEndpoint(TemporalVertex vertex) {
    this.endpoints.add(vertex);
}

/**
 * Get edge value
 *
 * Returns the value of this edge
 *
 * @return Edge value
 */
public E getValue() {
    return value;
}

/**
```

```
     * Set edge value
     *
     * Sets the value of the edge to the given parameter.
     *
     * @param value Value for the edge
     */
    public void setValue(E value) {
        this.value = value;
    }

    /**
     * Get edge temporal events
     *
     * Returns a set of time points during which this edge changes
     *
     * @return Set of time points
     */
    public Set<T> getTemporalEvents() {
        return this.presence.getEvents();
    }

    /**
     * To String
     *
     * Returns a string representation of this temporal edge
     *
     * @return String representation of the temporal edge
     */
    @Override
    public String toString() {
        String representation = "{ \"value\" : \"" + this.value.toString()
                + "\", \"endpoints\" : [";
        for (TemporalVertex point : this.endpoints) {
            representation +=  "{ \"value\" : " + point.getValue() + "},";
        }
        if (representation.endsWith(","))
            representation = representation.substring(0, representation.length()-1);
        representation += "],"
                + " \"timeline\" : [" + this.presence.toString() + "] }";
        return representation;
    }

    /**
     * Equals
     *
     * Checks the equality of the edge and the given parameter.
     *
     * @param o Temporal Edge to check equality against
     * @return True if equal, false otherwise
```

```
     */
    @Override
    public boolean equals(Object o) {
        if (!(o instanceof TemporalEdge))
            return false;

        // must have same number of endpoints
        if (this.endpoints.size() != ((TemporalEdge) o).endpoints.size())
            return false;

        // must have same endpoints
        boolean equal = true;
        for (TemporalVertex v : this.endpoints) {
            boolean inOther = false;
            for (Object ov : ((TemporalEdge) o).endpoints) {
                if (ov instanceof TemporalVertex && v.equals(ov)) {
                    inOther = true;
                    break;
                }
            }
            if (!inOther)
                equal = false;
        }
        for (Object ov : ((TemporalEdge) o).endpoints) {
            boolean inThis = false;
            for (TemporalVertex v : this.endpoints) {
                if (ov instanceof TemporalVertex && v.equals(ov)) {
                    inThis = true;
                    break;
                }
            }
            if (!inThis)
                equal = false;
        }

        return equal;
    }
}
```

F Additional Real-World Results

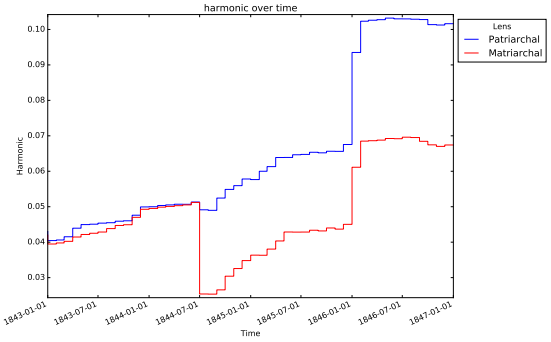
F.1 Nauvoo Marriage Project

As we included more individuals and marriages farther removed of the focused and well-vetted core group, the Anointed Quorum, the quality of the data dropped dramatically. For many of the less-defined individuals, our dataset does not have specific birth, death, or marriage dates. We accounted for the imprecision in the data in two ways: we assumed for analysis that any individual that did not have life dates existed throughout the entire observation window, and we only included those individuals who were connected by birth or marriage to the core group. Therefore, we created a network of 10 degrees of separation from the Anointed Quorum.

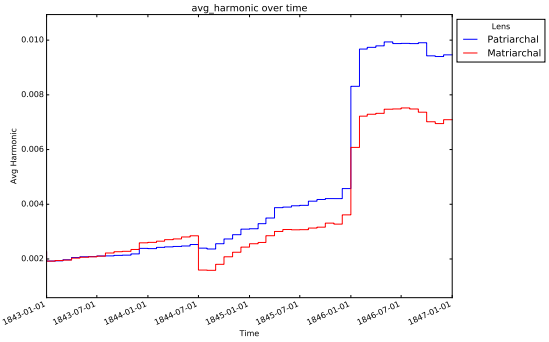
Let us therefore discuss Experiment 13. Figure F.2 displays the centrality values across time comparing the patriarchal and matriarchal networks. We notice a similar pattern to the centralities over the smaller set, the Anointed Quorum with one degree of separation, as seen in Figure 4.4. In contrast, the additional size of the network reduced the overall centrality measures.

In these plots, Joseph Smith's death date still produces a significant drop in betweenness for the matriarchal network and the marriages performed once the temple is completed still produce a significant rise in centrality for the network under both identity lenses. Average harmonic centrality, depicted in Figure F.2b, shows that the network is preferring the role of the patriarchy; specifically that the patriarchal network becomes more cohesive and central overall compared with its matriarchal counterpart. However, the additional marriages and individuals considered in the larger network produce a more highly "between" matriarchal network as compared to the patriarchal network, seen in Figure F.2d.

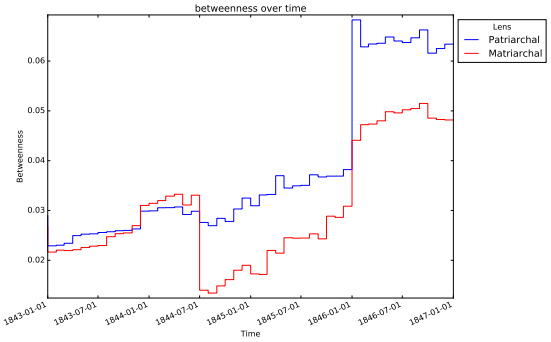
Figure F.2: Centrality measures across the patriarchal and matriarchal identity lenses.



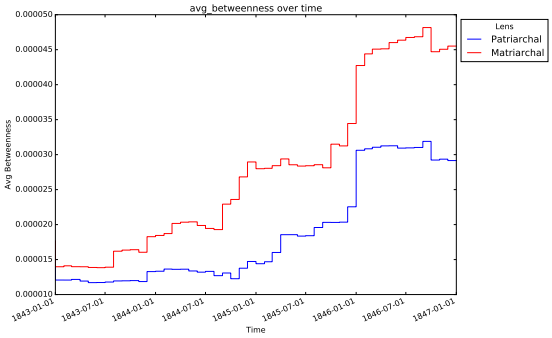
(a) Harmonic Centrality C_H



(b) Average Harmonic Centrality C^*_{AH}



(c) Betweenness Centrality C_B



(d) Average Betweenness Centrality C^*_{AB}

Figure F.3: Graphical representation of the SNAC Time-Varying Graph. This depiction shows the entire network flattened across its entire lifespan

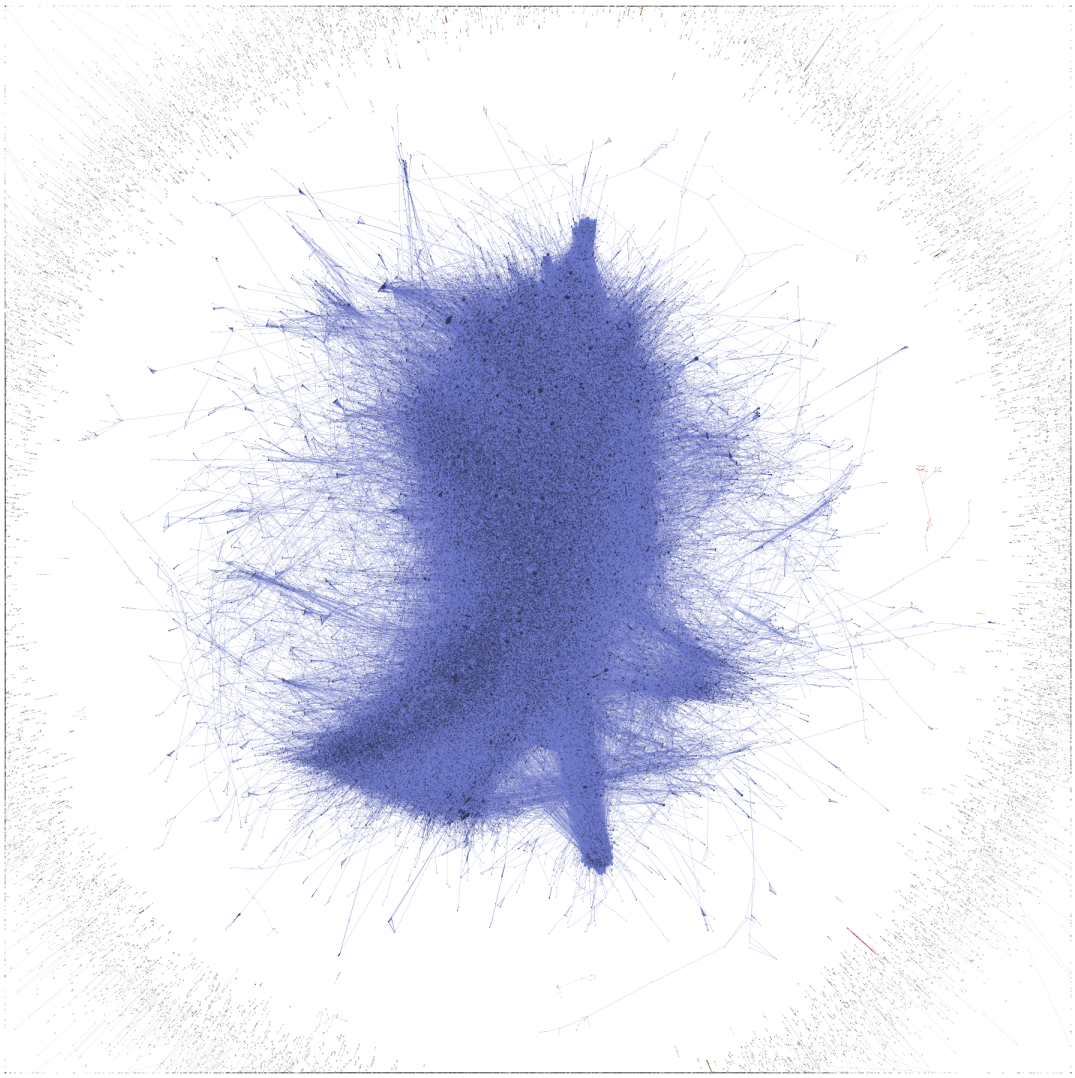


Figure F.4: Graphical representation of the ArXiv co-authorship network, from the institutional identity lens. This depiction shows the entire network flattened across its entire lifespan

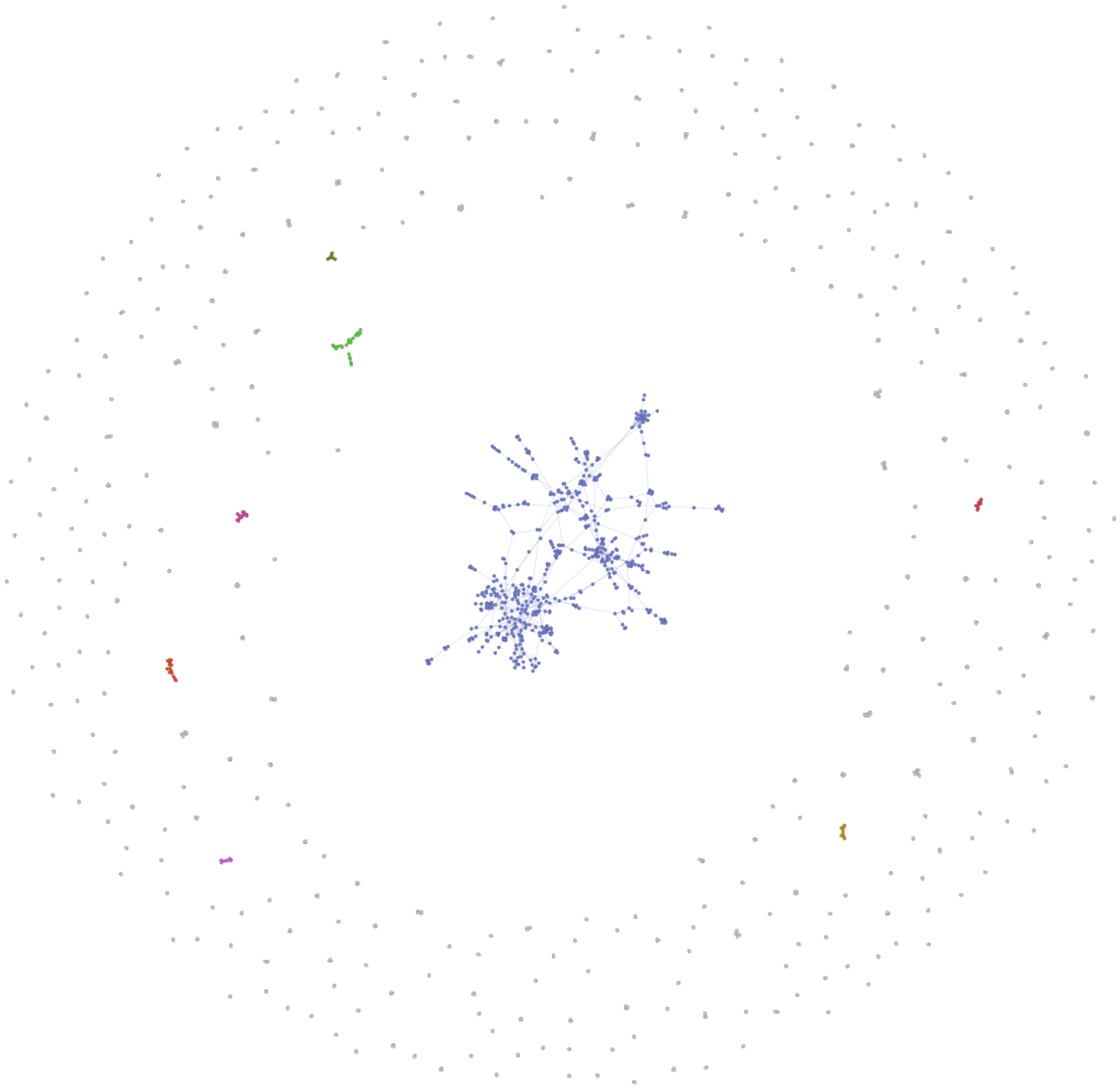


Figure F.5: Graphical representation of the ArXiv co-authorship network, from the author identity lens. This depiction shows the entire network flattened across its entire lifespan